A KNOWLEDGE-BASED TOXICOLOGY CONSULTANT FOR DIAGNOSING MULTIPLE DISORDERS

By

JOEL DANIEL SCHIPPER

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2008

For Alice, my beautiful wife.
May we grow ever nearer as the years go by.

ACKNOWLEDGMENTS

I am indebted and grateful to Dr. Jay L. Schauben, Dausear “Dar” McRae, and the Florida Poison Information Center in Jacksonville for their willingness to provide data, technical support, and consultation. Without them my research would not have been possible. I thank Dr. A. Antonio Arroyo and Dr. Douglas D. Dankel II for their guidance and encouragement throughout my doctoral studies. I also thank my parents and brothers, Tom and James, for their support and prayers throughout my academic career. I am grateful to Alice, my beautiful wife, for standing by me in all I do with constant, unwavering love. Above all, I give glory to God, my creator, without whom I am nothing.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Δ	A small positive parameter in the adjusted likelihood ratio equation
AAPCC	American Association of Poison Control Centers
CE	Clinical Effect
CF	Certainty Factor
FAR	False-Alarm Rate or False-Positive Rate
FN	False Negative
FNR	False-Negative Rate
FP	False Positive
FPIC	Florida Poison Information Center
FPR	False-Positive Rate or False-Alarm Rate
LR	Likelihood Ratio
MC	Minimum Exposure Cases
MCE	Minimum CE Occurrences
NPDS	National Poison Data System
NPV	Negative Predictive Value
PCC	Poison Control Center
PDA	Personal Digital Assistant
PPV	Positive Predictive Value
TESS	Toxic Exposure Surveillance System
TN	True Negative
TNR	True-Negative Rate or Specificity
TP	True Positive
TPR	True-Positive Rate or Sensitivity

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

A KNOWLEDGE-BASED TOXICOLOGY CONSULTANT FOR DIAGNOSING MULTIPLE DISORDERS

By

Joel Daniel Schipper

May 2008

Chair: A. Antonio Arroyo
Cochair: Douglas D. Dankel II
Major: Electrical and Computer Engineering

Every year, toxic exposures kill twelve hundred Americans. More than half of these deaths are the result of exposures to multiple substances. In addition to being dangerous, multiple exposures are particularly difficult to diagnose. At this time, no general solution exists for the diagnosis of multiple disorders due to the non-linear interactions observed in such cases.

This dissertation presents the development of a prototype knowledge-based system for diagnosing toxic exposures. The goal of the system is to generate differential diagnoses for unknown exposure cases based on the clinical effects observed in patients. The system is not meant to replace physicians, but, rather, to serve as a medical decision support system. Acting as a consultant, the system provides access to case-based summary data that is normally unavailable.

The system is automatically generated by applying data mining techniques to a database supplied by the Florida Poison Information Center. For diagnosis, the system uses pre-test probabilities and likelihood ratios—calculations commonly used throughout the medical profession. To overcome certain shortcomings of likelihood ratios, the equation employed by the system is adjusted to account for every possible outcome. Using the adjusted likelihood ratio

enables robust calculations while closely modeling the likelihood ratio that physicians know and trust.

Trained and tested on single exposures, the system achieved an accuracy of 81.0% on cases involving at least three clinical effects. Repeating the process for multiple exposures alone resulted in a failure, at least partially due to insufficient data. However, training on various combinations of single, double, and/or multiple exposures, the system achieved an accuracy of 86.9% when diagnosing the primary contributors for multiple exposure cases.

Although a solution for diagnosing multiple disorders remains elusive, the ability to identify primary contributors is a significant contribution to addressing the problem. This system is the first American diagnostic system for the field of clinical toxicology and its use of adjusted likelihood ratios serves as a method to bridge the gap between intelligent systems and the medical field. Furthermore, by automatically generating the system, this research addresses the knowledge acquisition bottleneck that plagues traditional expert systems.

CHAPTER 1
INTRODUCTION

Toxicology is the study of poisons and their effects on living organisms. One of the most prominent uses of toxicology for the benefit of mankind is the development of poison control centers. Thousands of people call poison control centers daily for free consultation and information regarding chemicals and drugs. In 2004, the American Association of Poison Control Centers (AAPCC) consisted of 62 poison control centers serving all 50 of the United States and handling more than 2.4 million reported human poison exposure cases (Watson et al., 2004). The AAPCC has compiled a database containing the details of over 38.7 million human poison exposure cases from the calls received and documented by its 62 poison control centers (Watson et al., 2004). A medical database of this magnitude represents a great opportunity for data mining and knowledge-based systems research. By tapping into the vast amount of data contained in the AAPCC database, a knowledge-based system could use the information to help diagnose and treat poison patients quickly and effectively.

Applicability of Knowledge-Based Systems to Toxicology

Knowledge-based systems should not be applied in every situation. In many cases, conventional algorithms offer a more appropriate and effective solution to the problem. However, the field of medicine inherently contains many traits that make it an ideal domain for knowledge-based systems. On a daily basis, physicians must make decisions based on experience using incomplete data. Knowledge-based systems also excel at solving problems from uncertain data using heuristics. Additionally, the field of medicine is continually changing as more knowledge is acquired and new technology becomes available. Likewise, a strength of knowledge-based systems is adaptability in dynamic domains. Beyond the general obstacles

common to all fields of medicine, the field of toxicology itself faces three specific challenges for which knowledge-based systems are well tailored.

The first challenge is making pertinent information available to physicians, emergency medical services, and the public involved at the time of a poisoning. Toxicology is a narrow specialization within the medical profession consisting of a small number of experts, called toxicologists. To make the expertise of toxicologists available to the medical field at large, the AAPCC offers direct consultation with toxicologists to physicians at hospitals around the country. Physicians may call poison control centers for information on how to treat a drug overdose or identify an unknown drug that a patient has ingested. In spite of the efforts of the AAPCC, the limited number of toxicologists makes expertise in toxicology a scarce commodity. Knowledge-based systems offer a solution to this scarcity. Creating a readily available system that can aid physicians in diagnosis when experts are unavailable could be an invaluable asset in saving lives.

A second challenge in toxicology is dealing with cases involving multiple substances. In many cases, consultations are a simple matter for the toxicologist, consisting mainly of matching signs and symptoms that are known to be directly associated with the mechanisms and behaviors of one class of drug. Cases that toxicologists find difficult tend to consist of multiple unknown drugs interacting to produce signs and symptoms that cannot be matched with any single substance. If all substances had linear interactions, determining multiple unknown drugs by their signs and symptoms would amount to identifying the drug combinations that, when summed together, produce the observed results. Unfortunately, many drug interactions are non-linear. Some drug combinations cause a dramatic increase in symptom severity, some mask symptoms normally observed with one of the drugs, and some can cause symptoms that normally would not

appear with any of the drugs individually. In 2004, although only 8.6% the exposures reported were multiple substance exposures, "50.6% of fatal cases involved 2 or more drugs or products" (Watson et al., 2004, p. 593). Being able to address multiple exposures is an important concern for saving lives. A knowledge-based system can aid in addressing multiple exposures by effectively making the relevant information in the AAPCC database available to the toxicologist. The goal of a knowledge-based system is not to replace the toxicologist, but to act as a powerful consulting tool providing case-based summary data for the toxicologist. Human beings have senses and intuition that are important for diagnosis, which computers cannot replicate. However, by offering speculative advice, the system may facilitate accurate and timely diagnoses.

A third toxicological challenge is ensuring the rapid diagnosis and treatment of exposures. When dealing with poisons and drug overdoses, time is of the essence. In 2004, 1183 people died of toxic exposures (Watson et al., 2004), many because they did not receive the correct treatment in a timely manner. Every minute spent waiting to speak with an expert or consult a clinical manual could make the difference between life and death for a patient. A knowledge-based system is a rapid aid in diagnosing toxic exposures. Because the system is computerized, it offers physicians a directed search with a faster response time than written literature.

System Overview

The goal of this research is to create a general purpose knowledge-based system that can automatically learn relationships in diagnostic domains. This particular application of the system uses the Florida Poison Information Center (FPIC) database as its foundation. By mining the FPIC database, the system extracts associations between the signs and symptoms observed in a patient and the final diagnosis. Automatically extracting these relationships enables system

designers to bypass much of the knowledge acquisition bottleneck by removing the need to interview experts, a requirement for traditional knowledge-based system design. Furthermore, by applying a generalized process, knowledge engineers need not acquire a comprehensive understanding of every domain for which they create a knowledge-based system.

Being applied to toxicology, the system utilizes the simple, standard medical mathematics of pre-test probabilities and likelihood ratios to calculate and communicate the relationships discovered in the database. Currently, the system is a proof-of-concept prototype and only primary contributors with a significant number of exposure occurrences are included. As the system grows to include more substances and substance combinations, however, the system's simple, mathematical representation will become essential for scalability purposes. Additionally, medical mathematics is not only more understandable to users in the medical field, but communicates information that is more relevant for medical diagnosis than other traditional measurements, such as accuracy (Cios & Moore, 2002; Lavrac, 1999).

In spite of the simplicity necessarily inherent in the system, the system seeks to diagnose complex cases involving multiple unknown substances. In the past, very little research has been performed in the area of diagnosing multiple disorders, and this system seeks to further the fields of machine intelligence and knowledge engineering by offering a simple and practical approach for addressing the problem. Fundamentally, the system treats multiple disorders in much the same way as single disorders. Multiple exposures are treated as a separate case from the individual substances involved, with identical operations being performed on each multiple exposure case to create associations. At this time, the data available is insufficient to fully test the diagnosis of multiple exposures; however, the system demonstrates significant potential in accurately identifying the primary contributors in multiple exposure cases.

Ultimately, the system's goal is to serve as a consultant to all physicians that may encounter toxic exposure cases. For a toxicologist, the system may serve as an idea generator by offering plausible drug combinations that perhaps the toxicologist failed to consider. For other physicians, the system may act as a solution finder or simply be used to confirm an uncertain diagnosis. As the system develops, expanding to encompass the entire FPIC database, the system may begin to discover relationships previously undocumented in the field of toxicology. Further development may lead to the real-time monitoring of cases as they are entered into the database so the system can signal a warning for epidemics or perceived threats, such as substances associated with terrorism.

Database Resources

The Florida Poison Information Center (FPIC) consists of three of the poison control centers in the AAPCC. Since 1996, the FPIC has compiled a database logging every call it receives. When a caller goes to the hospital, the FPIC makes a follow-up call to gather all the medical information available on the case. In 2004 alone, the FPIC received over 120 thousand calls and made more than 43 thousand follow-up calls related to human exposures (Florida Poison Information Center Network, 2005). The FPIC database also contains over 65 thousand records of multiple exposure cases. Entries in the database are regulated by AAPCC Toxic Exposure Surveillance System (TESS) standards that ensure the collection of a specific set of information about each case. In following TESS requirements, the majority of entries in the database have discrete values that are easy to process with a computer program. Furthermore, the national standardization by the AAPCC increases portability of a system designed for the FPIC to other poison control centers around the country.

For this research, the FPIC has generously granted access to all relevant information recorded in their database from 2002-2006. Initially, concerns were expressed regarding the

accuracy of the data. In some cases, patients may lie about the drugs they took. In others, nurses may relay either inaccurate or incomplete information to the FPIC. Although these errors affect system accuracy, the system's performance shows that, in general, the discrepancies can be treated as random errors whose contributions will become negligible as the database grows. Another problem is that two drugs taken together in varying proportions can yield different symptoms. The observed symptoms might vary depending on the amount of interaction occurring between the two drugs or which drug is affecting the body more strongly at the time. However, the findings of the system indicate that most multiple exposures are dominated by the signs and symptoms associated with a primary contributor. As a result, the system is capable of diagnosing the primary contributor, which is a significant contribution to addressing the problem of multiple disorder diagnosis.

Conclusion

This chapter began by introducing the field of toxicology and the American Association of Poison Control Centers. It then explained the applicability of knowledge-based systems to the medical field, particularly the field of toxicology. Finally, it presented a broad overview of the system followed by a discussion about the database used in the development of the system.

Chapter 2 presents a general overview of knowledge-based systems and data mining, while Chapter 3 elaborates on Chapter 2 by describing in greater detail the traditional approaches used in designing knowledge-based systems. Chapter 4 discusses medical mathematics and gives a literature review of relevant systems that have been created. Chapter 5 presents the system design in detail along with the results for diagnosing single exposure cases. Finally, Chapter 6 presents the results for diagnosing multiple exposure cases followed by some concluding remarks.

CHAPTER 2
OVERVIEW OF KNOWLEDGE-BASED SYSTEMS AND DATA MINING

In the past two decades, the availability of information has skyrocketed. Continual advances in computer technology have made the collection and storage of massive amounts of data a reality, while the advent of the Internet has enabled the data to be shared and accessed by many users throughout the world. Today, the volume of data generated and stored is so enormous that it has become impossible for the human mind to locate and process most of the available information. Furthermore, as we learn more about the complexity of our world, researchers and practitioners alike are forced to specialize to the point where only a few people are truly knowledgeable in any particular field. If humans are to continue in the quest to understand and subdue the world, it has become imperative that we create systems and algorithms capable of filtering out useless data while identifying, processing, and applying relevant information.

Knowledge-Based Systems

Utility and Structure

Knowledge-based systems, also known as expert systems, are computerized systems that use information to provide relevant advice and problem solutions within a specific domain. Knowledge-based systems enable expert knowledge to be accessed 24 hours a day, even when an expert is unavailable. They also provide a means to preserve information that otherwise might be lost when an expert retires.

Figure 2-1 shows the basic structure of an expert system consisting of an inference engine, a knowledge base, and a fact base. The inference engine is a program that manipulates the knowledge base and fact base using a general problem solving technique. The knowledge base is the fixed set of information or data that is necessary to solve problems within a particular

domain. The fact base contains problem-specific data, such as user inputs and information derived from the knowledge base by the inference engine (Gonzalez & Dankel, 1993).

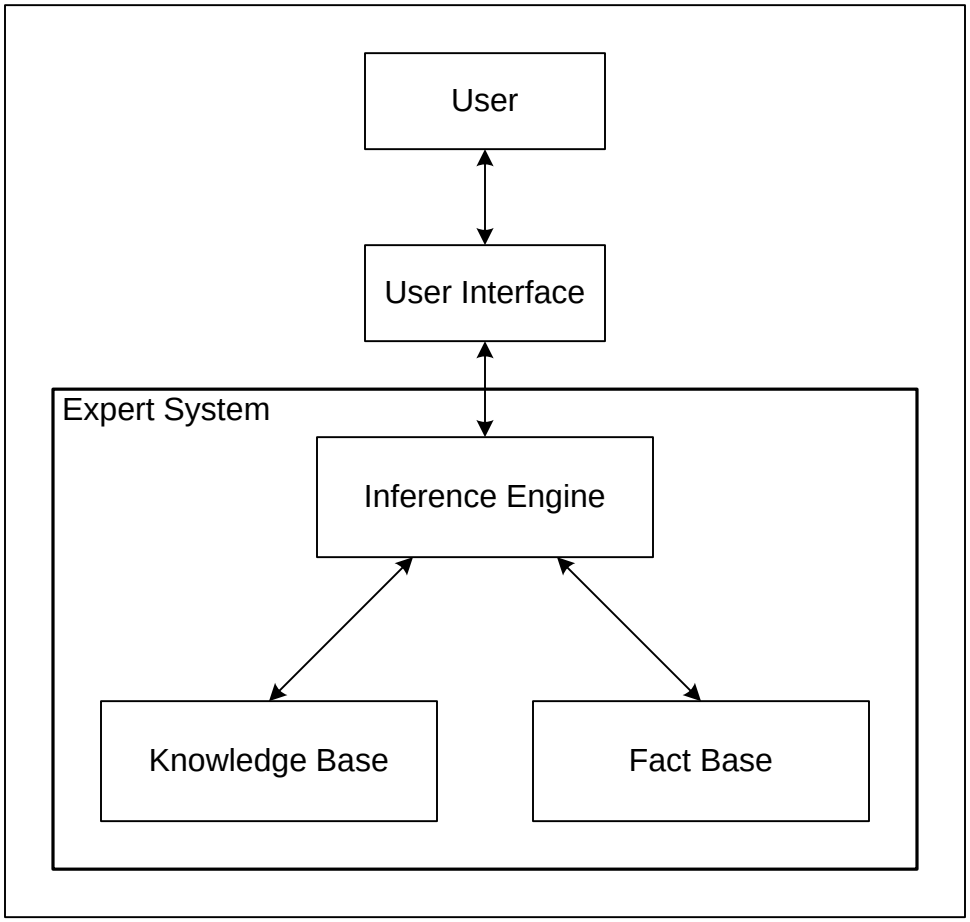

Figure 2-1. Expert system block diagram

Unlike conventional algorithms that embed domain knowledge within the program, inference engines are problem independent. Such independence provides versatility, enabling the inference engine to be applied to any number of domains by simply changing the knowledge base. The same diagnostic inference engine could be effectively applied to the medical field as well as automobile repair or trouble shooting a manufacturing process. The beauty of this independence is that it allows the programmer to focus on the domain knowledge, often expressed as facts and rules, without having to debug faulty algorithmic code.

Reasoning from Examples

For a knowledge-based system to produce accurate results, it must obtain its conclusions via some logical process. In logic, there are three fundamental ways of reaching a conclusion: deductive reasoning, inductive reasoning, and abductive reasoning. Deductive reasoning is reasoning from general to specific. For example:

Premise: All oceans have waves.
Premise: The Pacific is an ocean.
Conclusion: Therefore, the Pacific has waves.

Deductive reasoning is a sound form of argument, meaning that if its premises are true its conclusion is guaranteed to be true as well. Inductive reasoning is inferring from specific to more general statements. For example:

Premise: The Pacific is an ocean.
Premise: The Pacific has waves.
Conclusion: Therefore, oceans have waves.

Inductive reasoning is an unsound form of reasoning, meaning that even if the premises are true the conclusion is not guaranteed to be true. Abductive reasoning is drawing a hypothesis based on observed characteristics. For example:

Fact: Oceans have waves.
Observation: This body of water has waves.
Hypothesis: This body of water is an ocean.

Like inductive reasoning, abductive reasoning is unsound. In fact, abductive reasoning can be viewed as a form of inductive reasoning because it is reasoning from specific observations to draw generalized hypotheses that are plausible but not guaranteed (Gonzalez & Dankel, 1993).

An ideal knowledge-based system should offer the correct solution to every problem within its domain. To guarantee the validity of every solution, the system would have to contain all first principles within its domain and employ a sound reasoning technique, such as deductive reasoning. Although some systems attempt to reason from first principles, in general, attempting

to program a system in such a manner is not practical or even possible. Many fields are not understood well enough to compile a list of foundational rules and, even if they were, the compilation and programming of such rules would prove an extremely arduous task for any domain of significance.

Because we cannot create an ideal knowledge-based system, many systems take the practical approach of reasoning using examples. Rather than directly programming the system from first principles, the system is given examples from which it generates its own governing principles. These principles can be expressed as rules, statistics, case matching, or another representative form. Chapter 3 discusses many of these approaches. Like inductive reasoning, the system makes inferences from the specific, i.e. an example, to the general, i.e. a governing principle. At first, such an approach seems troublesome because the system's reasoning is unsound and, therefore, inherently can make mistakes. It is important to note, however, that scientists discovered every scientific principle that we accept as fact in the same manner: by observing that many examples, or experiments, all followed the same law of nature. Additionally, Kononenko et al. (1998) have shown that, in many domains, systems that automatically generate their own diagnostic rules are capable of performing with a higher degree of accuracy than physicians, when given identical information. Furthermore, knowledge-based systems that generate their own governing principles for problem solving take less time to create because the programmer need not spend time determining the governing principles by hand. Instead, the system itself can determine its own rules by processing a database of examples.

Data Mining and Knowledge Discovery in Databases

Defining Data Mining and Knowledge Discovery

In recent years, the development of mass storage devices has enabled the creation of extremely large and complex databases. The amount of available data has increased so greatly

that it would be impossibly tedious for the human mind to process all of the information. As a result, the rising demand for systems capable of meeting this new need has given birth to the field of data mining. Data mining is the process of extracting information from a database (Han & Kamber, 2001). Data mining is also referred to as knowledge discovery in databases, where "discovery…is the generation of novel, interesting, plausible, and intelligible knowledge about the objects of study" (Valdes-Perez, 1999, p. 336). The knowledge discovered through data mining comes in two different forms: novel information and established information.

Novel information is previously undiscovered knowledge that is a new concept within a domain. To demonstrate how a computer system can uncover valuable information, let us consider the medical field. Most facts within the medical field were established by researchers who performed studies and processed data numerically to determine relationships between various observations. In these studies, it is the numerical and statistical values that give credence to the study. If a physician claims that smoking increases one's chance of lung cancer based on a general trend the physician has observed in his patients, the physician will be asked for the numbers to support his statement. It is not until the physician performs a study using a numerically significant amount of cases and produces values that support his statement that his observation will be taken seriously. Since computers perform numerical analysis exceedingly well, it makes sense to create data mining systems that will automatically search for and output the numerically significant relationships they encounter. Researchers can then examine and determine the validity of the discoveries made by the computer system.

The second form of knowledge that data mining systems can discover is established information. Established information is information that is already known and available within a domain. At first, rediscovery of such information may seem like a confirmation of knowledge at

best or redundant reiteration at worst. However, the ability to automatically discover established principles through data processing is in fact a powerful tool in the field of knowledge-based systems. Traditionally, knowledge engineers create knowledge-based systems incrementally through an interview process with experts in the domain of interest. In each interview, the knowledge engineer tries to glean rules and heuristics from the expert so that these can be implemented in the system. Using this process of generating an expert system takes years of man-hours to complete and forces the knowledge engineer to train himself through immersion in the domain to create the system properly. It has always been desirable to shorten the creation time of these systems without compromising their accuracy. The solution to these problems lies in the power of data mining to automatically extract rules and knowledge from a database without the necessity of a human to accumulate and program these rules directly.

Seven Steps of Data Mining

According to Han and Kamber (2001), there are seven steps in data mining (Table 2-1). Do not be confused that data mining is listed as only one of the steps in knowledge discovery in databases. In practice, the step of data mining is indispensable and usually requires the most computation and intelligence. As a result, the whole process of knowledge discovery in databases has commonly become known as data mining.

Table 2-1. Seven steps of data mining

Step name	Description
Data cleaning	Removing noisy, inconsistent data from the database
Data integration	Combining data from multiple sources
Data selection	Choosing the data relevant to the task
Data transformation	Changing the selected data into a useable format for data mining
Data mining	Extracting relationships and patterns from the data
Pattern evaluation	Determining if the knowledge discovered has value
Knowledge presentation	Presenting the results to the user via tables, graphs, charts, etc.

Although Table 2-1 lists seven steps in data mining, these steps are not rigidly enforced in data mining system design. Depending on the form of the data, not every step on the table is required for every data mining problem. For example, data integration is not required if only one data source is used. Some steps may be performed in a different order. For example, data cleaning may be handled by allowing a robust algorithm implemented in the data mining step to eliminate noise. Although many of these steps may be indispensable with a particular data set, in general the data mining and pattern evaluation steps are the core components of a data mining system.

Mining Data: What and How?

Table 2-2 summarizes the six types of patterns that can be mined according to Han and Kamber (2001). The research presented here focuses primarily on classification in the medical field of toxicology. In general terms, classification attempts to take a database of cases belonging to known classes and create models that are used to identify cases with unknown classes based on the information supplied about the case. Specifically, given a database containing the signs and symptoms observed in a patient paired with the appropriate diagnosis of the substances affecting the patient, a system can learn to identify different substances based on the associated signs and symptoms.

Many different methods can be implemented to perform data mining. Some overlap with the methods of knowledge-based systems, discussed in more detail in Chapter 3. A good summary of the most common methods can be found in Lavrac (1999) or Han and Kamber (2001). Zhou (2003) discusses three philosophical approaches to data mining which focus primarily on the efficiency, effectiveness, or validity of the system design.

Table 2-2. Types of patterns that can be mined

Patterns	Description
Characterization & Discrimination	Summarizes different classes within the data so they can be compared and contrasted with other classes
Associational analysis	Searches for rules that reveal relationships between different classes in the data
Classification & Prediction	Identifies models where, given certain inputs, the system can output the most probable class or number associated with the inputs
Cluster analysis	Treats every parameter as a value and groups the most similar cases into clusters that will be treated as a single class
Outlier analysis	Identifies cases that are sufficiently deviant from all other cases so they can be examined further
Evolution analysis	Searches for tendencies of class parameters to change over time in a characteristic manner

Conclusion

This chapter has given a general overview of knowledge-based systems and data mining. The section on knowledge-based systems discussed the general structure and usefulness of knowledge-based systems followed by the importance of reasoning from examples. The section on data mining presented the concepts of novel information and established information as well as the seven steps to data mining and the types of patterns that can be discovered. The next chapter discusses many of the different approaches to knowledge-based system design. Although these approaches are presented within the context of knowledge-based systems, many are used jointly in the field of data mining.

CHAPTER 3
DESIGN APPROACHES TO KNOWLEDGE-BASED SYSTEMS

Since the inception of knowledge-based systems in the 1970's, researchers have developed many varied approaches for their design and implementation. This chapter presents a brief overview of the most common design schemes, with an emphasis on those most similar to the system presented in Chapters 5 and 6. Although the schemes are presented within the context of knowledge-based systems, most are used jointly in the field of data mining. The chapter begins with the foundational topics of rule-based systems, case-based reasoning, nearest neighbor classification, and Bayes' rule, followed by a discussion of lesser topics, including fuzzy logic, Dempster-Schafer, rough sets, genetic algorithms, and artificial neural networks. The chapter concludes by discussing the modern approaches most relevant to solving problems involving multiple disorders, namely Bayesian belief networks and set covering theory.

Rule-Based Systems

In designing knowledge-based systems, the use of rules is an obvious choice. Not only do humans naturally use rules when they reason and solve classification problems, but rules inherently are heuristic in nature, enabling them to handle uncertainty. As discussed in Chapter 2, rule-based systems consist of an inference engine, a knowledge base, and a fact base (Figure 3-1). The inference engine is the general problem solving technique utilized by the system, such as the forward and backward chaining approaches discussed below. The knowledge base consists of a domain specific list of if-then statements, known as rules, used to gather information and solve problems. The "if" portion of a rule is known as the premise and the "then" portion of the rule is known as the conclusion. The fact base is problem specific and consists of knowledge obtained from the user and sensors along with all knowledge derived from

implemented rules. Greater detail on rule-based systems can be found in Gonzalez and Dankel (1993).

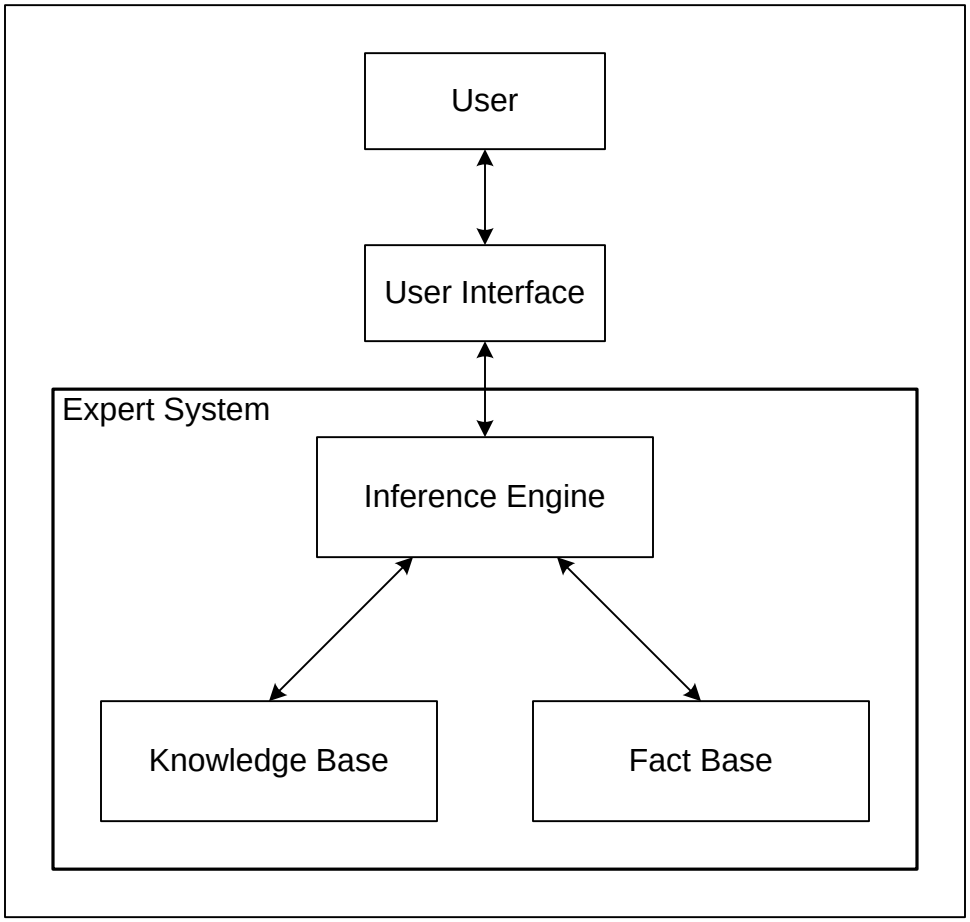

Figure 3-1. Rule-based system block diagram

To better understand rule-based systems, let us consider an example using Doug and his cats. Let us assume that Doug owns four cats named Princess, Panther, Ivan, and Jimmy, each requiring special care that it must receive daily. Doug wants to go on vacation, so he hires a pet-sitter and creates a list of the treatments for each cat (Table 3-1).

Table 3-1. Treatments required for each of Doug's cats

Cat's name	Treatment
Princess	Requires at least 30 minutes of petting per day
Panther	Given 50% more food
Ivan	Must not be allowed outside at all costs
Jimmy	Must receive antibiotics once a day

Doug soon becomes aware, however, that the pet-sitter does not know the names of the cats. To ensure that each cat receives the necessary treatment, he decides to create a rule-based system to help the pet-sitter identify the cats. He begins by writing down the distinguishing characteristics of each cat, including the cat's major color, fur length, and whether the cat's fur is a solid color or not (Table 3-2).

Table 3-2. Cat characteristics for identification

Cat's name	Major color	Solid color?	Fur length
Princess	Tan	No	Medium
Panther	Black	Yes	Short
Ivan	Tan	No	Short
Jimmy	Black	No	Short

Doug begins to define the parameters used by his system as well as their allowable values (Table 3-3). He quickly realizes that defining fur length as being short, medium, or long is a subjective measurement. To reduce uncertainty, he creates a new parameter called "FurMeasurement," that allows the pet-sitter to input a length of fur in inches. From this measurement, the fur length is determined.

Table 3-3. System parameters for cat identification

System parameters	Allowable values
MajorColor	black, tan
SolidColor	yes, no
FurMeasurement	Length of fur in inches
FurLength	short, medium, long
Cat	Princess, Panther, Ivan, Jimmy

Finally, Doug creates seven system rules that identify each cat based on the characteristics observed by the pet-sitter. As a whole, these rules are known as the knowledge base (Figure 3-2). The fact base contains any facts entered by the pet-sitter, such as stating that the unknown cat has a FurMeasurement = 1". Additional facts derived from the rule set are also included in the fact base, such as rule R1 deriving that the cat must have FurLength = short if FurMeasurement = 1".

Rules:

R1	If	FurMeasurement ≤ 1"
	Then	FurLength = short
R2	If	FurMeasurement > 1" AND FurMeasurement ≤ 2"
	Then	FurLength = medium
R3	If	FurMeasurement > 2"
	Then	FurLength = long
R4	If	FurLength = medium
	Then	Cat = Princess
R5	If	SolidColor = yes
	Then	Cat = Panther
R6	If	MajorColor = tan AND FurLength = short
	Then	Cat = Ivan
R7	If	MajorColor = black AND SolidColor = no
	Then	Cat = Jimmy

Figure 3-2. Rules for identifying Doug's cats

The following subsections discuss the basic inference engine algorithms used in rule-based systems. Throughout these sections, this example of Doug and his cats is referenced frequently.

Forward Chaining

Forward chaining is the process of reasoning from inputs to conclusions. The first step in a forward chaining system is to receive user and sensor inputs by storing them in the fact base. Next, the system searches the rule set and identifies those rules whose premises are satisfied by the facts contained in the fact base. The process of identifying these rules is called pattern matching. If more than one rule is satisfied, the system identifies the rule with the highest priority and executes it, also known as rule firing. The results obtained from the fired rule are added to the fact base. This process of pattern matching, prioritizing, and rule firing continues until a solution is reached or no solution can be reached. If no solution is attained, complex systems may request information from the user that might enable the system to reach a conclusion. Alternately, the system might apply uncertainty management to offer the most fitting solutions based on the facts it has received.

Using the example of Doug and his cats, the user might input that a cat has MajorColor = tan, FurLength = 1.5", and SolidColor = no. The forward chaining system adds these facts to the fact base and then searches the premise of every rule for a match. It discovers a match on R2 and a partial match on R6. Being the only rule satisfied, R2 fires, adding the fact FurLength = medium to the fact base. Once again, the system searches through the premises and finds matches on R2 and R4 as well as a partial match on R6. The system must now prioritize the rules. Since the rules closer to solutions are further down the list, the system gives priority to rules with higher rule numbers. Note that this also prevents the system from entering an infinite loop by evaluating R2 over and over again. R4 is selected as higher priority than R2, so R4 fires yielding the result that Cat = Princess. Since the variable Cat is the solution variable, the system stops and informs the user that the cat being observed must be Princess.

In complex rule-based systems, forward chaining is extremely inefficient due to the exhaustive search performed during pattern matching. To alleviate this bottleneck, an algorithm known as the Rete algorithm was developed. The Rete algorithm creates predetermined networks, known as the pattern network and the join network, to limit the amount of matching that must take place every cycle of the pattern matching process. The Rete algorithm along with the formation of pattern and join networks are discussed in detail by Gonzalez and Dankel (1993).

Forward chaining systems are used primarily for problems that involve a small number of inputs compared to the number of possible solutions. Synthesis problems, including design, configuration, planning, and scheduling problems, are good candidates for forward chaining applications. These types of problems are often open ended, where many solutions or configurations can satisfy all the given constraints. Since the solutions cannot be known until

they are generated, it would be impossible to work from the problem conclusions to the inputs. There are, however, many problems with a finite number of solutions and, in these cases, it may be advantageous to begin with the conclusions and work towards the inputs.

Backward Chaining

Backward chaining is the process of reasoning from conclusions to inputs. Backward chaining systems assume an answer and then attempt to prove or disprove the truth of that assumption. To begin this process, the system selects a rule whose conclusion yields a solution. The system then attempts to satisfy the rule by obtaining values for the variables in the premise of the solution. For each premise, the system first checks the fact base for the value, then searches for a rule that can generate the necessary value to satisfy the premise, and finally asks the user when all else fails. If the fact base contains a value that contradicts the premise, the system disregards the solution as invalid and assumes a new solution by moving on to a different rule. When examining a rule, if the fact base contains a value matching one of the rule's premises, the system continues to assume that the rule is correct and attempts to prove the next premise until all the premises are satisfied. If no value is found in the fact base for a premise, but a rule is discovered that can derive its value, the system attempts to prove the premises of that rule through the same process. If, however, no rule capable of satisfying the premise can be found, the system asks the user as a last resort. Then the user can enter a value, which the system adds to the fact base. If the value entered by the user corresponds to the necessary premise value, the system continues trying to prove the rule. If it contradicts the premise, the system moves to a new rule that can generate a solution. This process continues until the system has either found a solution or exhausted all rules capable of yielding a solution.

The backward chaining process is much easier to understand with an example, so let us return to Doug and his cats. When the pet-sitter enters Doug's house, Ivan comes over to greet

him. To identify Ivan, the pet-sitter consults the system that Doug designed for him. The backward chaining system knows that when the variable Cat has a value it has reached a solution. It begins by searching the list for a rule whose conclusion assigns a value to Cat. Rules R1, R2, and R3 do not have the variable Cat in the conclusion, so the system begins with R4. To prove that R4 is true, the system must satisfy the premise that FurLength = medium. It searches the fact base and finds nothing. Next, it searches for a rule whose conclusion can generate FurLength = medium and discovers that R2 satisfies this requirement. The system now attempts to prove R2 by looking at its first premise, FurMeasurement > 1". The system again checks the fact base and finds no matching values, so it searches for a rule that generates a corresponding solution. Finding none, it asks the user to input the length of the cat's fur in inches. The user inputs 0.5", the length of Ivan's fur. This value is saved in the fact base, but since 0.5" is not greater than 1", R2 fails and the system returns to R4. The system discovers that there are no more rules that can satisfy the premise FurLength = medium, so it discards R4 as false and proceeds to R5. R5's premise requires SolidColor = yes. The system searches the fact base and finds no values corresponding to SolidColor. It then searches for rules that can generate the value SolidColor = yes. Again if finds none. As a last resort, the system asks the user if the cat is one solid color, and the user enters "no." Since SolidColor now has a contradicting value, R5 fails and the system moves to R6. The first premise on R6 is MajorColor = tan. The system again checks the fact base and then searches for rules that can satisfy the parameter, but finds none. It asks the user for the cat's fur color. The user enters "tan." Since this satisfies the first premise, the system attempts to prove the second premise, FurLength = short. The system finds no facts in the fact base corresponding to FurLength; however, it finds that R1 can generate the desired solution. The system then attempts to satisfy R1 by looking at its premise

FurMeasurement ≤ 1". It checks its fact base and discovers that the fact base contains the value FurMeasurement = 0.5". Since this value satisfies FurMeasurement ≤ 1", both premises for R6 are satisfied and the system concludes that the cat is Ivan.

Backward chaining systems can only be used for problems that involve a finite number of conclusions. Diagnostic problems, where the inputs outnumber the solutions are good candidates for backward chaining applications. Diagnostic systems can vary from determining automobile malfunctions to properly identifying diseases in the medical field (Gonzalez & Dankel, 1993).

Inference Networks

Inference networks are some of the simplest rule-based systems and can only be used when the relationship between each rule is known in advance. Figure 3-3 shows an inference network that was directly translated from the rules in Figure 3-2. Note that at the intersection of each line an arc is drawn. The arc represents the AND operator and, although not included in this example, the absence of arc implies an OR operator. Because the relationship between each rule is known ahead of time, inference networks only need to execute the rules directly connected to facts and rules that have been satisfied. This makes inference networks significantly more efficient than the exhaustive search used by the pattern matching systems discussed above. The drawback is that inference networks are often impractical or unfeasible for complex systems with a large number of interacting rules.

Although Figure 3-3 is correct for the rules that Doug generated in the example above, it should be noted that the inference network does not contain all of the information available for each cat shown in Table 3-2. Should a second solid colored cat be added to the knowledge base, the inference network as drawn would require modification. To allow for expansion of the knowledge base, it may be better to include all of the information available (Figure 3-4).

Unfortunately, this results in a loss of efficiency, since the system would require all three of Panther's characteristics to identify him.

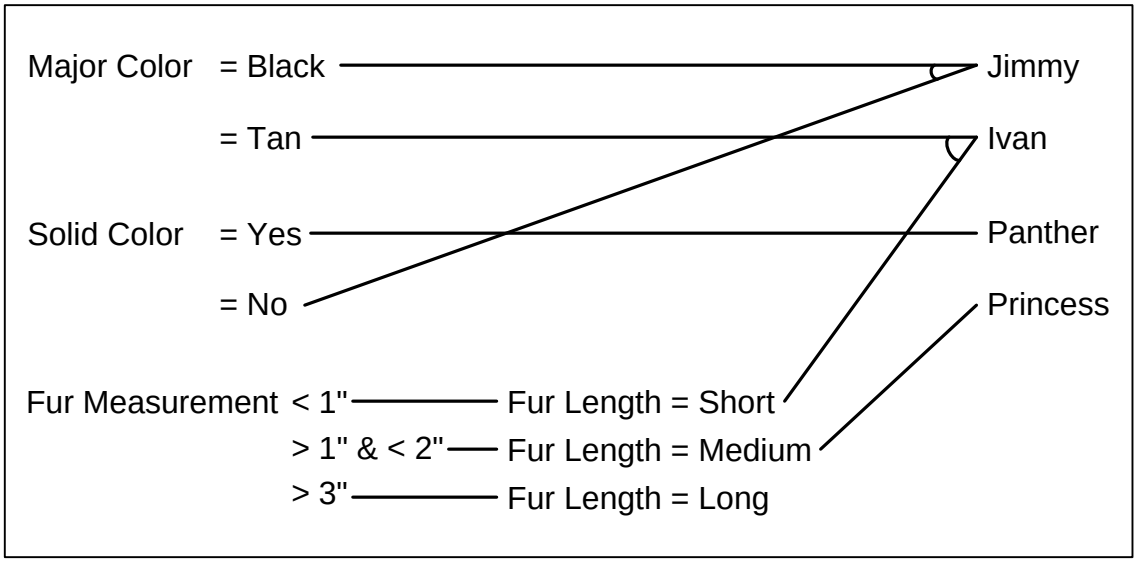

Figure 3-3. Inference network for Doug's cats based on rules in Figure 3-2

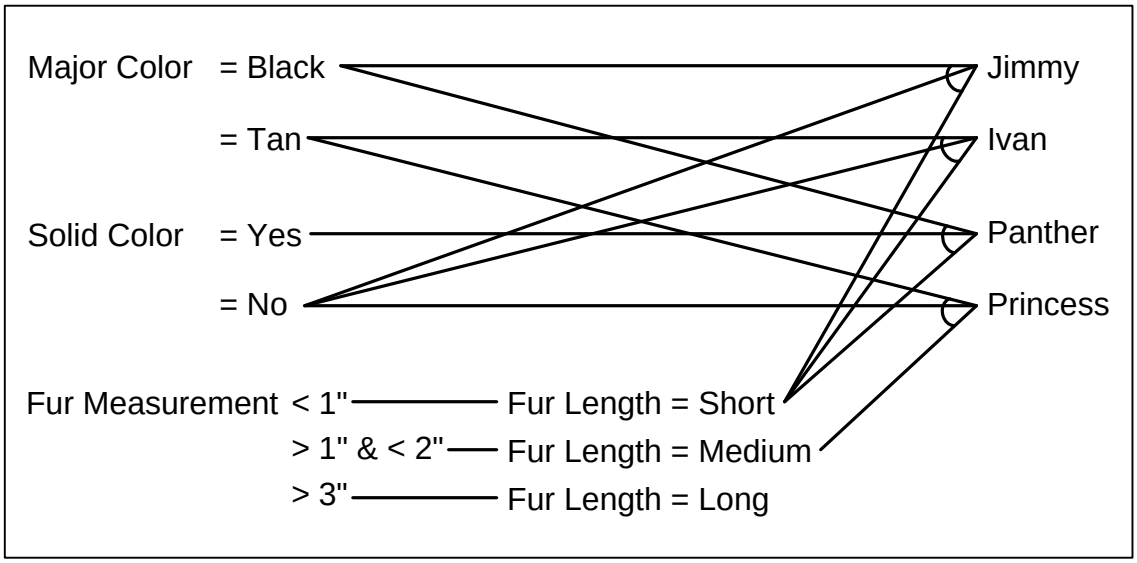

Figure 3-4. Exhaustive inference network for Doug's cats

Decision Trees

A decision tree is a specialized form of inference network that arranges inputs in a hierarchical fashion. Rather than utilizing every input available from the beginning, decision trees only address one input at a time. Once that input has been assigned a value, the next input

in the tree is addressed until enough information has been gathered to offer a solution. Figure 3-5 displays an example decision tree for Doug and his cats. As shown, the system first asks the user for the cat's major color. After receiving an input, it asks whether the cat is a solid color or not. Given a black cat, the system can offer a solution after two questions. Tan cats, however, require three questions to identify. Note that if the user informs the system that the cat is tan and solid colored, a null set is reached, causing the system to output an error message or backtrack in an attempt to find a solution. It is also important to understand that the decision point for solid color need not be located at the same depth on every branch of the tree.

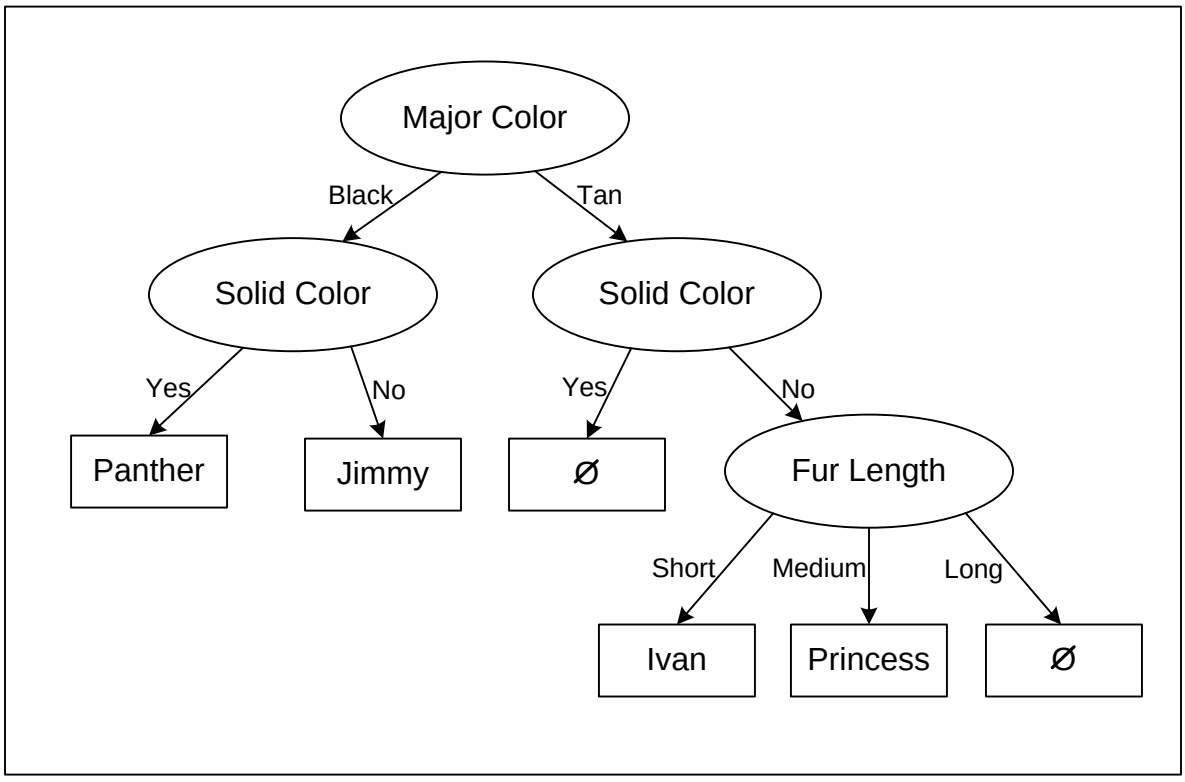

Figure 3-5. Decision tree for identifying Doug's cats

The ordering of the decision tree in Figure 3-5 was assigned in an arbitrary manner; however, there exist mathematical approaches based on information theory that seek to minimize the number of branches necessary to solve a problem. The most widely recognized approach is known as the ID3 algorithm, which was created by J. Ross Quinlan and is discussed in detail in

Gonzalez and Dankel (1993). More recently, a descendent of the C4.5 algorithm, also created by Quinlan, has become popular (Quinlan, 1996). Other variations of decision trees enable the system to handle uncertainty. One such approach, implemented by Althoff et al. (1998), adds an extra branch to encompass uncertainty. For example, the node requesting a major color would not only include the responses black and tan, but also a branch representing that the color could not be determined. Certainty factors, discussed in the following section, are another method for handling uncertainty in rule-based systems.

Certainty Factors

The use of certainty factors is one of the oldest and most established methods of handling uncertainty in rule-based systems. Certainty factors were originally created for use in MYCIN, an expert system for the treatment of infectious blood diseases (Buchanan & Shortliffe, 1984a). MYCIN was the first medical expert system and is discussed in more detail in Chapter 4.

Certainty factor values range from -1 to 1, where -1 represents a statement being false, 1 represents a statement being true, and 0 represents complete uncertainty whether the statement is false or true. Each rule is assigned a certainty factor (CF) that represents confidence that a statement is true or false. The general form of a rule with a certainty factor is:

If <Evidence>,
Then <Hypothesis> (CF),

where <Evidence> is the premise containing the observed or derived facts and <Hypothesis> is the conclusion that results from satisfying the premise. CF represents the confidence that the hypothesis is correct, given the evidence.

Certainty factors can be assigned in a number of ways. Some may be assigned subjectively by asking an expert to assign a value of confidence to a rule based on past experience. Others may be determined by using probability to calculate a measure of belief and

disbelief, then mathematically combining the results to yield a certainty factor. Regardless of how they are determined, the math for combining and propagating certainty factors is as follows (Buchanan & Shortliffe, 1984b):

$$CF_{revised} = CF_{old} + CF_{new} * (1 - CF_{old}), \quad \text{if } CF_{old} \text{ and } CF_{new} \geq 0, \quad (3\text{-}1)$$

$$CF_{revised} = -(-CF_{old} - CF_{new} * (1 + CF_{old})), \quad \text{if } CF_{old} \text{ and } CF_{new} \leq 0, \quad (3\text{-}2)$$

$$CF_{revised} = \frac{CF_{old} + CF_{new}}{1 - \min(|CF_{old}|, |CF_{new}|)}, \quad \text{if } CF_{old} \text{ XOR } CF_{new} < 0. \quad (3\text{-}3)$$

These equations assume that a rule's premises are known with absolute certainty. Unfortunately, such an assumption is often false because premises also have associated certainty factors. To handle situations where the evidence itself may be uncertain, the following rules of combination are used (Gonzalez & Dankel, 1993):

1. A rule with a single uncertain premise yields a CF that is the product of the conclusion's CF and the premise's CF.
2. A rule with a conjunction of uncertain premises yields a CF that is the product of the conclusion's CF and the minimum CF of all the premises.
3. A rule with a disjunction of uncertain premises yields a CF that is the product of the conclusion's CF and the maximum CF of all the premises.

To better understand the use of these equations and rules, let us consider an example from driver training. In driver training, students are told that they should scan ahead for dangerous driving situations. When they identify a possibly dangerous situation, they should predict what might occur. At this point, they should decide what to do and execute their planned course of action. Let us imagine that Steve is an android that has been created to function as a normal human being in society. One day, he is riding his motorcycle along a narrow side street where a car is parked on the left hand side. In general, he would want to stay farther to the right of the street to minimize the danger of an unseen child running out from behind the car or a person in the car opening a door into his driving path. As Steve nears the car, however, he notices a young

boy on the right hand side of the street. He must now decide whether the parked car or the child is more likely to introduce a dangerous driving situation. Surveying the situation, Steve notices that the child's mother is present, but appears preoccupied with her gardening. The child is playing with a ball and seems completely oblivious to the approach of the motorcycle. Steve's computer brain begins processing this information by assigning certainty factors to his observations. He is confident that the child is playing with a ball and that the mother is present, so he assigns these premises a CF = 0.9. He is only half certain that the mother is inattentive, so he assigns this a CF = 0.5. He is fairly certain that the child has not noticed the vehicle, so he assigns that a CF = 0.7. Steve recalls the rules of thumb that he learned from similar driving experiences:

Rule 1: If the child is playing with a ball,
then the child will run out into the street. (CF = 0.6)

Rule 2: If the child's parent is present AND not attentive,
then the child will run out into the street. (CF = -0.3)

Rule 3: If the child has not noticed the vehicle,
then the child will run out into the street. (CF = 0.7)

Steve initializes his CF to zero and starts at Rule 1. Based on the first rule of combination, he combines the certainty factors of the premise (CF = 0.9) and the conclusion (CF = 0.6) by multiplying to obtain a CF = 0.54. He then implements Equation 3-1 to calculate his new certainty factor:

$$CF_{revised} = 0 + 0.54 * (1 - 0) = 0.54 . \qquad (3\text{-}4)$$

Next, Steve moves to Rule 2. Using the second rule of combination, he selects the lower of the two CF's for the premise (CF = 0.5) and multiplies it with the CF for the conclusion (CF = -0.3) to obtain a CF = -0.15. He then combines the current CF of 0.54 with the new CF using Equation 3-3:

$$CF_{revised} = \frac{0.54 - 0.15}{1 - \min(|0.54|, |-0.15|)} = 0.46 . \quad (3\text{-}5)$$

Finally, Steve addresses Rule 3. Using the first rule of combination, he combines the certainty factors of the premise (CF = 0.7) and the conclusion (CF = 0.7) by multiplying them to get CF = 0.49. Using Equation 3-1, he combines the current CF of 0.46 with the new CF to obtain his answer:

$$CF_{final} = 0.46 + 0.49 * (1 - 0.46) = 0.72 . \quad (3\text{-}6)$$

So, Steve has a certainty factor of 0.72 that the child will run out into the street in front of him. Since this is a fairly high value, Steve chooses to move away from the right hand side of the street where the child is and closer to the car on the left hand side.

Case-Based Reasoning

Case-based reasoning is the method of using documented examples and their solutions to solve problems. Unlike traditional methods where the system designer must generate a cohesive set of rules that yield the correct answer, case-based reasoning systems generally maintain a database of examples, or cases, that are used to solve a problem. The basic structure of a case-based reasoning system consists of a library or database of historical cases, a way to retrieve similar cases from the library, and a way to modify the solutions if the retrieved case is not identical to the problem. When a problem is introduced to a case-based reasoning system, the system searches its database for cases with similar attributes. When a sufficient number of similar cases are discovered, the solutions to these cases are then combined and modified to better match the problem being solved before the final solution is presented to the user. Greater detail on case-based reasoning can be found in Kolodner (1993).

To better understand case-based reasoning, let us examine an example adapted from Gonzalez and Dankel (1993). Alice is considering selling her home in Wonderland; however,

she is unsure how much her house is worth. Searching the Internet, Alice discovers an online case-based reasoning system designed to calculate the current market value of houses based on region. The system accomplishes this by keeping a record of all the recent home sales by region as well as basic information on each house, including the square footage, number of bedrooms, number of bathrooms, and whether the house has a pool or not. Alice tells the system that her house has 1500 square feet, 3 bedrooms, 3 bathrooms, and a pool. The system proceeds to look up all the property sales in Wonderland with comparable size and characteristics to Alice's house. At the end of the system search, the top five most similar houses are selected (Table 3-4). The system selects the first house as most similar to Alice's house, giving it a starting value of $150,000. The system now seeks to adapt the house value to include the value of Alice's pool. The primary difference between the second and fourth house is that the second house has a pool and the fourth house does not. The system takes the difference between the values of the two houses and determines that a pool is worth approximately $45,000. The same operation is performed for the third and fifth houses, yielding an approximate value of $65,000 for the pool. Averaging these two values, the system determines that Alice's pool is worth approximately $55,000. Adding the pool's value to the starting value, the system approximates the value of Alice's house to be $205,000.

Table 3-4. Houses most similar to Alice's house

House ID	Price	Square footage	# of bedrooms	# of bathrooms	Pool?
1	$150,000	1470	3	3	No
2	$225,000	1540	4	3	Yes
3	$180,000	1480	3	2	Yes
4	$180,000	1520	4	3	No
5	$115,000	1460	3	2	No

There are some distinct advantages to case-based reasoning. First, case-based reasoning bypasses the bottleneck of gathering information from experts and converting them into rules.

Second, case-based reasoning can be used in fields where examples abound, but the fundamental principles are not well understood. As long as there are sufficient cases that the system can access, the system can still function. The drawback is that without a well documented set of available cases a case-based approach cannot be implemented. Much research has been performed in the area of case-based reasoning in recent years. For a summary of this research refer to Nilsson and Sollenborn (2004). For an example of a case-based reasoning applied to the medical field, refer to papers on CASEY, a system for diagnosing heart failure (Koton, 1988). Also, see Althoff et al. (1998) for a case-based reasoning system directly applied to the field of toxicology.

Nearest Neighbor Approaches

Another common approach for solving classification problems is the use of nearest neighbor methods. Nearest neighbor methods require a number of training samples with characteristics that have been parametrized to create a numerical vector. Each vector can be thought of as the point in *n*-dimensional solution space that is occupied by the sample, where *n* is the number of parameters in the vector. Once all the training samples are situated in the solution space, a clustering algorithm is used to label classification regions within the space as corresponding to a specific class of objects. When an unknown object is introduced to the system, the system parametrizes the object by creating a vector of the same form as the training samples. These vector coordinates are used to calculate the distance between the object's vector and each of the classification regions. Finally, the unknown object is classified based on the label of the nearest region (Han & Kamber, 2001).

Let us examine a simple nearest neighbor system designed to identify sports balls at a recreation center. Each ball is classified based on size and color. The system could use the diameter of the ball in centimeters as its size value; however, our system will simplify the

problem by identifying balls as small, medium, or large with the corresponding values of 1, 2, and 3 respectively. Since some colors might be misidentified by certain people, it is important to assign similar colors with consecutive numbers. For this reason, the system will use the order of the visible light spectrum in numbering the colors where red = 1, orange = 2, yellow = 3, green = 4, blue = 5, indigo = 6, and violet = 7. Table 3-5 shows six types of balls that the recreation center stocks along with their sizes, colors, and the corresponding values. (For readers unfamiliar with indoor soccer balls, simply imagine a tennis ball the size of a standard soccer ball.) In the final column of the table, the vectors containing size and color values are shown. This vector situates each ball as a point in the domain's 2-dimensional solution space (Figure 3-6).

Table 3-5. Characteristics of various sports balls

Type	Size	Size value	Color	Color value	Vector
Tennis ball	Small	1	Green	4	(1,4)
Racket ball	Small	1	Blue	5	(1,5)
Water polo ball	Medium	2	Yellow	3	(2,3)
Indoor soccer ball	Medium	2	Green	4	(2,4)
Four square ball	Large	3	Red	1	(3,1)
Basketball	Large	3	Orange	2	(3,2)

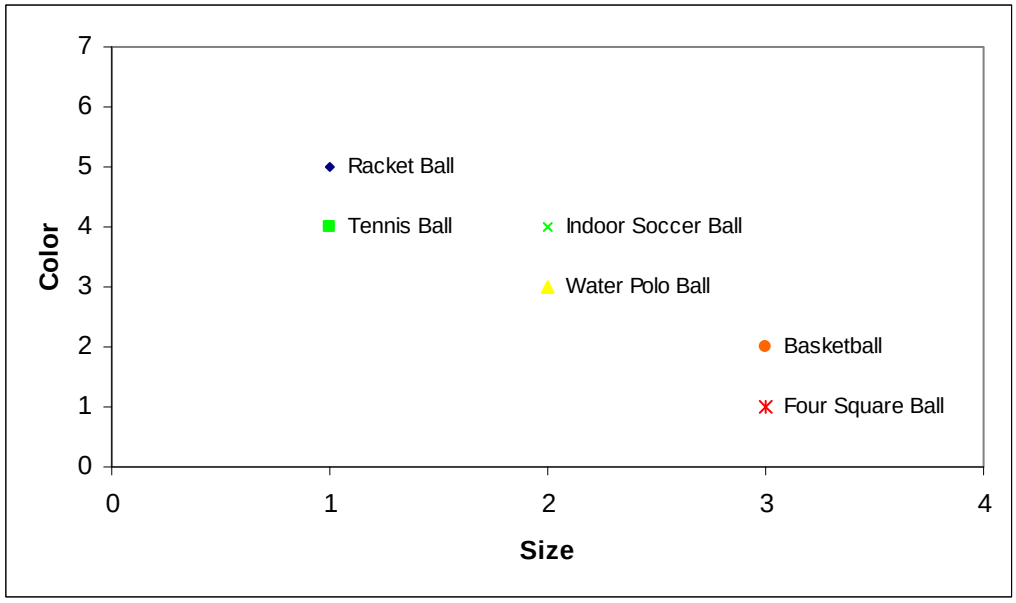

Figure 3-6. Vectors for sports balls plotted in 2-dimensional solution space

If one of the workers at the recreation center were trying to identify an unknown ball that he describes as a "large, green ball," he could input the description into the system. For this example, let us assume that the system uses Euclidean distance to determine the nearest neighbor match. Euclidean distance from point *p* to point *q* is defined as:

$$d(p,q) = \sqrt{\left|x_{p1} - x_{q1}\right|^2 + \left|x_{p2} - x_{q2}\right|^2 + ... + \left|x_{pn} - x_{qn}\right|^2}, \quad (3\text{-}7)$$

where $p = (x_{p1}, x_{p2}, ..., x_{pn})$ and $q = (x_{q1}, x_{q2}, ..., x_{qn})$ are *n*-dimensional vectors that define the object. To identify the "large, green ball," the ball's description must be parametrized. Looking at the definitions above, we can see that large is defined as 3 and green is defined as 4 giving the ball a vector of (3, 4). Next, the distance between the unknown ball and each ball type within the system must be measured as follows:

$$d(p,q) = \sqrt{\left|x_{pSize} - x_{qSize}\right|^2 + \left|x_{pColor} - x_{qColor}\right|^2}, \quad (3\text{-}8)$$

where *p* represents the unknown ball and *q* represents a ball recorded in the system. Calculating the distance to each ball in the system yields:

$$d(p, TennisBall) = \sqrt{\left|3-1\right|^2 + \left|4-4\right|^2} = 2, \quad (3\text{-}9)$$

$$d(p, RacketBall) = \sqrt{\left|3-1\right|^2 + \left|4-5\right|^2} = 2.236, \quad (3\text{-}10)$$

$$d(p, WaterPoloBall) = \sqrt{\left|3-2\right|^2 + \left|4-3\right|^2} = 1.414, \quad (3\text{-}11)$$

$$d(p, IndoorSoccerBall) = \sqrt{\left|3-2\right|^2 + \left|4-4\right|^2} = 1, \quad (3\text{-}12)$$

$$d(p, FourSquareBall) = \sqrt{\left|3-3\right|^2 + \left|4-1\right|^2} = 3, \quad (3\text{-}13)$$

$$d(p, BasketBall) = \sqrt{\left|3-3\right|^2 + \left|4-2\right|^2} = 2. \quad (3\text{-}14)$$

From these results, the ball with the smallest Euclidean distance is selected as the nearest neighbor. Namely, the unknown "large, green ball" is identified as an indoor soccer ball.

The example above is a simplistic system. It is extremely limited in size and represents only one approach to solving classification problems using the nearest neighbor method. In

reality, there is no limit, other than computational power, to the number of parameters that can be included in a vector. Characteristic parameters are not limited to linear mathematical values. They can also include non-linear values, binary values, nominal variables with multiple states, and many other representations. The Euclidean distance is only one of many methods for determining the nearest neighbor and may not always be appropriate for some parameters. Although our system did not use a clustering algorithm, most real world systems do. One way to incorporate a clustering algorithm into the sports ball recognition system would involve taking many samples of each ball type and entering them into the system. Perhaps only some tennis balls are green, while others are yellow or orange. Some systems might take a sampling of all of these types of tennis balls and then calculate the centroid as the point to which distance should be calculated. More complex systems might define multiple points or a region to represent the tennis ball. For more general information on nearest neighbor methods and clustering algorithms, consult Han and Kamber (2001). Bradley et al. (1998) discuss the scaling of clustering algorithms to handle large databases.

Bayes' Rule

Bayes' rule is perhaps the most widely known and implemented technique for uncertainty management. Given certain knowledge, it enables the user to identify and select the most likely solution through the use of probability theory. Bayes' rule is defined as:

$$p(y \mid x) = \frac{p(x \mid y) * p(y)}{p(x)}, \qquad (3\text{-}15)$$

where p(x) and p(y) are the probabilities of events x and y occurring, respectively. The probability of event x occurring, given that event y has occurred, is represented by p(x | y). Likewise, the probability of event y occurring, given that event x has occurred, is represented as p(y | x) (Duda et al., 2001).

To put this equation in perspective, let us look at an example. The University of Florida campus includes a lake called Lake Alice. The lake is known to have alligators. For this reason, many people go to Lake Alice in the hopes of seeing an alligator in the wild. Many birds also inhabit the regions of Lake Alice, including ducks. Unlike humans, who may have a hard time locating an alligator, ducks are more aware of their surroundings and tend to steer clear of areas where an alligator is present. By gathering data from many visits to Lake Alice, it has been determined that the probability of seeing a duck, p(duck), is 0.8, and the probability of seeing an alligator, p(gator), is 0.4. Since ducks avoid alligators, the probability of seeing a duck given that an alligator is present, p(duck | gator), is only 0.2. With this knowledge, we visit the lake in an attempt to locate an alligator. Looking around, we notice that there are ducks present, so we use Bayes' rule to calculate the probability that an alligator is present:

$$p(gator \mid duck) = \frac{p(duck \mid gator) * p(gator)}{p(duck)} = \frac{0.2 * 0.4}{0.8} = 0.1. \qquad (3\text{-}16)$$

We find that the probability of an alligator being present is only one in ten, so we probably should come back to look for alligators on a different day.

Bayes' rule has a distinct advantage over many other methods in that it has the support of well established mathematical theory. Bayes' rule is limited, however, in that it assumes that all observations are mutually independent. Unfortunately, this is not the case in the real world. One proposed solution to this problem is the use of Bayesian belief networks, which are discussed below in the section entitled "Modern Approaches for Diagnosing Multiple Disorders." In spite of its limitations, Bayes' rule can be applied effectively in many situations and can be expanded to include the probability of many events. For further information on Bayes' rule, including its derivation, refer to Duda et al. (2001).

Other Approaches to Knowledge-Based Systems

Fuzzy Logic

In 1965, Lotfi Zadeh wrote a paper introducing "fuzzy sets" to the world. This paper was the birth of fuzzy logic. Fuzzy logic is an advanced form of Boolean algebra that allows partial membership within different sets or categories. Boolean variables can only be absolutely true, represented by a 1, or absolutely false, represented by a 0. Fuzzy logic, however, allows variables to be partially true and partially false, represented by any value between 0 and 1.

In a normal Boolean representation, a person can be tall or not tall. The problem with this representation is that there must be a clean cutoff for where tall begins and ends. If 6' were set as the cutoff for being tall, someone with a height of 5'11.9" would be considered not tall. Such differentiation does not fully represent the world in which we live because our world is not discrete. To compound the problem, human beings often think and speak using general, imprecise language where the characteristics of things, such as tallness, are subjective in nature. Fuzzy logic is an attempt to capture the meaning of the imprecise, or fuzzy, statements inherent to human thinking and represent them in a manner that enables a system to solve problems.

Figure 3-7 shows a fuzzy logic graph with four sets: midget, short, tall, and giant. The graph shows that the set(s) to which a person belongs have memberships ranging from 0 to 1 depending on the height of the individual. Based on the graph, a person with a height of 5' is considered fully short with a value of 1 and belongs to no other sets because they all have a membership value of 0 at 5'. Likewise, a person with a height of 6' is tall, a person shorter than 4' is a midget, and a person taller than 7' is a giant with no membership in any other sets. What happens between these heights demonstrates the difference between fuzzy logic and Boolean algebra. If Joann is 5'6" tall, as shown on the chart, she has a 0.5 membership in both short and tall sets. Likewise, Ryan, who is 6'9", has a membership of 0.25 in tall and 0.75 in giant.

Although Figure 3-7 is drawn with linear slopes, any variety of functions may be used. Furthermore, it is not necessary for memberships to add up to 1. If a medium height membership were created with its peak at 5'6", Joann would then have a membership of 1 in the medium set in addition to her 0.5 membership in both short and tall sets.

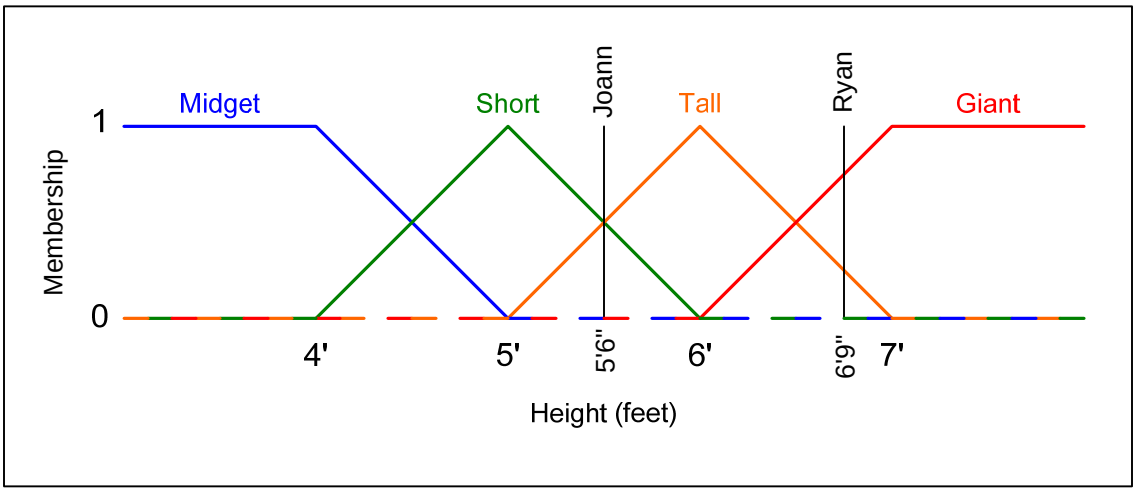

Figure 3-7. Fuzzy logic graph for human heights

Fuzzy logic uses a modified version of Boolean operators to perform its operations. The fundamental operators are as follows:

1. Compliment: NOT(A) = 1-A,
2. Union: A OR B = Max(A,B),
3. Intersection: A AND B = Min(A,B).

Other operators have been created to mimic human language, such as using A^2 to represent "very" and $\sqrt{A}$ to represent "more or less" functions.

Due to its imprecise nature and lack of mathematical proofs, fuzzy logic has many opponents in the technical world. In spite of this, it has been successfully implemented in a variety of fields including data mining and knowledge-based systems. Recent research in applying fuzzy logic to data mining includes a system by Delgado et al. (2000) to mine medical databases, Au and Chan's (2003) system for mining rules from a large banking database, and Wang's (2003) system for generalized data mining. Liu and Yan (1997) have also created a

system that combines fuzzy networks and case-based reasoning to solve diagnostic problems. For more general information on fuzzy logic, consult Gonzalez and Dankel (1993).

Dempster-Schafer

The Dempster-Schafer theory was developed by Arthur Dempster and Glenn Schafer in 1967. Although rarely used in practice due to its high computational requirements, the Dempster-Schafer theory is one of the classic approaches to handling uncertainty in knowledge-based systems. Dempster-Schafer is unique in that it gives confidence values to sets, rather than solely individual facts, and is capable of representing our "certainty about certainty" (Gonzalez & Dankel, 1993, p. 253). For more information regarding the Dempster-Schafer method, refer to Gonzalez and Dankel (1993) or the original paper by Arthur Dempster (1967).

Rough Sets

Rough set theory was proposed for the field of data mining by Zdzislaw Pawlak in 1982. Rough sets are formed by examining the data available to the system and identifying any extraneous feature points that are not necessary for differentiating between cases. These extraneous features are then removed and the remaining features form a construct called a reduct that is used for classification or identification of unknowns (Kusiak et al., 2000). Although computationally effective to use reducts, users of expert systems can be "reluctant to make decisions based on the minimum number of features, rather they would like to see the same decision reached by alternative sets of features" (Kusiak et al., 2001, p. 225). For this reason, the use of rough sets can be abhorrent to doctors that feel the more information used in making a decision, the better.

Kusiak et al. (2000) offer an excellent short summary of rough sets as it relates to their research in diagnosing and treating lung abnormalities called solitary pulmonary nodules. Kusiak et al. (2001) present an algorithm based around rough sets for extracting rules relating to

heart arrhythmia. Tsumoto (2000) also presents a rough set approach to diagnosing diseases. To increase system accuracy, Tsumoto's system creates both positive rules to "rule in" and negative rules to "rule out" possible diseases.

Genetic Algorithms

John Holland introduced the idea of genetic algorithms in 1975. The philosophy behind genetic algorithms is to model natural selection in nature. In short, natural selection states that the organisms with the most advantageous genes for survival tend to pass their genetics on to the next generation of organisms, while those with inferior genes tend to die before they reproduce. Through this process, the offspring of a species gradually become fitter and more capable of survival.

In the same way that the DNA of a species is divided into chromosomes, genetic algorithms are made of building blocks of code, called primitives. These primitives are the smallest functional units of code and cannot be separated. By randomly assembling algorithms from primitives, the first generation of algorithms is created. Each of these algorithms is then evaluated by a fitness function to quantify its performance, or fitness. The fitness of each algorithm is used to determine the probability that the algorithm is selected to contribute to the next generation of algorithms. The better the algorithm's fitness, the more likely it is to be selected. There are three ways that an algorithm can contribute to the next generation: reproduction, crossover, and mutation. For each of these methods, a certain percentage of the current generation of algorithms is randomly selected. The algorithms selected for reproduction are copied directly to the next generation without modification. The algorithms selected for crossover are paired with a second algorithm. Both algorithms are broken at a random location in their code. The segments are then exchanged between the algorithms so that each algorithm has a piece of the other's code and these newly modified algorithms become a part of the next

generation. Algorithms chosen for mutation have a random segment of code deleted from their programming and replaced by another randomly generated set of code. Once all of this has occurred, the algorithms selected for reproduction, crossover, and mutation are all compiled to become a new generation of algorithms. Like their predecessors, the new generation will be evaluated by a fitness function and then some are selected to reproduce, crossover, or mutate to create the next generation of algorithms (Nilsson, 1998).

The process of creating a useful algorithm using the genetic algorithm approach takes thousands to millions of iterations to complete and may never fully optimize the algorithm's code. Due to their randomness, however, genetic algorithms have great utility in optimization problems because they are much less likely to converge on local maxima or minima. Vinterbo and Ohno-Machado (2000) have also applied genetic algorithms to the problem of diagnosing multiple disorders. For more general information on genetic algorithms, refer to Nilsson (1998).

Artificial Neural Networks

Like genetic algorithms, artificial neural networks were inspired by nature. Artificial neural networks are composed of units that roughly approximate the firing of a neuron in a biological organism. In a biological organism, a neuron sits inactive until it is stimulated beyond its activation threshold. When this occurs, the neuron fires, sending its signal to the brain. The firing of a neuron is an all or nothing response. There is no variability in the signal it sends. In the same way, units in artificial neural networks can be thought of as having an activation threshold that turns on when its inputs exceed a certain threshold and has a value of zero at all other times. In practice, the units employ a differentiable function, like a sigmoid, to approximate a step response. Differentiability is important in the training of a neural network, which is discussed below.

The most common artificial neural network is known as a feedforward multilayer perceptron. This type of neural network consists of an input layer, an output layer, and any number of hidden layers. Figure 3-8 shows an example neural network with two hidden layers. In creating a neural network, the designer must determine the inputs to the system as well as how many outputs are necessary to solve the problem. The designer must also select the number of hidden layers and the number of units to be included per layer. As seen in the figure, inputs are directly connected to every unit in the first hidden layer. Likewise, every unit in the first hidden layer is connected to every unit in the second layer and so on, all the way through the output layer. Each connection has an associated weight that acts as a multiplier. By adjusting the weights, the influence an input has on a specific unit can be controlled. The weights of a neural network are usually initialized randomly with small values. A bias unit, that always outputs 1, is also included at every layer. Adjusting the weight between the bias unit and another unit shifts the activation threshold of that unit (Nechyba, 2003).

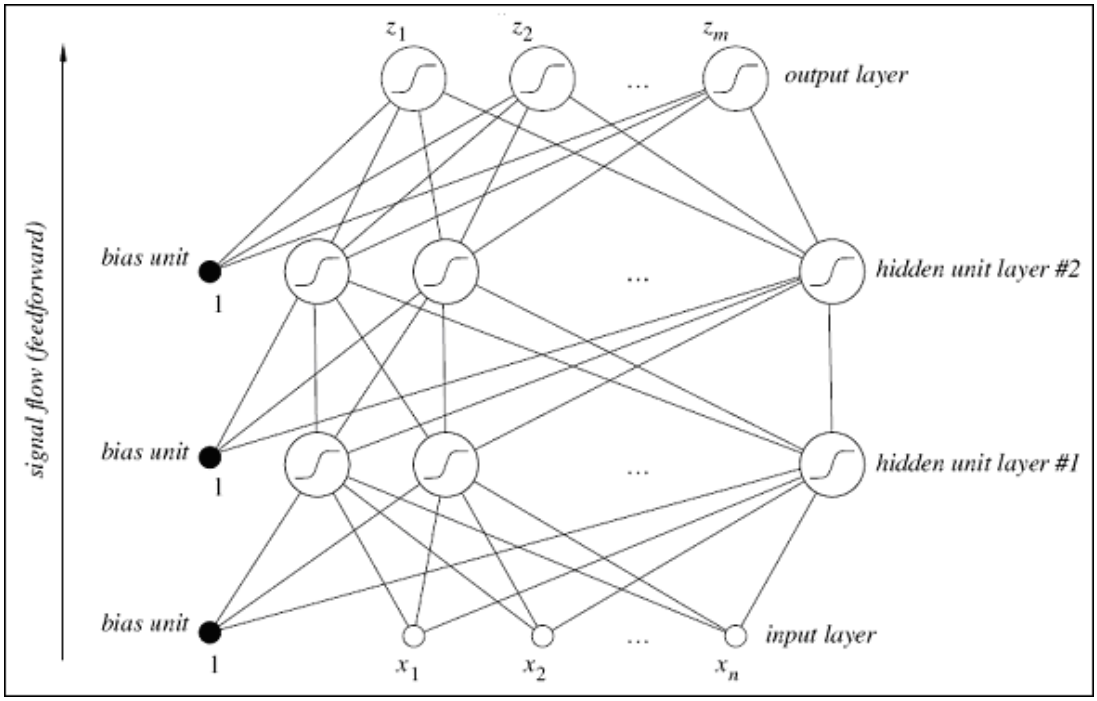

Figure 3-8. Typical artificial neural network with two hidden layers. Figure used with permission from Nechyba (2003, p. 7).

Training a neural network to solve problems involves adjusting the weights for the connections between each unit. The most well known algorithm for adjusting weights is the backpropagation algorithm, published in 1986 by Rumelhart and McClelland. By introducing an input with a known solution to the neural network, the difference between the output of the system and the desired solution output can be compared. The backpropagation algorithm is then used to adjust the weights accordingly. By repeating this process many times with a variety of samples, the weights gradually converge to a local minima in an attempt to maximize the number of samples the system can correctly identify.

Artificial neural networks are non-linear function approximators. Their strength lies in their ability to train themselves from sample cases, however, this is also their weakness. Because of the complexity of the network itself, it is hard to understand and explain the internal workings of a trained neural network. Abidi and Manickam (2002) have created a hybrid system using case-based reasoning and neural networks to data mine medical systems. For a general discussion on neural networks, refer to Nilsson (1998).

Modern Approaches for Diagnosing Multiple Disorders

Over the years, researchers have implemented a variety of methods, including those discussed above, in an attempt to diagnose problems involving multiple disorders. In most cases, linearity and statistical independence cannot be assumed in problems of this nature. For this reason, the challenge of efficiently and effectively diagnosing multiple disorders remains an important area of research today. In recent years, two problem solving methods appear to be taking the forefront. The first involves the modification of Bayesian methods to account for dependencies. The second involves the use of set theory to approach the problem.

Bayesian Belief Networks

As discussed above, Bayes' rule requires statistical independence of events to solve problems. Over the years, many variations using Bayesian methods have been developed to account for dependencies in a data set. In this section, we discuss perhaps the best documented of these approaches, namely Bayesian belief networks.

Bayesian belief networks allow dependencies to be included in a system's probability calculations. Figure 3-9 displays a graphical representation of a belief network. As shown, it can be seen that belief networks consist of two parts, a directed acyclic graph and conditional probability tables (Han & Kamber, 2001). The graph portion consists of nodes, which represent random events, and arcs, which portray statistical dependencies between nodes. The example in Figure 3-9 contains six random events, including DarkClouds, Humidity, and Rain. Arcs are drawn between DarkClouds and Rain as well as Humidity and Rain to show that the presence of DarkClouds and/or Humidity influences the likelihood of Rain. A conditional probability table is drawn to the right of the graph. Belief networks have one table for every node in the graph, however, only the table for the Rain node is given in this example. The table shows the various probabilities for the occurrence of Rain given the presence or absence of Rain's parents, DarkClouds and Humidity. Represented mathematically, the first column of the table states that:

$$P(Rain = True \mid DarkClouds = True, Humidity = True) = 0.9, \quad (3\text{-}17)$$

$$P(Rain = False \mid DarkClouds = True, Humidity = True) = 0.1. \quad (3\text{-}18)$$

There are many ways to train a Bayesian belief network. If the structure of the network is known and the events are observable, conditional probability tables can be calculated using standard probability and statistics calculations. If the structure is known but not all of the events are observable, a gradient descent method can be used to determine a local optimum of

probabilities. For more general information on the structure and generation of Bayesian belief networks, refer to Han & Kamber (2001).

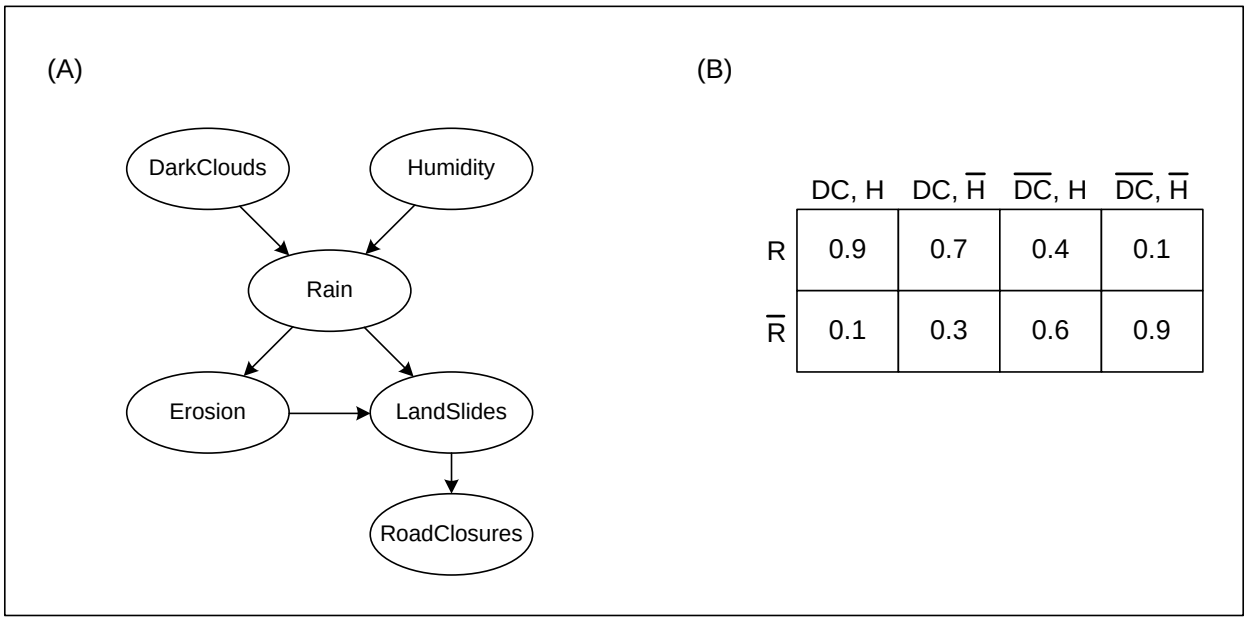

Figure 3-9. Example Bayesian belief network. (A) Directed acyclic graph of dependencies. (B) Conditional probability table for Rain, where R = Rain, DC = DarkClouds, and H = Humidity.

The application of modified Bayesian methods is one of the most promising areas of research in diagnosing multiple disorders. Bayesian belief networks have been used by van der Gaag and Wessels (1994) in an attempt to efficiently diagnose multiple disorders. The HEPAR II system, by Onisko et al. (2000, 2001), also uses belief networks to diagnose multiple disorders in the field of hepatology. Other Bayesian variations exist, including Peng and Reggia's (1989) use of a "comfort measure" that attempts to adapt Bayes' rule to the diagnosis of multiple disorders. Additional research by Peng and Reggia (1986, 1987) includes the creation of a hybrid system, combining both Bayesian classification techniques and the set covering model. The next section provides an introduction to set covering.

Set Covering

Another approach to diagnosing multiple disorders is the use of set covering theory. Given a case with a set of observed symptoms, set covering seeks to construct a solution set of disorders that can best account for the symptoms. In generating solution sets, it is not uncommon for several plausible problem solutions to exist. In such cases, the principle of Occam's Razor is generally applied, meaning that the simplest explanation is usually the best. For this reason, set covering is also known as the parsimonious covering theory (Peng & Reggia, 1986).

To understand set covering, we must begin by formally defining three universal sets:

1. $D = \{d_1, d_2, ..., d_m\}$, where D is the set containing every possible disorder, d,
2. $S = \{s_1, s_2, ..., s_n\}$, where S is the set containing every possible symptom, s,
3. $R = \{r_1, r_2, ..., r_p\}$, where R is the set containing every possible relationship, r.

The relationships within set R are tuples consisting of a disorder and a corresponding symptom such that $r_k = (d_i, s_j)$, where s_j is a symptom that may be caused by disorder d_i. It is important to note that disorder d_i does not always result in symptom s_j. Likewise, symptom s_j may be caused by a disorder other than d_i (Peng & Reggia, 1986).

Generate-and-test is the simplest algorithm for solving cases involving set covering. To implement the algorithm correctly, three more sets must be defined:

4. $F_O \subseteq S$, where F_O contains the observed symptoms for a particular case,
5. $H \subseteq D$, where H is the hypothesized disorder set that may be responsible for F_O,
6. $F_H \subseteq S$, where F_H contains the symptoms associated with H by set R.

When a set of observed symptoms is presented to the system for diagnosis, each of these symptoms is stored in F_O. The system then generates hypothetical sets of H to determine the disorders that could be causing the observed symptoms. When a hypothetical set H is generated, F_H is populated by all of the symptoms that can be caused by the proposed disorders in H. F_H is

then compared with F_O to determine if F_H "covers," or contains, every symptom in set F_O. If F_H covers F_O, the solution is a plausible solution. As stated above, set covering systems generally follow the principle of Occam's Razor. For this reason, generate-and-test algorithms usually begin with sets of H consisting of single disorders. If no suitable solution is found, the system then considers double exposure cases followed by complex multiple exposure cases as necessary (Baumeister et al., 2001).

Figure 3-10 presents a graphical representation of the relationships, R, between disorders, D, and symptoms, S. Let us assume that that all five symptoms are presented to the system as observed findings, F_O. Using a simple generate-and-test algorithm, the system checks every individual disorder, d, in an attempt to find a single disorder solution to the problem. As seen from the graph, no single disorder can satisfy the observed symptoms. Then, the system checks for multiple disorders capable of covering the symptoms contained in F_O. It should be apparent that there exist at least three solution sets to the problem, (d_2, d_3), (d_1, d_3, d_4), and (d_1, d_4, d_5). Using the heuristic of minimality, the system would select (d_2, d_3) as the solution. At times, however, minimality will not yield the best solution to a problem. For example, if we were aware that disorder d_2 was an extremely rare disease and highly unlikely to occur, either of the other two possible solutions might provide a better answer. Furthermore, in comparing (d_1, d_3, d_4) to (d_1, d_4, d_5), we can see that the former results in more redundancy due to the presence of d_3 in the solution. If redundancy is considered a negative aspect in a solution, the system should ultimately select (d_1, d_4, d_5) as the solution set. From this example, it can be seen that the principle of parsimony and Occam's Razor can be applied in many different ways and, although favoring the smallest solution set is often a useful heuristic, there are many instances where this approach does not yield the best result. For more information on the nature of

parsimony, refer to Peng & Reggia (1986). Atzmueller et al. (2004a) also address this topic from a different perspective.

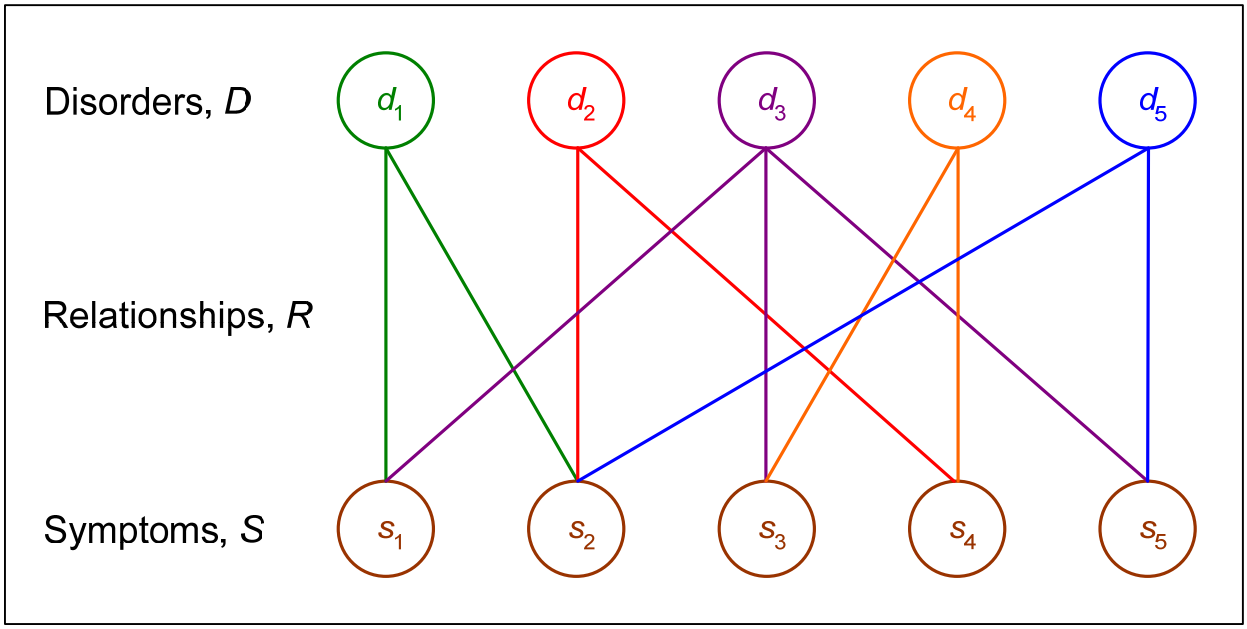

Figure 3-10. Set covering graph of relationships between disorders and symptoms

Set covering theory has been applied to numerous systems for diagnosing multiple disorders. The paper by Reggia et al. (1983) offers a solid introduction to set covering theory and its applications to knowledge-based systems. Peng & Reggia (1986, 1987) expanded this work by creating a hybrid system to take advantage of both set covering and Bayesian classification techniques. In more recent years, a paper by Baumeister et al. (2001) presents a set covering system that incrementally refines itself along with an excellent overview of set covering theory. In the past 5 years, Atzmueller et al. (2003a, 2003b, 2004a, 2004b) have presented a significant amount of research including the expansion of set covering to make use of diagnostic scores and case-based reasoning.

Conclusion

In this chapter, we discussed the methods used in designing and implementing both knowledge-based systems and data mining systems. We began with an extensive discussion of

rule-based systems and certainty factors. From there we moved to other foundational topics, including case-based reasoning, nearest neighbor classification, and Bayes' rule. Next we discussed the less mainstream topics of fuzzy logic, Dempster-Schafer, and rough sets along with other approaches less relevant to our research, such as genetic algorithms and artificial neural networks. The chapter concluded by discussing Bayesian belief networks and set theory, which are two of the most relevant modern approaches to diagnosing multiple disorders with knowledge-based systems.

The following chapter begins with a discussion of the mathematics used throughout the medical field. It continues with a discussion of important knowledge-based systems in the medical field. Finally, it concludes with a literature review of systems that have been designed for the purpose of diagnosing medical disorders.

CHAPTER 4
MEDICAL MATHEMATICS AND RELEVANT KNOWLEDGE-BASED SYSTEMS

The previous chapter discussed a variety of established approaches for knowledge-based systems and data mining. This chapter gives an overview of many systems that have been developed using those techniques. A strong emphasis is given to historical medical expert systems, diagnostic systems in toxicology, and modern systems for diagnosing multiple disorders, as these are most relevant to this research. The information presented in this chapter is cursory at best, and the reader is encouraged to study the references for a proper understanding of any systems of interest. Before addressing the systems, however, the chapter begins with a discussion of the mathematics employed in the field of medicine.

Medical Mathematics

The medical field presents a unique set of challenges for knowledge engineering. Distinctions ranging from ethical and legal issues to fundamentally different mathematical understandings set the domain of medicine apart from all other domains. Cios and Moore (2002) thoroughly discuss the considerations that must be observed in the medical field. For our purposes, however, let us focus on the standard mathematical approaches used for decision-making in medicine.

Probabilistic Measurements

Many knowledge-based systems use precision as a conclusive measurement of performance. Precision is the percentage of true positives (TP) compared to the total number of cases classified as positive events:

$$precision = \frac{TP}{TP + FP} \times 100\%, \tag{4-1}$$

where FP represents false positives. According to Cios and Moore (2002), "This measurement is very popular in machine learning and pattern recognition communities, but is not acceptable in

medicine because it hides essential details of the achieved results" (p. 4). To better understand the performance of a diagnostic test, the medical profession defines a number of other measurements. Let us begin by examining a contingency table (Table 4-1). Contingency tables contain four variables: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). A true positive occurs when a test correctly diagnoses a patient as having a disorder. A true negative occurs when a test correctly diagnoses a patient as not having a disorder. A false positive occurs when a test incorrectly diagnoses a patient as having a disorder. A false negative occurs when a test incorrectly diagnoses a patient as not having a disorder.

Table 4-1. Contingency table

Test results	Disorder present	Disorder absent	Total
Positive	TP	FP	TP + FP
Negative	FN	TN	TN + FN
Total	TP + FN	TN + FP	

Another measurement of performance, frequently used in conjunction with precision, is accuracy. Accuracy is the number of correctly classified cases compared to the total number of cases presented to a system:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%. \tag{4-2}$$

Even when used in combination, however, precision and accuracy do not fully capture the information necessary for medical diagnosis. Perhaps the most common measurements in the medical field are sensitivity and specificity, which are defined as:

$$sensitivity = \frac{TP}{TP + FN}, \tag{4-3}$$

$$specificity = \frac{TN}{TN + FP}. \tag{4-4}$$

Sensitivity is also known as the true-positive rate (TPR). It represents the probability that the test detects the disorder, given that the patient has the disorder. Specificity is also known as the

true-negative rate (TNR). It represents the probability of the test detecting no disorder, given that the patient truly does not have the disorder. These measurements are important in diagnosis because tests are never absolutely accurate. There are always instances where a patient with a disorder displays fewer symptoms than a patient without the disorder. For this reason, diagnostic tests in medicine are tuned to "rule in" or "rule out" a diagnosis. If a physician is attempting to "rule in" a disorder, he should select a test with a high specificity so that a positive test result strongly confirms his premonition. Likewise, a physician attempting to "rule out" a disorder should use a test with a high sensitivity.

Two measurements often used in conjunction with sensitivity and specificity are the false-negative rate (FNR) and the false-positive rate (FPR), also known as the false-alarm rate (FAR):

$$FNR = \frac{FN}{FN + TP} = 1 - sensitivity\,, \tag{4-5}$$

$$FAR = \frac{FP}{FP + TN} = 1 - specificity\,. \tag{4-6}$$

The false-negative rate and false-alarm rate are probabilities associated with a test inaccurately diagnosing a patient. The false-negative rate represents the probability of a test failing to detect a disorder that is present, whereas, the false-alarm rate represents the probability of a test falsely indicating that a patient has a disorder.

Alternates to sensitivity and specificity are the positive predictive value (PPV) and negative predictive value (NPV):

$$PPV = \frac{TP}{TP + FP}\,, \tag{4-7}$$

$$NPV = \frac{TN}{TN + FN}\,. \tag{4-8}$$

The positive predictive value is the likelihood that, given positive test results, the patient does indeed have the disorder. In a similar manner, the negative predictive value is the likelihood that, given negative test results, the patient truly does not have the disorder. There is an important difference between the measurements of sensitivity and specificity and the measurements of positive and negative predictive values. While sensitivity and specificity are independent of the population being tested, positive and negative predictive value are effected by the prevalence of a disease within a population.

To contrast sensitivity and specificity with positive and negative predictive values, let us consider an example from the Medical University of South Carolina Doctoring Curriculum (2000). Imagine that a new test for detecting HIV is discovered. To determine its usefulness, an experiment with 10,000 HIV infected blood samples and 10,000 non-infected blood samples is performed. The testing results in all correct answers except for 10 false positives and 10 false negatives, yielding a sensitivity, specificity, PPV, and NPV of 99.9% (Table 4-2). Additionally, the pre-test probability indicates that there is a 50% chance for a randomly selected blood sample to contain HIV.

Table 4-2. Experimental HIV testing extended contingency table

Test results	Disorder present	Disorder absent	Total	
Positive	9,990 (TP)	10 (FP)	10,000	PPV = 9,990/10,000 = 99.9%
Negative	10 (FN)	9,990 (TN)	10,000	NPV = 9,990/10,000 = 99.9%
Total	10,000	10,000	20,000	
	Sensitivity = 9,990/10,000 = 99.9%	Specificity = 9,990/10,000 = 99.9%		Pre-test probability = 10,000/20,000 = 50%

Now let us apply the test to a population of one million people where 1% of the population is infected with HIV. Since sensitivity and specificity are a function of the ability of a test to identify HIV carriers, their values do not change (Table 4-3). In contrast, the PPV decreases by 8.9% and the NPV increases slightly. The significance of the decrease in PPV is that if the physician informed patients that they had HIV based solely on this test, 990 individuals would be falsely informed that they were infected. Neither sensitivity nor specificity gives the physician an indicator of this change from the previous contingency table.

Table 4-3. HIV testing (1% chance of HIV) extended contingency table

Test results	Disorder present	Disorder absent	Total	
Positive	9,990 (TP)	990 (FP)	10,980	PPV = 9,990/10,980 = 91.0%
Negative	10 (FN)	989,010 (TN)	989,020	NPV = 989,010/989,020 = 99.999%
Total	10,000	990,000	1,000,000	
	Sensitivity = 9,990/10,000 = 99.9%	Specificity = 989,010/990,000 = 99.9%		Pre-test probability = 10,000/1,000,000 = 1%

Table 4-4. HIV testing (0.1% chance of HIV) extended contingency table

Test results	Disorder present	Disorder absent	Total	
Positive	999 (TP)	999 (FP)	1,998	PPV = 999/1,998 = 50.0%
Negative	1 (FN)	998,001 (TN)	998,002	NPV = 998,001/998,002 = 99.999%
Total	1,000	999,000	1,000,000	
	Sensitivity = 999/1,000 = 99.9%	Specificity = 998,001/999,000 = 99.9%		Pre-test probability = 1,000/1,000,000 = 0.1%

Let us look at one more example in relation to HIV testing. Applying the test to a pool of blood donors that have already been screened for HIV risk factors, we would expect the

percentage of HIV infected individuals to be closer to 0.1%. Again, the contingency table is shown for a population of one million people (Table 4-4). The calculations show that the PPV drops to 50%, while sensitivity and specificity remain constant. The results from these three contingency tables demonstrate that while sensitivity and specificity normally should remain constant, the PPV and NPV are dependent upon the prevalence of a disease within a population.

The final mathematical expression commonly used in the medical field is the likelihood ratio (LR). The likelihood ratio is the odds that a specific test result is given to a patient with the disorder compared to the same test result being given to a patient without the disorder. There are two types of likelihood ratios, LR^+ and LR^-, which can be calculated as follows:

$$LR^+ = \frac{\left(\frac{TP}{TP+FN}\right)}{\left(\frac{FP}{FP+TN}\right)} = \frac{sensitivity}{FAR}, \tag{4-9}$$

$$LR^- = \frac{\left(\frac{FN}{TP+FN}\right)}{\left(\frac{TN}{FP+TN}\right)} = \frac{FNR}{specificity}, \tag{4-10}$$

where LR^+ is the odds that a positive test result occurs for a patient with the disorder versus one without the disorder and LR^- is the odds that a negative test result occurs for a patient with the disorder verses one without the disorder. Thus, good diagnostic tests should have a high LR^+ and a low LR^-. One major advantage of likelihood ratios is that they can be easily combined through multiplication. For this reason, the system discussed in Chapters 5 and 6 utilizes LR^+ to diagnose toxic exposures.

In this section, we have discussed the most common statistical expressions in medical literature. For more detailed information, refer to Owens and Sox (2001). The next section discusses a different process frequently used in medical diagnosis.

Diagnostic Scores

The use of diagnostic scores is a simple approach to risk analysis in the medical field. In this method, signs and symptoms are assigned a point value, or score, based on their correlation to a specific disorder. In forming a diagnosis, the physician gathers a list of all the signs and symptoms observed in the patient. He then looks up the score for each observation on a chart. Adding the scores together yields the final diagnostic score, which is compared to another chart to determine the risk of the patient for a specific disorder.

Table 4-5 and Table 4-6 are examples of the charts that a physician might use when implementing diagnostic scores for appendicitis, based on the research by Ohmann et al. (1999). As can be seen, if a patient's only variables are rigidity and having an age less than 50 years old, the final diagnostic score sums to 2.5. In Table 4-6, we find that the odds of the patient having appendicitis are 3%. From these observations, the physician can be fairly certain that the patient does not have appendicitis. However, if the patient satisfied the requirements for every variable in Table 4-5, the patient's final diagnostic score would be 16.0 indicating a 68% risk of appendicitis. Depending on the final diagnostic score, the physician may recommend different tests or treatments for the patient.

Table 4-5. Diagnostic scores for acute appendicitis (Ohmann et al., 1999)

Variable	Points
Tenderness, right lower quadrant	4.5
Rebound tenderness	2.5
No micturition difficulties	2.0
Steady pain	2.0
Leukocyte count $\geq 10.0 \times 10^9$/L	1.5
Age < 50 years	1.5
Relocation of pain to right lower quadrant	1.0
Rigidity	1.0

Table 4-6. Final diagnosis score significance (Ohmann et al., 1999)

Diagnostic score	Frequency
< 4.0 points	3%
4.0–5.5 points	5%
6.0–7.5 points	11%
8.0–9.5 points	24%
10.0–11.5 points	32%
12.0–13.5 points	55%
> 14.0 points	68%

Literature Review of Knowledge-Based Systems

Although the field of medicine possesses its own established mathematical methods, surprisingly few systems have taken advantage of them. The majority of systems rely on established engineering approaches or devise their own representation scheme. This section begins by discussing two significant historical systems in the field of medicine. It then presents a selection of systems directly applied to the field of toxicology. The chapter concludes by addressing the systems specifically designed for the diagnosis of multiple disorders.

Historical Medical Expert Systems

Research in the field of medical expert systems began in the early 1970's with the development of MYCIN. Created at Stanford University by Buchanan and Shortliffe (1984a), MYCIN was designed for the purpose of diagnosing infectious blood diseases to recommend the appropriate antibiotics for treatment. MYCIN utilized a knowledge base consisting of approximately 500 rules, and its inference engine architecture was constructed as an inference network. The system would query the user using simple yes or no questions until enough information was gathered to identify the bacteria responsible for the symptoms. To handle uncertainty, MYCIN employed certainty factors, discussed in Chapter 3. Research has shown that using these methods MYCIN was able to outperform faculty at the Stanford medical school in diagnosing diseases within its domain (Yu et al., 1984). Interestingly, later research seems to indicate that the use of certainty factors was superfluous and that MYCIN could perform equally

effectively without them (Buchanan & Shortliffe, 1984b). Although a foundational pillar in the field of expert systems, MYCIN was never used in the medical field. Ethical and legal issues along with a mistrust of computer systems within the field of medicine were major contributors in preventing its commercialization. For an exhaustive discussion about MYCIN, see Buchanan & Shortliffe (1984a).

A second foundational system in the field of medical expert systems is INTERNIST. INTERNIST was developed by Pople (1977) at the University of Pittsburgh during the same era as MYCIN. The goal in developing INTERNIST was to create a system capable of handling general internal medicine, as opposed to the specialized domains traditionally occupied by expert systems (Pople, 1985a). INTERNIST was developed and refined over the course of a decade through interviews with Jack Myers, MD, and became one of the largest and broadest expert systems ever created. Before completion, the system contained information on more than 3550 symptoms (Miller et al., 1982) and could diagnose more than 750 diseases (Pople, 1985b). INTERNIST's inference engine employed a ranking program to perform diagnosis. To make the domain size manageable, it also used heuristically guided partitioning rules. By breaking problems down into smaller subsets, the system was able to better handle the broad domain of internal medicine. CADUCEUS, an eventual successor to INTERNIST, implemented a problem decomposition method in an attempt to better handle multiple disorders. Regrettably, CADUCEUS suffered from other limitations due to its requirement for prior knowledge of domain structure. Pople (1985b) presents a thorough progression of the INTERNIST system from its origins through the development of CADUCEUS. Other references of note include Pople (1977) and Miller et al. (1982). For an excellent example of INTERNIST's interface and interaction with the user, see Pople (1985a).

Expert Systems in Toxicology

There exist surprisingly few knowledge-based systems in the field of clinical toxicology. In fact, according to Darmoni, in 1995 "Toxline and Toxlit [showed] that less than ten computer-aided decision support systems [had] been developed in clinical toxicology" (p. 234). Of these systems, two in particular stand out from the rest: a French system called SETH and a Bulgarian system called MEDICOTOX-CONSILIUM. In recent years, two more systems of interest have been developed, the Inreca system for use in Russia and a Polish veterinary system. A summary of each of these four systems is given below.

SETH was developed in France by Darmoni et al. (1994, 1995) for use in the Rouen University Hospital. The system uses 70 signs and symptoms for diagnosis and contains over 1000 drugs from over 75 toxicological classes (Darmoni, 1994). SETH was implemented on a commercial off the shelf, object oriented, expert system shell called KBMS. Its inference engine is a rule-based, forward chaining system that utilizes the Rete algorithm for pattern matching. SETH also makes use of set theory for diagnosing cases involving multiple drugs. In 1992, the system began experimental use at the Rouen University Hospital where it was used in the diagnosis of over 2000 drug intoxication cases (Darmoni, 1995). Although its creators caution that the system was not designed for use by experts, the ratings given by residents at the hospital indicate that they were pleased with the system. For more information on SETH, refer to Darmoni et al. (1994, 1995).

MEDICOTOX-CONSILIUM was developed for use in the hospitals of Bulgaria as a diagnostic system for first aid clinical toxicology. It was first implemented in 1988 at a single hospital and eventually distributed to 11 more hospitals around the country. The system is described as a classical system that uses frame structures, rules, and scores provided by experts for diagnosis. Within the frame structure, poisons are divided into 10 classes with 310 groups

containing a total of 2500 different kinds of poisons (Monov et al., 1992). The system contains 1000 rules and facts that use 47 syndrome and 134 symptom definitions to identify poisons and supply the user with information about the appropriate cure from any of 86 treatments and 55 antidotes (Monov et al., 1992). MEDICOTOX-CONSILIUM is focused on user interaction and, rather than simply producing a diagnosis, seeks to leave the final decision to the user. It also offers three different modes to maximize its utility in different circumstances. The first mode is the clinical orientation mode and is useful for diagnosing urgent cases where immediate action is required. The second mode is the diagnostic research mode that can be used to carefully reason through less urgent cases. The final mode is the expert-reference mode that enables the user to look up information on any of the drugs and toxins contained within the system. For more information on MEDICOTOX-CONSILIUM, refer to Monov et al. (1992).

Another system for toxicology was developed by Althoff et al. (1998) for use by the Russian Toxicology Information and Advisory Center in Moscow. The system is based on previous research called the Inreca (Induction and Reasoning from Cases) European project. The Inreca approach is a case-based reasoning system designed to use historical cases to diagnose disorders. This particular system utilizes the database of the Toxicology Information and Advisory Center of the Russian Federation Ministry of Health and Medical Industry to supply the cases for diagnosing poison exposures. The location of Russia was explicitly chosen for its abundance of data because "every year Russia has more intoxication cases than any other country in Europe" (Althoff et al., 1998, p. 27). A distinct aspect to this case-based reasoning system is that, rather than interpreting the cases at run time, it compiles the data into an Inreca-Tree in advance to improve performance. The Inreca-Tree is basically a specialized decision tree that includes a branch at every decision node to account for the possibility of

unknown measurements. For more information about the Inreca system for diagnosing poison cases, refer to Althoff et al. (1998).

The final toxicology system of interest is being developed at Warsaw Agriculture University in Poland by Kluza (2004) for the area of veterinary medicine. The system utilizes case-based reasoning for the purpose of offering remote consultations to veterinarians working in the field. As presented by Kluza in 2004, the project is still in the launching phase; however, since it is being designed for veterinarian medicine the system faces some unique challenges of interest. First, being designed for animals, Kluza's system must not only include gender and age in its diagnosis, but must also account for differences between various species and breeds. Second, because animals cannot verbally communicate with veterinarians, every diagnosis must be performed without certain prior knowledge that is often available in toxicology cases involving humans. For more information, see Kluza (2004).

The four systems presented in this section were selected to give the reader a general understanding of the techniques that have been used in clinical toxicology. SETH and MEDICOTOX-CONSILIUM are two of the most prominent systems in the field. The Inreca approach is an excellent example of the simplicity and robustness that are necessary for advancement in the fields of knowledge-based systems and data mining. Finally, the Polish veterinary system represents a current topic of research in the field and poses some unique challenges for consideration. Very few knowledge-based systems exist for the field of clinical toxicology. For a fairly exhaustive list of the systems being used throughout the field, consult Darmoni et al. (1994).

Knowledge-Based Systems for the Diagnosis of Multiple Disorders

The previous section discussed expert systems in the field of clinical toxicology. Although most of these systems are by necessity forced to address the challenge of diagnosing multiple

disorders to some degree, it was not the primary thrust of the research. This section presents an overview of the research that has been done explicitly for the purpose of diagnosing multiple disorders. Like the field of clinical toxicology, relatively little knowledge-based system research has been performed in the area of multiple disorders. This section presents research on four major types of systems used in multiple disorder diagnosis. The four methods discussed include Bayesian approaches, case-based reasoning, set covering, and diagnostic scores. Note that, although we divide these systems into four types for the sake of discussion, many systems may contain aspects from multiple approaches.

To begin, let us discuss systems that use Bayesian approaches. Bayes' rule is a probability based equation that can be used to identify the most likely disorder. The problem is that Bayes' rule requires independence of the symptoms used in diagnosis. Much research has focused on the generalization of Bayes' rule to account for dependencies within a domain.

Research by Ben-Bassat et al. (1983) presents a Bayesian pattern recognition algorithm used in the MEDAS emergency diagnosis system to overcome the limitations of Bayes' rule. Ben-Bassat et al. (1980) also discuss some of the other early approaches for "handling a violation of the conditional independence assumption in classical Bayesian diagnosis models" (p. 153). One of the most noteworthy accomplishments of Bayesian research was the construction of Bayesian belief networks, discussed in Chapter 3. One system, created by van der Gaag and Wessels (1994) uses belief networks to diagnose multiple disorders. The distinctive feature of the system is that it utilizes a clustering algorithm to strategically focus on small sets within the domain as a method of improving efficiency.

In more recent years, another system called HEPAR II was developed by Onisko et al. (2000). HEPAR II uses belief networks to diagnose multiple disorders in the field of hepatology.

Onisko et al. (2001) further developed the system by creating a method for building belief networks from a small data set. To accomplish this, they implemented what they refer to as "Noisy-OR gates" to increase the accuracy of the system.

A second approach to diagnosing multiple disorders is case-based reasoning. The advantage of case-based reasoning is that systems can essentially create themselves from historical cases, unlike most complex models, including Bayesian networks, that generally require knowledge acquisition from experts (Atzmueller et al., 2004b). It is important to note that case-based reasoning is an approach to system development rather than a method for reconciling uncertainty and probabilistic dependencies. For this reason, many case-based reasoning systems make use of other methods. The ADAPtER system combines case-based reasoning with abductive model based reasoning to diagnose multiple car engine faults (Portinale & Torasso, 1995). SONOCONSULT makes use of inductive methods to augment its case-based reasoning and recognize multiple disorders in the field of sonography (Baumeister et al., 2002). Atzmueller et al. (2003a) continued the research on SONOCONSULT by exploring the use of decomposition methods within a case-based system. Finally, Atzmueller et al. (2004b) present three approaches to case-based reasoning for the diagnosis of multiple disorders. The approaches presented include compositional case adaptation, where a group of cases is recalled for diagnosis rather than a single case, the partition class approach, where domains are divided into independent subsets for diagnosis, and set covering, which is discussed in Chapter 3.

Set covering is a method that seeks to find combinations of disorders that can account for observed symptoms. The simplicity and elegance of the approach makes it one of the most promising areas in research relating to multiple disorder diagnosis. In the 1980's, Reggia and Peng published a large amount of foundational research on set covering. The paper by Reggia et

al. (1983) is one of the clearest and most referenced papers in the area of set covering for multiple disorders. The system they propose, however, only holds "for the extreme case that might be called complete decomposability" (Wu, 1991, p. 240). In later research, Peng and Reggia (1986, 1987) expand on what they refer to as "parsimonious covering theory" by adding Bayesian calculations to allow for multimembership classification. Parsimonious covering theory is essentially set covering where the simplest solution is considered the best solution. In their 1989 paper that presents further enhancements to the system, Peng and Reggia implement the use of "comfort measures." The purpose of comfort measures is to ensure that the system maintains a certain level of quality in the solutions it returns to the user. In the early 1990's, Wu extended the field of set covering by developing algorithms to increase efficiency. Wu's research primarily centers around decomposing a problem into smaller sub-problems using a clustering algorithm (Wu, 1990, 1991). In other research, genetic algorithms were applied to generate a set covering system for multi-disorder diagnosis (Vinterbo & Ohno-Machado, 2000) and simple systems were given the ability to incrementally refine themselves, adding complexity as more samples become available (Baumeister et al., 2001).

The final approach to be discussed is the use of diagnostic scores for the diagnosis of multiple disorders. In particular, this discussion details the research performed by Atzmueller et al. (2003b, 2004a), as it bears the most resemblance to the system presented in Chapters 5 and 6. Atzmueller et al. (2003b) have implemented a case-based system for the diagnosis of multiple disorders in the field of sonography. The system is semi-automatic, meaning that the system generates its rules automatically but still requires an expert to oversee its development and adjust parameters as necessary to ensure the system functions properly. Atzmueller et al. (2003b) believe that "understandability and interpretability...is of prime importance" and so their system

attempts to apply "the same representation the human expert favors" by using diagnostic scores (p. 23). Diagnostic scores are a simple approach for risk analysis used in the medical field, discussed earlier in this chapter.

Using a case base, the system creates scoring rules, *r*, of the form:

$$r = f \xrightarrow{s} d, \tag{4-11}$$

where *f* represents a finding, such as the observation of a sign or symptom, *d* represents the diagnosis related to that finding, and *s* represents a qualitative measure of uncertainty with $s \in \{S_3, S_2, S_1, 0, S_{-1}, S_{-2}, S_{-3}\}$. Scores of $s \in \{S_1, S_2, S_3\}$ represent a positive correlation, where S_3 strongly supports diagnosis *d* and S_1 weakly supports diagnosis *d*. Likewise, scores of $s \in \{S_{-1}, S_{-2}, S_{-3}\}$ represent a negative correlation, where S_{-3} strongly opposes diagnosis *d* and S_{-1} weakly opposes diagnosis *d*. When $s = 0$, no significant correlation is found and the rule is later pruned from the rule set. As defined by Atzmueller et al. (2003b), four scores from the same category yield the next higher score, such that:

$$S_1 + S_1 + S_1 + S_1 = S_2, \tag{4-12}$$
$$S_2 + S_2 + S_2 + S_2 = S_3, \tag{4-13}$$
$$S_{-1} + S_{-1} + S_{-1} + S_{-1} = S_{-2}, \tag{4-14}$$
$$S_{-2} + S_{-2} + S_{-2} + S_{-2} = S_{-3}. \tag{4-15}$$

Also, any two scores of equal and opposite number cancel, such that:

$$S_1 + S_{-1} = 0, \tag{4-16}$$
$$S_2 + S_{-2} = 0, \tag{4-17}$$
$$S_3 + S_{-3} = 0. \tag{4-18}$$

A diagnosis *d* is considered "probable" if the aggregate score is equal to or greater than S_3. Note that substituting a score of 1 for S_1, 4 for S_2, and 16 for S_3 makes the system presented here comparable to the diagnostic scoring system presented earlier in the chapter, where a final diagnostic score of 16 represents the cutoff point for a diagnosis being considered highly likely.

To determine the score value that a rule should receive, Atzmueller et al. (2003b) use a quasi probabilistic score. The quasi probabilistic score is calculated by a mathematical equation that combines the statistical dependence of a finding with its precision and specificity. The resulting value ranges from -1.0 to 1.0 and is mapped to a corresponding *s* value.

Atzmueller et al. (2003b) are concerned with the balance between accuracy and complexity. For this reason, their system utilizes diagnostic profiles, P_d, defined as:

$$P_d = (F_d, frec_F), \tag{4-19}$$

where F_d represents the findings most frequently associated with a diagnosis and $frec_F$ contains the frequencies of those findings. The frequencies in the diagnostic profile are used to prune less important rules for system efficiency. Further efficiency is gained through other pruning criteria as well as partitioning the domain using background knowledge provided by an expert in the field. In later work, Atzmueller et al. (2004a) proposed a quality measure equation as a means to measure and determine the appropriate balance between accuracy and simplicity.

Conclusion

This chapter began by presenting the common mathematical calculations used in the medical field. From there, the discussion moved from historical medical expert systems to systems designed specifically for the field of toxicology. The chapter concluded by discussing four approaches to diagnosing multiple disorders. Considerable emphasis was placed on the system created by Atzmueller et al. (2003b) due to its similarity to the system presented in the following chapters. The next chapter details the development of a system for toxic exposure diagnosis and its performance when diagnosing single exposure cases.

CHAPTER 5
DIAGNOSING SINGLE EXPOSURE CASES

The previous chapter presented diagnostic systems for toxicology as well as modern research towards the diagnosis of multiple disorders. This chapter describes the first stage of exploratory research performed using data from the Florida Poison Information Center (FPIC) to create a system for diagnosing multiple exposure cases. The system presented in this chapter is capable of generating a differential diagnosis for exposures to a single toxin. The chapter begins by describing the source data, continues by discussing system design principles and development, presents the system's operation with respect to the user interface, and concludes with a discussion of system testing, research results, and, finally, system performance.

Source Data

Since 1996, the FPIC has collected data on every call received and made follow-up calls to obtain additional information about cases referred to hospitals. The collected data is stored in a relational database, consisting of tables where each entry in a table is an object with a key that enables relationships to be drawn between tables. In 2004 alone, the FPIC received over 120 thousand calls and made more than 43 thousand follow-up calls related to human exposures (Florida Poison Information Center Network, 2005). The FPIC database also contains over 65 thousand records of multiple exposures. For this research, the FPIC provided access to all the cases recorded in its Jacksonville database from 2002 through 2005. The information supplied contains more than 160 thousand toxic exposure cases, with nearly 14 thousand cases involving multiple toxins. To improve data quality, the database records were cleaned so that only cases with clinical effects that were followed to a known outcome remained. The cleaned database contained 30,152 single exposure cases and 7,096 multiple exposure cases, however, the system's training only involved single exposure cases for this portion of the research.

The database supplied by the FPIC conforms to the Toxic Exposure Surveillance System (TESS) standard. TESS is the older of two national standards defined by the American Association of Poison Control Centers (AAPCC) to regulate the fields contained within the database of each poison control center (PCC). The newest standard, known as the National Poison Data System (NPDS), was not fully developed at the time of this research. As a result, the system presented here utilizes TESS standardized data fields. However, using TESS rather than NPDS standards does not affect the system's general design principles because both TESS and NPDS use the same paradigm and record the same set of data. Both are national standards that will enable the system to be expanded to a national level and implemented at various PCC's throughout the country. Both require that the majority of entries in the database have discrete values, which are easy to process with a computer program. Most importantly, both contain the observed signs and symptoms, jointly called clinical effects, and the final diagnosis of patients referred to hospitals for treatment.

Although the FPIC database is a valuable resource, it may contain errors. Patients may lie about the substances they consume or physicians and nurses may not fully recount all the important details of a case when reporting to the PCC. Fortunately, these errors can be viewed as random errors. As the case base for the system grows, the incorrect information should become negligible when contributing to system calculations.

System Design Principles

From the outset, a major objective of this research was to bypass the knowledge acquisition bottleneck by generating a knowledge-based system capable of producing meaningful and useful results without the need for an active, overseeing expert. In developing this system to diagnose unknown exposures, certain guiding principles were followed to produce the desired system characteristics, which include simplicity, understandability, automatic system generation,

and incremental updates. Each of these characteristics is discussed briefly in the following paragraphs.

The characteristic of simplicity is of the utmost importance. Holsheimer et al. (1995) have shown that success in extracting information from databases does not require complex algorithms. In fact, "simpler, even trivial, processes are better than complicated ones if they are enough for the job of discovery" (Valdes-Perez, 1999, p. 336). Simplicity inherently gives systems several advantages. Generally, systems with simple representations and algorithms are more efficient and require less processing power. Simple, linear calculations grant the system scalability, which is extremely important given the size and continual growth of the FPIC database. Systems designed with simpler architectures are often more portable to other systems. Portability is desirable not only for aiding other PCC's around the country, but also so that the system approach can be used to solve diagnostic problems in other domains. Finally, simplicity of design gives the system inherent understandability. Not only should the system and its processes be easier to comprehend and implement by other knowledge engineers, but the solutions yielded by the system should be explained in terminology that physicians will understand.

The understandability of system results was another chief concern during development. If physicians understand the method by which the system obtains its answers, they are more likely to trust the system and use it within the spectrum of its intended purpose. According to Atzmueller et al. (2003b), "understandability and interpretability of…learned models is of prime importance" and "ideally, the learning method constructs knowledge in the same representation the human expert favors" (p. 23). For this reason, the final system design makes use of likelihood ratios. Likelihood ratios are commonly used throughout the medical field and are

discussed in Chapter 4 along with other medical mathematics. After processing, the system presents its results to the user as a differential diagnosis. A differential diagnosis is a list of various disorders that can produce similar clinical effects. It is used to determine the most likely cause of a disorder and is a method commonly practiced in the medical field. By using these familiar approaches, physicians should find the system to be relevant, understandable, and easy to operate. Furthermore, the methods used in the system's mathematics are similar to medical case studies seeking to identify patterns of clinical syndromes. It is believed that this will help the system gain acceptance in the medical field.

Automatic system generation is another desirable trait. Atzmueller et al. (2003b) state that "pure automatic learning methods are usually not good enough to reach a quality comparable to manually built knowledge bases" (p. 23). In spite of this deficiency, automatic methods offer certain advantages that should not be overlooked. Automatically trained systems fully bypass the knowledge acquisition bottleneck of obtaining information from an expert. An expert's time is valuable, and the more processing a system can do without expert input, the more rapidly it can be developed and implemented. Additionally, automatically generated system designs can be broadly applicable to solving problems, which makes the system significantly more portable than one containing expert input that leads to specialization within a given field. The system presented in this chapter was generated by an engineer with no expertise in the area of toxicology and no guidance from toxicologists regarding specific diagnostic approaches. Bypassing the information bottleneck and using a generally applicable, medical solution increases the value of the system as a whole.

The final desired attribute of the system is the ability to perform incremental updates. As the FPIC database grows in size, more valuable information will become available for aiding in

diagnosis. Although the system could recompile all the data from 1996 to the present with every update, such an operation would be inefficient and could require significant processing time. Rather than beginning anew each update, the system can maintain key information about current values and incorporate the information from the latest cases into its calculations. Currently, incremental updates have not been implemented because the system is not directly linked to the central database. However, the use of likelihood ratios makes the implementation of incremental updates a straightforward procedure. To calculate likelihood ratios, a count of true positives, true negatives, false positives, and false negatives must be determined for each clinical effect. By saving a table of these four values with their corresponding substance, the likelihood ratio can be calculated. Updating the system then becomes a simple matter of querying the new data for a count of each of the four values, adding the results to the old table, and recalculating the likelihood ratios. Graefe et al. (1998) presents examples of other information that can be used in incremental updates. Han and Kamber (2001) also briefly discuss incremental and parallel data mining for the combining of gathered information.

System Development

The goal of the research presented in this chapter is to create a system using data mining and knowledge engineering techniques on a database obtained from the FPIC to aid in the diagnosis of exposures to a single unknown toxin. The system must receive a physician's input in the form of signs and symptoms observed in a patient, process the data, and return a list of the substances that are most likely to induce these clinical effects. The aim of the system is not to produce infallible results for every case. Rather, the system attempts to give the physician easy access to a refined and organized version of the knowledge stored in the FPIC's vast database. The system offers direction by presenting a differential diagnosis of drugs and other toxic

substances that should be considered. Ultimately, the physician makes the final decision regarding the treatment the patient should receive.

In generating the system, data mining techniques are used to clean the records and extract the appropriate information from the FPIC database. First, informational calls are removed so that only exposure cases remain. Then, the exposure cases are filtered so that only cases with clinical effects that were followed to a known outcome remain. Although this reduces the size of the dataset to 30,152 single exposure cases and 7,096 multiple exposure cases, the filtering process ensures that only significant representative cases with the best documentation are used to train the system. Each exposure case has clinical effects associated with it. The clinical effects observed in a patient are rated as either "related," "unknown if related," or "not related" to the substance involved in the exposure. For the purposes of system training, clinical effects that are "not related" are removed from the database while those that are "unknown if related" are used for training in the same way as "related" clinical effects.

After extracting and cleaning the cases in the database, a table of prior probabilities, also known as pre-test probabilities, is calculated for each toxin. A prior probability represents the likelihood of a particular substance being involved, given that a toxic exposure has occurred. Prior probability, P, is calculated as:

$$P = \frac{Cases}{Total}, \tag{5-1}$$

where Cases is the number of cases involving a particular substance and Total is the total number of exposure cases in the database.

In addition to prior probabilities, a table of likelihood ratios is calculated. When calculating likelihood ratios, the system treats each clinical effect as a diagnostic test that is useful in detecting the presence of a toxic substance. Likelihood ratios represent the odds that an

observed clinical effect is caused by a particular toxin versus the odds that the clinical effect is the result of exposure to any other toxin. The likelihood ratio, LR^+, is calculated as:

$$LR^+ = \frac{\left(\frac{TP}{TP + FN}\right)}{\left(\frac{FP}{FP + TN}\right)}, \tag{5-2}$$

where TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives. An exhaustive table of likelihood ratios relating every individual clinical effect to every possible substance exposure is the primary resource utilized by the system in creating a differential diagnosis. An advantage of likelihood ratios over many other medical measurements (i.e. sensitivity, specificity, positive and negative predictive values, etc.) is that likelihood ratios can be easily combined through multiplication. Additionally, by including the prior probability, likelihood ratios can account for disorder prevalence. Furthermore, likelihood ratios are easily calculated and characterize many cases with a single number, making the system scalable to large databases and ensuring a rapid response time.

Although the likelihood ratio has its advantages, it inherently contains the drawbacks of every mathematical ratio, the possibility of evaluating to zero or causing a divide-by-zero error. A likelihood ratio of zero only occurs when $TP = 0$ and may not seem like a problem until we understand how the system calculates combined likelihood ratios. Every clinical effect is treated as a test for detecting the presence of a toxic substance. If there are multiple clinical effects, their likelihood ratios are multiplied together to obtain a combined likelihood ratio. If any of the clinical effects has a likelihood ratio of zero, then the substance's combined likelihood ratio also evaluates to zero, regardless of the evidence presented by other clinical effects. The problem is that the absence of cases associating a substance with a clinical effect does not mean that the substance absolutely cannot cause that clinical effect. Furthermore, even if the substance truly

cannot cause the clinical effect, patients may have unassociated clinical effects caused by other ailments.

The divide-by-zero error is an obvious problem for any computer system. Looking at Equation 5-2, we can see that if $TP + FN = 0$ or $FP = 0$, the calculation fails. (Note that although $FP + TN = 0$ causes an error, addressing $FP = 0$ also prevents that error from occurring.) The sum of true positives and false negatives ($TP + FN$) is the total number of cases where a particular substance is involved. Due to the structure of the database and its queries, a substance with no recorded cases in the database is ignored and not included as a valid diagnosis in the system. As a result, $TP + FN$ never equals zero. The second divide-by-zero error, $FP = 0$, occurs whenever a clinical effect only appears in the database with an association to one particular substance. As calculated, the likelihood ratio concludes that since no other substance causes the clinical effect, that substance absolutely must be the cause, so it divides by zero to obtain an infinite likelihood. In reality, however, no single substance is the only possible cause for any clinical effect in the system. The problem is lack of sufficient data. The divide-by-zero error was encountered during development because the database contains only one instance of fetal death. Although fetal death can be caused by any number of substances, the system attempted to conclude that only acetaminophen could cause the death of a fetus.

The preliminary system used a simple-minded approach to solving the multiplication by zero and divide-by-zero problems. Multiplication by zero was handled by replacing all zero-valued likelihood ratios with a value of one. Although this prevents the system from gaining any knowledge about a substance from a clinical effect not associated with the substance, it prevents that clinical effect from destroying the knowledge gained from other clinical effects. The divide-by-zero error was solved by examining the data set and manually

modifying the offending clinical effect records. The likelihood ratios calculated using the method described in this paragraph are referred to as "non-adjusted" likelihood ratios from this point forward.

Although using non-adjusted likelihood ratios expedited the research process, it introduced significant drawbacks. First, replacing likelihood ratios of zero with the value of one ignores the information that could be gained from the calculation. Likelihood ratios can be fractional, indicating a negative correlation to a substance, and multiplying by zero indicates an infinitely negative correlation. Rather than throwing the negative association out completely, the zero value might be tempered by using some fractional likelihood ratio. Second, manually removing problematic cases from the database violates the important design principle of automatic system generation. To solve these problems, a generalized equation was developed to replace the likelihood ratio:

$$LR^{+}_{Adj} = \frac{\left(\frac{TP + \Delta}{TP + \Delta + FN + \Delta}\right)}{\left(\frac{FP + \Delta}{FP + \Delta + TN + \Delta}\right)}, \tag{5-3}$$

where TP represents true positives, TN represents true negatives, FP represents false positives, FN represents false negatives, and Δ is a small, positive constant. As discussed in Chapter 4, TP, TN, FP, and FN represent the four possible outcomes of a diagnostic test. By adding Δ to each outcome, the equation states that any of these outcomes is a possibility, even if no supporting cases exist in the database. The end result is a stable equation that closely approximates the likelihood ratio, avoids the difficulties of multiplying by zero, prevents the divide-by-zero error, and converges to the same value as the likelihood ratio as the number of cases increases. A variety of Δ values were calculated and compared, including 1.0, 0.1, 0.01, and 0.001.

Ultimately, a Δ of 0.01 was selected as it appeared to improve diagnosis significantly while still yielding a suitable substitute for the likelihood ratio. Equation 5-3 with $\Delta = 0.01$ is referred to as the "adjusted" likelihood ratio from this point forward.

The system described is a hybrid system containing elements of data mining, case-based reasoning, rule-based systems, and uncertainty management. Data mining techniques are used to clean and extract relevant information from the database. Case-based reasoning methodology makes use of the example cases obtained by data mining to develop a system that runs on composite observations. The system calculations are essentially a set of simple rules running in parallel with likelihood ratios implemented to handle uncertainty. From the results of these rules, a ranked list is generated to indicate the most likely substances that account for the given signs and symptoms. Moreover, uncertainty management is employed by the use of adjusted likelihood ratios to make system calculations robust in the face of database anomalies.

System Operation and User Interface

As discussed in the previous section, the system utilizes two tables of calculations to create a differential diagnosis. The first table contains the prior probabilities for every substance. The second table consists of likelihood ratios relating every individual clinical effect to every possible substance exposure. When supplied with a set of clinical effects, the system calculates a combined likelihood ratio, including prior probability, for every potential single exposure diagnosis. The results are then sorted and presented as a differential diagnosis to the user.

The user interface reveals more about the functionality of the system (Figure 5-1). Clinical effects are grouped into nine categories defined by TESS: cardiovascular, dermal, gastrointestinal, heme/hepatic, neurological, ocular, renal/GU, respiratory, and miscellaneous. Each group of clinical effects can be viewed by selecting the appropriate tab from the top of the

interface. In the figure, the gastrointestinal disorders tab is selected to show the various TESS defined clinical effects associated with this category. Three disorders are selected: abdominal pain, dehydration, and diarrhea. More disorders may be selected from other category tabs as well.

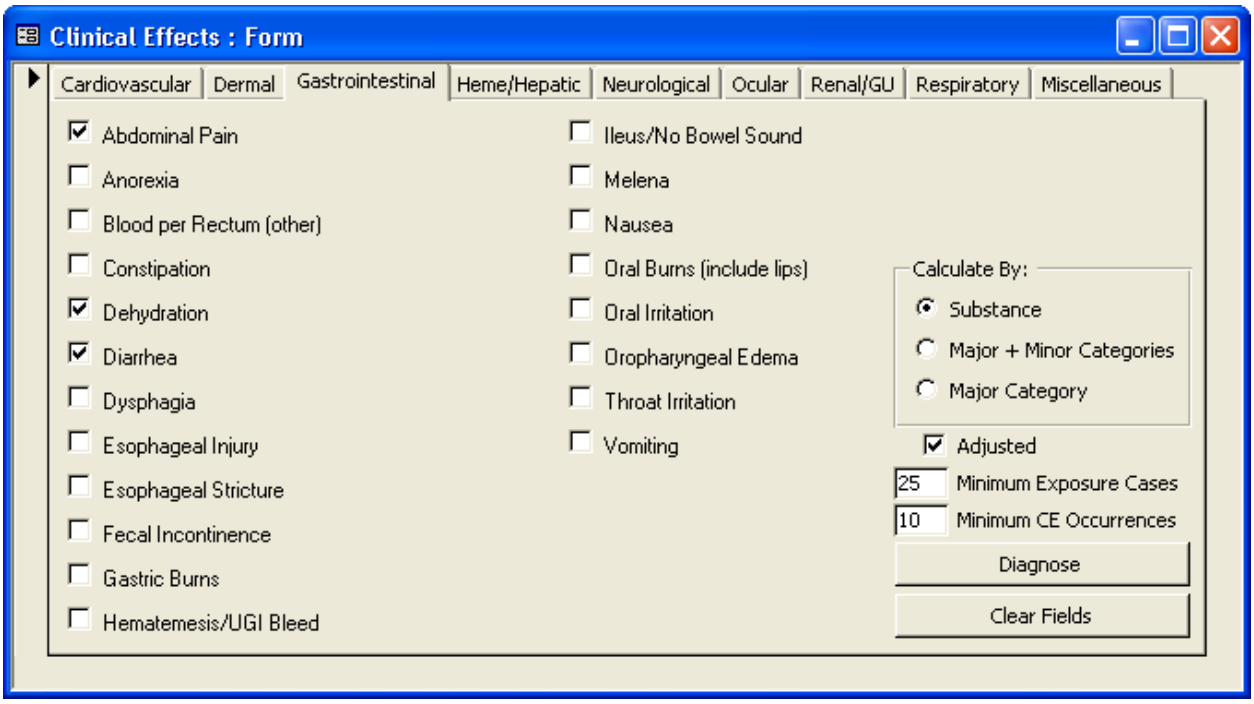

Figure 5-1. User interface

The controls for various system parameters are on the right hand side of the user interface. The "Calculate By" selection box enables the user to select substance, major and minor categories, or major category. Thus far, we have discussed the research only in terms of diagnosing exposures to a single toxic substance; however, each substance belongs to a minor category, which in turn belongs to a major category. In the same manner that the system is trained to diagnose individual substances, it can be trained to diagnose based on major and minor categories or even solely based on major category. Giving physicians a general idea of the drug categories they should consider may prove every bit as valuable as attempting to directly diagnose a substance.

Below the "Calculate By" selection box is a check box where the user can select to calculate likelihood ratios as non-adjusted or adjusted. As discussed in the previous section, non-adjusted calculations replace all likelihood ratios of zero with a one and require the system designer to manually remove anomalies that might cause divide-by-zero errors. The adjusted likelihood ratio, shown in Equation 5-3, makes a slight modification to the traditional likelihood ratio to create a more robust equation that prevents the system from failure due to multiplication or division by zero. As shown in Figure 5-1, the box is checked so that the system calculates the adjusted likelihood ratio.

Below the "Adjusted" check box are numerical values for "Minimum Exposure Cases" (MC) and "Minimum CE Occurrences" (MCE). These numbers serve as data filters that are used to eliminate diagnoses and clinical effects with poor representative sampling sizes. The MC box enables the user to set the minimum required number of cases for a diagnosis. If a diagnosis does not have at least as many cases in the database as the number in the box, the diagnosis does not appear on the results table. The MCE box enables the user to set the minimum number of times a clinical effect (CE) must appear in the database. If a clinical effect does not appear in the database at least as many times as the number entered, the clinical effect is ignored when calculating the likelihood ratio even if the clinical effect is checked by the user.

The last two features of the user interface are the "Clear Fields" and "Diagnose" buttons. The "Clear Fields" button removes all check marks from every clinical effect regardless of the selected tab. This enables the user to be sure that the check marks on other tabs have been cleared without having to manually flip through each tab individually. The "Diagnose" button runs the system program, displaying a differential diagnosis table to the user. Clicking on the "Diagnose" button with the settings in Figure 5-1 displays a table similar to the one shown in

Figure 5-2. The table contains the calculated likelihood ratio (LR) on the left and the associated diagnosis on the right. The results in the figure indicate bacterial food poisoning, with a likelihood ratio of 148.9, is by far the most likely cause of abdominal pain, dehydration, and diarrhea. The second most likely cause is mushrooms, with a likelihood of 3.32. It should be noted that, although likelihood ratios are helpful for indicating the strength of support for various diagnoses, rank on the list is more important. Physicians should consider many of the substances in the top ten before making their final diagnosis.

RESULTS_SS_ADJ : Table

LR	SubDesc
148.949829478	BACTERIAL FOOD POISONING: UNKNOWN TYPE
3.32482311621	UNKNOWN MUSHROOM
0.31018510633	MULTI-BOTANICAL WITHOUT MA HUAN OR CITRUS AURANTIUM
0.29315333996	CARDIAC GLYCOSIDE
0.18296274892	LITHIUM
0.13801692951	ORGANOPHOSPHATE
0.11248568761	SUSPECTED FOOD POISONING-UNKNOWN TYPE-PATIENT SYMPTOMATIC
0.06391201604	ACETAMINOPHEN WITH HYDROCODONE
0.06183397575	MULTI VITAMIN-TABLET: CHILD WITH IRON (NO FLUORIDE)
0.05855603720	LAXATIVE

Record: 1 of 265

Figure 5-2. Results table

System Testing and Results

For testing, the system's prior probabilities and likelihood ratios were trained on approximately 90% of the cases in the database. After training, the system attempted to diagnose the remaining 10% of the cases using only the associated clinical effects. The correct diagnosis for each case was then compared to the system's differential diagnosis and the rank of the correct diagnosis was saved to a summary table. The system then retrained on a new set of data and was tested against a different 10% of the database. The training and testing datasets were determined by the last digit of each case identification number, ensuring a unique test set every cycle. The process was repeated ten times, completely testing the system against every case contained in the

database. Throughout the process a large amount of data was gathered, the results of which are presented in the following paragraphs.

The ten-cycle testing process was used to compare the effectiveness of adjusted versus non-adjusted likelihood ratios. Both likelihood ratio calculations were tested at all three diagnostic levels: diagnosing by substance, diagnosing by major and minor categories, and diagnosing by major category alone. Additionally, the settings for the MC and MCE filters were varied to produce multiple points of comparison. While maintaining a constant MCE value of 10, MC was tested at 10, 25, and 100. Likewise, while maintaining a constant MC value of 25, MCE was tested at 0, 10, and 50. Furthermore, four levels of medical outcomes were tested against the system: all exposures with a minor severity or worse, moderate severity or worse, major severity or worse, and a severity level where the outcome was death. These tests yielded sixty resultant sets for both adjusted and non-adjusted likelihood ratios.

After generating these results, the accuracy of the sixty adjusted sets was compared to the accuracy of the sixty non-adjusted sets. Accuracies were calculated in three ways: the percentage of exposures appearing as the top diagnosis, the percentage of exposures appearing in the top ten diagnoses, and the percentage of exposures appearing in the top 10% of the trained diagnoses. Comparing adjusted accuracies with non-adjusted accuracies, it was determined that adjusted likelihood ratios appear to be a good approximation of non-adjusted likelihood ratios, with adjusted calculations yielding a higher accuracy 90% of the time. Of the 180 accuracy calculations, there were eighteen exceptions where non-adjusted calculations outperformed adjusted calculations. Ten of these exceptions involved the outcome of death. There are a few explanations for this anomaly. First, there are very few death cases recorded in the database, making it more likely that random variation might favor one system approach over another.

Second, death cases may often display clinical effects that are not normally associated with a particular toxic exposure. The reason is that the systems in the body begin to shut down and extreme failures begin to cause cascading effects. In such cases, it becomes impossible to reliably compare two diagnostic systems. The accuracies of the remaining eight exceptions were within 0.5% of the corresponding adjusted performances. This nominal gain is more than compensated for by the 127 instances where adjusted calculations outperformed non-adjusted calculations on test cases not limited to the outcome of death. Additionally, a system based on adjusted calculations is much easier to generate automatically than one based on non-adjusted calculations because it does not require any manual intervention by the system designer. Having established that the adjusted likelihood ratio is a valid substitute for the traditional likelihood ratio, the remainder of the research results is discussed in terms of adjusted calculations.

The next step in system development was to determine the best values for the MCE and MC filters. Beginning with a constant MC value of 25, MCE was varied and tested for values of 0, 2, 5, 10, and 50. For each of these values, the adjusted system was also tested at the three diagnosis levels of substance, major and minor categories, and major category alone. Each of the three diagnosis levels yields a system with a significantly different number of trained diagnoses. To enable comparisons between the three diagnosis levels, the percentage of exposures appearing in the top 10% of the trained diagnoses was used as the accuracy measurement. Table 5-1 shows the accuracy of the system when diagnosing by substance, Table 5-2 when diagnosing by major and minor categories, and Table 5-3 when diagnosing by major category alone. Looking at Table 5-1 under minor severity, it can be seen that varying MCE has no effect on the accuracy of the system. Under major severity, the accuracy decreases from 77.8% to 77.6%, a negligible change. Likewise, looking at Table 5-2 and Table 5-3 it becomes obvious

that varying MCE causes little to no change for minor, moderate, and major severities. Once again, the exception is the severity where the outcome is death, which is most likely due to a small sampling size. For example, the 5.1% increase in accuracy observed in exposures with an outcome of death being diagnosed by major and minor categories is a difference of only four additional cases being diagnosed in the top 10%. Prior to these tests, it was believed that using too low of an MCE cutoff might create falsely high or low likelihood ratios in some substances, decreasing diagnosis accuracy. However, based on these results, it is reasonable to conclude that filtering by MCE yields negligible changes in system accuracy. Using an adjusted likelihood ratio with $\Delta = 0.01$ already mitigates the potential problem, thus, the filter can be removed from the system.

Table 5-1. Accuracy by substance in 10% (MC = 25)

		Minimum CE Occurrences (MCE)				
		0	2	5	10	50
Severity	Minor	64.7%	64.7%	64.7%	64.7%	64.7%
	Moderate	74.4%	74.4%	74.4%	74.4%	74.2%
	Major	77.8%	77.8%	77.7%	77.7%	77.6%
	Death	62.2%	62.2%	62.2%	62.2%	58.1%

Table 5-2. Accuracy by major and minor categories in 10% (MC = 25)

		Minimum CE Occurrences (MCE)				
		0	2	5	10	50
Severity	Minor	64.1%	64.1%	64.1%	64.1%	63.9%
	Moderate	72.7%	72.7%	72.7%	72.6%	72.4%
	Major	75.4%	75.4%	75.4%	75.1%	75.4%
	Death	58.2%	58.2%	58.2%	58.2%	63.3%

Table 5-3. Accuracy by major category in 10% (MC = 25)

		Minimum CE Occurrences (MCE)				
		0	2	5	10	50
Severity	Minor	63.9%	63.9%	63.8%	63.8%	63.8%
	Moderate	70.0%	70.0%	69.9%	69.9%	69.8%
	Major	70.5%	70.5%	70.4%	70.0%	70.5%
	Death	55.7%	55.7%	54.4%	54.4%	54.4%

The second filter to be examined was the MC filter. Using adjusted calculations with a constant MCE value of 10, MC was tested for values of 0, 2, 5, 10, 25, 50, and 100. Again, to enable comparisons between the three diagnosis levels of substance, major and minor categories, and major category alone, the percentage of exposures appearing in the top 10% of the trained diagnoses was used as the accuracy measurement. Additionally, since varying MC directly affects the number of trained diagnoses in the system, it was hoped that the 10% accuracy measurement would enable comparisons between systems generated by different MC filter values. Table 5-4 shows the accuracy of the system when diagnosing by substance, Table 5-5 when diagnosing by major and minor categories, and Table 5-6 when diagnosing by major category alone. Looking at the accuracies for minor, moderate, and major severities in both Table 5-4 and Table 5-5, it is readily apparent that accuracy generally appears to decrease as MC increases. Table 5-6 shows the same tendency for MC steps from 10 to 25 and 25 to 100, but appears to plateau for MC values from 0 to 10 and 25 to 50. At first it might appear that using a lower MC yields a more accurate system, and, therefore, the MC filter should be removed. However, such a conclusion fails to account for the purpose of the MC filter. As MC decreases, more possible diagnoses with less supporting cases are added to the system. As more diagnoses are added to the system, the accuracy calculation based on the top 10% includes substances that are ranked lower on the differential diagnosis. It turns out that the number of diagnoses that are added to the top 10% outweighs the number of new exposure cases being tested against the system. As a result, the lower the MC value, the more accurate the system appears. The plateaus observed in Table 5-6 are also accounted for by this explanation because the top 10% of cases evaluates to the same number for MC's of 0, 2, 5, and 10 as well as for MC's of 25 and 50.

Table 5-4. Accuracy by substance in 10% (MCE = 10)

		Minimum Exposure Cases (MC)						
		0	2	5	10	25	50	100
Severity	Minor	74.4%	72.9%	69.6%	67.4%	64.7%	62.9%	58.9%
	Moderate	80.3%	79.7%	78.0%	76.6%	74.4%	71.6%	64.8%
	Major	80.6%	81.7%	81.3%	79.8%	77.7%	75.1%	68.6%
	Death	62.0%	65.8%	63.3%	62.8%	62.2%	61.2%	46.8%

Table 5-5. Accuracy by major and minor categories in 10% (MCE = 10)

		Minimum Exposure Cases (MC)						
		0	2	5	10	25	50	100
Severity	Minor	71.7%	70.1%	68.9%	67.5%	64.1%	63.0%	58.4%
	Moderate	79.1%	78.0%	76.9%	75.6%	72.6%	71.8%	66.7%
	Major	81.1%	80.3%	79.7%	78.5%	75.1%	74.2%	69.2%
	Death	67.1%	68.4%	68.4%	67.1%	58.2%	59.2%	49.3%

Table 5-6. Accuracy by major category in 10% (MCE = 10)

		Minimum Exposure Cases (MC)						
		0	2	5	10	25	50	100
Severity	Minor	68.5%	68.5%	68.5%	68.6%	63.8%	64.1%	60.3%
	Moderate	73.4%	73.4%	73.4%	73.5%	69.9%	70.2%	66.4%
	Major	73.9%	73.9%	74.0%	74.0%	70.0%	70.3%	65.2%
	Death	58.2%	58.2%	58.2%	58.2%	54.4%	56.4%	51.3%

Since comparing MC values using an accuracy based on the top 10% of trained diagnoses failed to yield the desired results, a second accuracy measurement was calculated using the correct diagnoses appearing in the top ten slots of the differential diagnosis. From a user standpoint, this accuracy measurement is more appropriate because the list size that a user can process without being overwhelmed is not dependent on the number of trained substances. Table 5-7 shows the accuracy of the system when diagnosing by substance, Table 5-8 when diagnosing by major and minor categories, and Table 5-9 when diagnosing by major category alone. Looking at the minor, moderate, and major severity rows in Table 5-7, Table 5-8, and Table 5-9, it can be seen that as MC increases, accuracy also increases. The data tells us little about selecting a value for MC because it indicates what is expected of any system: As more cases are used to define each substance, system accuracy should increase. Another contributor to

the increase in accuracy is that fewer substances are trained as MC increases. With fewer substances, the top ten substances become a larger portion of the available diagnoses. Even random guessing would experience an increase in accuracy under these circumstances.

Table 5-7. Accuracy by substance in 10 (MCE = 10)

		Minimum Exposure Cases (MC)						
		0	2	5	10	25	50	100
Severity	Minor	41.2%	41.4%	42.1%	43.2%	47.2%	54.4%	71.0%
	Moderate	50.0%	50.5%	51.4%	52.9%	57.1%	63.2%	76.4%
	Major	53.4%	54.6%	56.0%	58.2%	62.4%	68.1%	77.7%
	Death	35.4%	39.2%	41.8%	44.9%	45.9%	50.7%	57.4%

Table 5-8. Accuracy by major and minor categories in 10 (MCE = 10)

		Minimum Exposure Cases (MC)						
		0	2	5	10	25	50	100
Severity	Minor	63.0%	63.1%	63.2%	63.4%	64.1%	65.3%	71.0%
	Moderate	71.4%	71.6%	71.7%	72.0%	72.6%	74.1%	78.1%
	Major	73.6%	74.2%	74.3%	74.7%	75.1%	76.4%	79.0%
	Death	58.2%	58.2%	58.2%	58.2%	58.2%	60.5%	61.3%

Table 5-9. Accuracy by major category in 10 (MCE = 10)

		Minimum Exposure Cases (MC)						
		0	2	5	10	25	50	100
Severity	Minor	79.5%	79.5%	79.5%	79.6%	79.8%	80.1%	82.4%
	Moderate	83.8%	83.8%	83.9%	83.9%	84.0%	84.5%	86.3%
	Major	83.9%	84.1%	84.3%	84.3%	84.5%	84.7%	85.4%
	Death	69.6%	70.9%	72.2%	72.2%	72.2%	71.8%	71.8%

In an attempt to normalize accuracies, a ratio of the data in Table 5-7, Table 5-8, and Table 5-9 versus the accuracy of diagnosing by random guessing was calculated. However, it was found that the ratio suffered from problems similar to the accuracies calculated in Table 5-4, Table 5-5, and Table 5-6. Lowering MC increases the number of trained diagnoses in the system, adversely effecting random guessing. As a result, the ratio falsely indicated that a lower MC cutoff would yield better results. A second attempt at normalizing the accuracies calculated the ratio of the data in Table 5-7, Table 5-8, and Table 5-9 against a system that selected its top ten choices based on prior probabilities alone. Figure 5-3 shows a graph of the ratio for minor,

moderate, and major severities when diagnosing by substance. Likewise, Figure 5-4 displays the ratio for diagnosing by major and minor categories and Figure 5-5 the ratio for diagnosing by major category alone. The graphs indicate that as MC increases, ensuring better representative likelihood calculations, the system tends to perform better. The increase appears to be almost linear, with perhaps a slight tendency towards diminishing returns as MC increases. There is no evidence of any breakpoints that would yield a superior MC cutoff. These results indicate that the adjusted likelihood ratio is performing well and that the exact value used for MC is unimportant. However, a reasonable MC value of at least ten should be chosen to ensure that outliers do not excessively influence diagnosis.

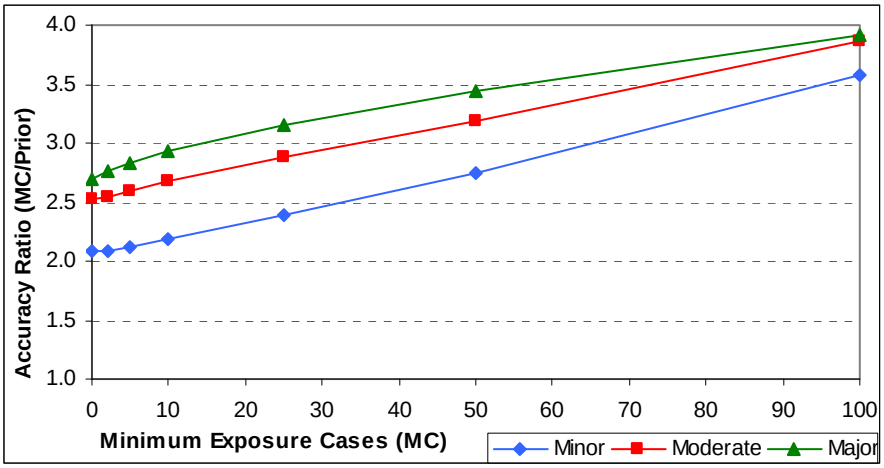

Figure 5-3. Accuracy ratios by substance

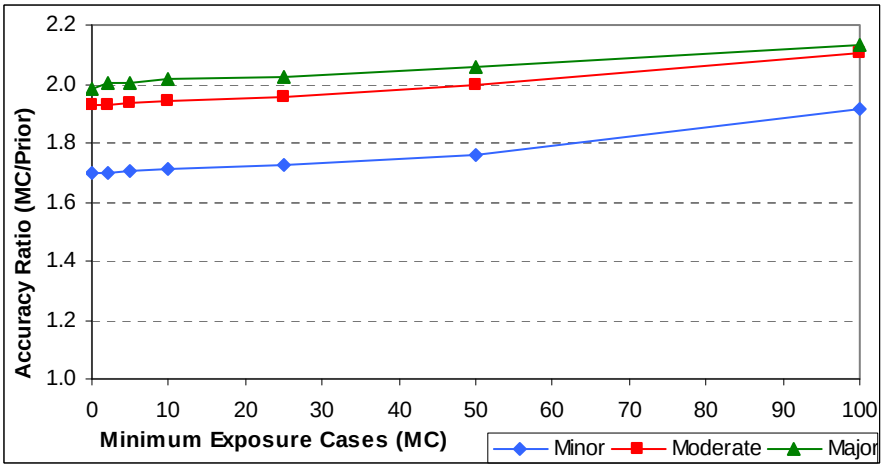

Figure 5-4. Accuracy ratios by major and minor categories

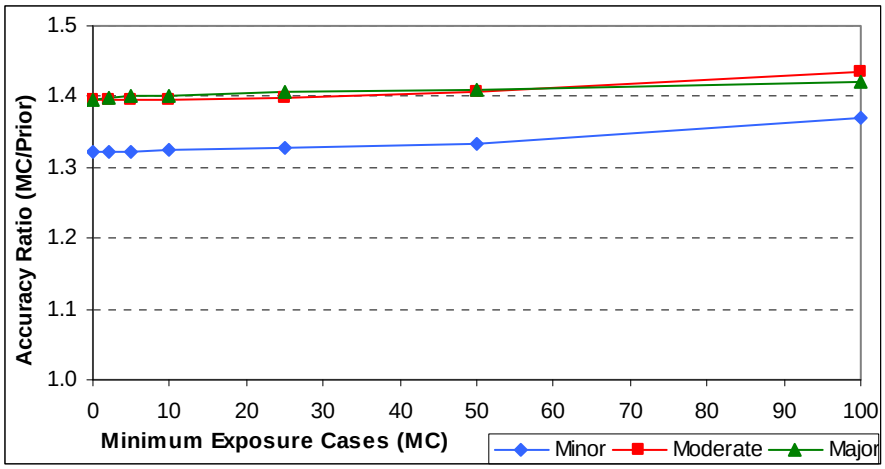

Figure 5-5. Accuracy ratios by major category

Comparing Figure 5-3, Figure 5-4, and Figure 5-5, it can be seen that the slopes and the ratios are higher for diagnosis by substance than diagnosis by major and minor categories, which in turn are higher than diagnosis by major categories. The reason is that the number of diagnoses trained for diagnosing by substance (around 200 to 600) is significantly more than diagnosing by major and minor categories (around 100 to 200), which is more than diagnosing by major category alone (around 50 to 60). With more possible diagnoses, the problem becomes more difficult to diagnose in the top ten without intelligence. Thus, the system's performance ratio improves as more substances are added. Additionally, the curves indicate that the system scales well to a large number of diagnoses since the ratios and slopes increase as the available diagnoses increase. Another notable characteristic of the curves is that they indicate that the system performs better the more severe the case. The primary reason is that more severe cases generally have more associated clinical effects. With more clinical effects, the system has more information to properly differentiate between various diagnoses, yielding a higher accuracy. The good news is that the most important cases are the most severe cases, and this is precisely where the system performs best.

Table 5-10 displays a representative chart of system performance with MC = 10 and MCE = 0. To enable comparison between the various forms of diagnosis, the percentage of exposures appearing in the top 10% of trained diagnoses is used as the accuracy calculation. Table 5-10 reiterates the fact that the system performs better the more severe the case. Once again, death is the exception due to limitations in the data and system failures in the body leading to cascading clinical effects. Moreover, the difficulties associated with the cases involving death make it futile to discuss trends for that severity. For major and moderate severities, diagnosing by substance performs best, followed by major and minor categories and finally by major category alone. The converse is true for minor severity cases, where diagnosing by major category alone performs best. Though not universally observed in the test runs, this accuracy inversion is not uncommon and is most likely due to the lack of clinical effects in minor severity cases. With minimal clinical effects it is easier to classify the general major category of a toxin than to identify the specific toxin involved.

Table 5-10. Accuracy in 10% with MC = 10 and MCE = 0

		Diagnosis by:		
		Substance	Major & Minor categories	Major category
Severity	Minor	67.4%	67.5%	68.6%
	Moderate	76.6%	75.7%	73.5%
	Major	79.8%	78.9%	74.8%
	Death	62.8%	67.1%	59.5%

The accuracy calculations in Table 5-10 show a high value of 79.8%, which occurs when diagnosing major severity cases by substance. These accuracy calculations include a large number of cases involving only a single clinical effect, which would be difficult for even the most experienced expert to diagnose without additional information. To better demonstrate system functionality, the accuracies from Table 5-10 are recalculated in Table 5-11 to include only cases with at least three recorded clinical effects. A large improvement in system accuracy

is observed, particularly in minor severity cases where accuracies are boosted from the 60% range into the mid-70% range. Additionally, the accuracy of diagnosing major severity cases by substance and by major and minor categories is raised above 80%. Further system improvements could be achieved by removing useless categories, such as the "unknown drug" diagnosis, and consolidating nearly redundant substances, such as "aspirin: pediatric formulation," "aspirin: unknown if adult or pediatric formulation," and "aspirin: adult formulation." However, one purpose of the research presented here is to bypass the need for expert input when generating a system and making such improvements would assume knowledge in the domain of toxicology.

Table 5-11. Accuracy in 10% with MC = 10, MCE = 0, and 3+ CE's

		Diagnosis by:		
		Substance	Major & Minor categories	Major category
Severity	Minor	75.1%	74.6%	74.0%
	Moderate	78.4%	77.2%	75.0%
	Major	81.0%	80.5%	75.9%
	Death	69.4%	71.4%	66.7%

Further credence to the system's viability was subjectively given by two toxicologists at the FPIC who experimented with the system's user interface. The toxicologists found the top diagnosis to be reasonable for every input given to the system as well as several other appropriate diagnoses listed in the top ten. Considering the purpose of the system, as an automatically generated toxicology consultant, and the intentional simplicity of the system design, the resulting accuracies and positive reactions by toxicology experts confirm that the research performed to create this system was a success.

System Performance

An important aspect of system usability is the amount of processing time required to train the system and the response time of the user interface to diagnostic queries. System calculations

were intentionally kept simple to enable scalability, rapid system generation, and a low response time. For research purposes, the system was developed using Microsoft Access 2002 on a Compaq Presario 2100 laptop with a 2.4GHz processor and 320MB of RAM. Training the system on four years of data took just under three minutes. Running a diagnosis under worst case conditions takes approximately three seconds when the program is first queried. Once loaded into RAM, however, the diagnosis runtime is cut in half. Obviously, porting the program to the dedicated SQL server used by the FPIC would offer further speed improvements.

As the number of cases in the system increases, system training time could increase significantly, though there should be a minimal impact on diagnosis time due to the architecture of the system. The section in this chapter entitled "System Design Principles," discusses a method for incremental updates that enables the system to retain its training from previous years and simply adjust the system's calculations based on new data. Based on the performance measurements taken, it is reasonable to expect that the system could be trained rapidly by a central database server. In the future, training results could be downloaded to applications on handheld personal digital assistants without significant loss of usability.

Conclusion

This chapter has presented the research and development of a system capable of generating a differential diagnosis for exposures to a single toxin. First, we discussed the source data and guiding system design principles of simplicity, understandability, automatic system generation, and incremental updates. Next, the theory behind system development was explained. Finally, the system operation, user interface, system results, and system performance were presented. The system presented here serves as a foundation for the multiple exposure research presented next chapter.

CHAPTER 6
DIAGNOSING MULTIPLE EXPOSURE CASES

The previous chapter presented the development of a system for diagnosing exposures to a single toxin. The resulting system serves as a foundation for the multiple exposure research discussed in this chapter. Although system development did not proceed as expected, the results reveal intriguing insights into the diagnosis of multiple exposures in the field of toxicology. The chapter begins with the motivation for developing the system, continues by briefly describing the system approach, discusses the results of diagnosing multiple disorders using various sets of training data, presents the research conclusions, and closes with a discussion of future work.

Motivation for Diagnosing Multiple Exposures

Although many established methods for designing knowledge-based systems exist, as discussed in Chapter 3, none have fully solved the problem of diagnosing multiple disorders. The difficulty is that multiple disorder cases can display non-linear interactions, a problem observed in the field of toxicology. When simultaneously present in the body, toxins can interact antagonistically or synergistically, masking or otherwise altering the signs and symptoms that would normally appear for each individual exposure. Little documentation exists for the majority of toxic exposure combinations that can occur and only a limited number of systems exist in the field of toxicology that attempt to account for multiple exposures in some way, as covered in Chapter 4. None of these systems fully solve the problem, nor are they readily available for use by American toxicologists.

Beyond the motivation of developing technology that addresses an unsolved diagnostic problem, the more important concern of saving lives is at stake. The Toxic Exposure Surveillance System (TESS) report states that in 2004 “50.6% of fatal cases involved 2 or more drugs or products” (Watson et al., 2004, p. 593). This statistic makes it plain that timely and

accurate identification of exposures involving multiple substances is extremely important. For the sake of advancing the field of information engineering as well as the preservation of life, the problem addressed by the research in this chapter is both relevant and important.

System Approach

Chapter 5 presents the development of a system for diagnosing exposures to a single toxin. That system serves as a foundation for the diagnosis of multiple exposure cases discussed in this chapter. Like the single exposure system, the goal of the multiple exposure system is to serve as a consultant by producing differential diagnoses based on the clinical effects supplied by the user. Unless otherwise noted, the training and testing procedures for the system described in this chapter conform to the following characteristics:

- The clinical effects and substance identifiers are based on TESS standards.
- Prior probabilities and adjusted likelihood ratios with a Δ of 0.01 are used to determine the differential diagnoses, see Equations 5-1 and 5-3 for details.
- The system is tested at three diagnostic levels: diagnosing by substance, diagnosing by major and minor categories, and diagnosing by major category alone.
- Three levels of medical outcomes are tested against the system: exposures with a minor severity or worse, moderate severity or worse, and major severity or worse. (Note that testing solely based on exposures resulting in death is not included due to the inaccuracies discussed in Chapter 5.)
- A minimum exposure cases (MC) value of 10 and minimum clinical effect occurrences (MCE) value of 0 serve as the cutoffs for testing the system.
- Accuracies are calculated as the percentage of test exposures identified correctly in the top 10% of the trained diagnoses.

For this research, the Florida Poison Information Center (FPIC) provided access to all the cases recorded in its Jacksonville database from 2002 through 2006. With the addition of a fifth year, the cleaned database used for system generation now contains 37,617 single exposure cases and 8,901 multiple exposure cases.

Diagnosing Multiple Exposures using Solely Multiple Exposure Cases

During the initial phase of testing, all multiple exposure cases are extracted from the database. TESS standards require that each substance involved in a toxic exposure be assigned a sequence number that ranks the substance in accordance with its relative contribution to the observed clinical effects. To simplify the initial attempts to diagnose multiple disorders, only the primary and secondary contributors in each multiple exposure case are considered. TESS standards also require that substances be recorded by a product specific code as well as a generic substance code. From this requirement, a problem arises. When determining the number of substances involved in an exposure, the FPIC database uses the product specific code. As a result, two products marketed by different companies are listed as separate substances, even if their active ingredient is the same. When cleaning the data, if the generic substance codes for the top three contributing substances are identical, the case is removed from the dataset. If the first two generic substance codes are identical but the third is different, the third substance is treated as the secondary contributor for the case. Finally, the multiple exposure cases are filtered so that only cases resulting in minor effects, moderate effects, major effects, or death are used to train and test the system. The cleaned dataset contains 8,901 multiple exposure cases.

When generating the multiple exposure system, each pair of primary and secondary contributors is trained individually as a single diagnosis. Prior probabilities, adjusted likelihood ratios, and both MC and MCE filters are calculated and implemented in the same manner as discussed in Chapter 5. Testing also follows the same process of training the system on approximately 90% of the cases and then attempting to diagnose the remaining 10% of the cases. By repeating the process ten times, the system is completely tested against every case in the database. Finally, the results are combined and accuracies calculated as the percentage of test exposures identified correctly in the top 10% of the trained diagnoses.

The original results from training and testing the system on multiple exposure cases are displayed in the first column of Table 6-1. With an accuracy ranging from 28.3% to 50.1%, the system's deplorable performance is painfully obvious. To further explore the failure, the system was tested for MC values of 15, 20, and 25. The results of these tests show a similar lack of accuracy (Table 6-1). Looking at the rows in the table from left to right, we can see that the performance gradually decays as MC increases. As discussed in Chapter 5, such an observation is expected due to the MC cutoff lowering the number of diagnoses included in the top 10%. The most interesting characteristic of the data in Table 6-1 is that as the severity increases, the accuracy decreases. This observation is contrary to the results observed in the single exposure system. Normally, the system's accuracy increases with severity because more severe cases contain more clinical effects, making diagnosis easier for the system.

Table 6-1. Accuracy (varying MC) of system trained & tested on multiple exposures

		Minimum Exposure Cases (MC)			
Diagnosed by	Severity	10	15	20	25
Substance	Minor	33.5%	30.4%	29.0%	27.6%
	Moderate	30.0%	26.9%	25.3%	22.9%
	Major	28.3%	23.3%	21.8%	18.5%
Major & Minor categories	Minor	47.3%	43.6%	39.5%	38.2%
	Moderate	45.9%	42.1%	37.6%	36.5%
	Major	37.6%	34.5%	30.9%	30.6%
Major category	Minor	50.1%	46.8%	45.7%	43.4%
	Moderate	47.2%	44.1%	43.0%	40.4%
	Major	43.0%	39.6%	38.2%	36.5%
	Average	40.3%	36.8%	34.5%	32.7%

There are a number of plausible explanations for why accuracy might decrease with severity, but two are particularly compelling. The first explanation is that the decrease in accuracy is caused by the non-linear interactions between multiple toxins. As the severity of an exposure increases, there is greater opportunity for a combination of toxins to produce effects not normally associated with any of the toxins individually. This could lower the accuracy of the

system because the clinical effects would behave more erratically and might not correspond to the majority of cases. The second explanation is that the decrease in accuracy is simply caused by lack of quality data. As the severity cutoff becomes more stringent, fewer cases are tested against the system, leading to a poor sampling and quite possibly lower accuracies on average. Lack of quality data could account for both the low accuracy observed overall as well as the decrease in accuracy as the severity increases.

Another parameter that might contribute to the system's poor accuracy is the Δ parameter implemented in the adjusted likelihood ratio equation, see Equation 5-3. The Δ parameter is meant primarily to safeguard against multiply-by-zero and divide-by-zero errors, however, a small training set might cause Δ to adversely influence the diagnostic results. Table 6-2 compares the original system accuracy, when using a Δ of 0.01, to accuracies calculated with a Δ of 0.1 and 0.001. It was discovered that increasing Δ to 0.1 causes an average decrease in accuracy of 1.6%, while decreasing Δ to 0.001 causes an average increase in accuracy of only 0.1%. These results imply that a Δ of 0.01 yields satisfactory relative performance compared to other Δ parameters that might be selected.

Table 6-2. Accuracy (varying Δ) of system trained & tested on multiple exposures

Diagnosed by	Severity	Δ = 0.1	Δ = 0.01	Δ = 0.001
Substance	Minor	32.3%	33.5%	33.5%
	Moderate	29.0%	30.0%	29.8%
	Major	26.4%	28.3%	28.0%
Major & Minor categories	Minor	46.5%	47.3%	47.4%
	Moderate	44.6%	45.9%	46.1%
	Major	35.0%	37.6%	38.3%
Major category	Minor	49.4%	50.1%	50.2%
	Moderate	46.1%	47.2%	47.2%
	Major	39.5%	43.0%	43.4%
	Average	38.8%	40.3%	40.4%

In an attempt to improve accuracy and better understand the system's poor performance, a number of system variations were tested. The resulting accuracies for these systems are presented in Table 6-3, where the column labeled "original accuracies" represents the original system. The first column of accuracies displays the results for a system that assumes all trained diagnoses are equally likely. As expected, the system performs worse than the original. However, the results of this test do reveal a few important insights. Note that, unlike the original, the accuracies for diagnosis by substance as well as major and minor categories increase as severity increases. The significance of this observation is that the system is indeed processing clinical effects correctly. Thus, the accuracies decreasing with increased severities in the original testing are not due to the non-linear interactions of multiple substances. Rather, the results imply that the prior probability is dominating the original diagnoses. The most likely cause for this problem is lack of quality data. Additionally, the fact that diagnosis by major category alone still displays a decreasing accuracy with increasing severity fits the explanation. Major categories cover a broad variety of substances, making it difficult to train a general model that properly fits the major category as a whole. The problem is compounded when attempting to identify two different major categories in the same diagnosis.

Table 6-3. Accuracy comparison of various systems for multiple exposure diagnosis

Diagnosed by	Exposure severity	No prior probability	Original accuracies	Double exposures	Order reversed	Primary correct
Substance	Minor	16.5%	33.5%	35.3%	42.4%	64.8%
	Moderate	17.5%	30.0%	30.9%	40.3%	63.0%
	Major	23.9%	28.3%	28.5%	39.8%	63.7%
Major & Minor categories	Minor	21.7%	47.3%	47.1%	54.0%	82.7%
	Moderate	23.0%	45.9%	45.1%	53.7%	82.9%
	Major	23.5%	37.6%	42.3%	49.4%	81.2%
Major category	Minor	24.2%	50.1%	50.8%	56.0%	81.3%
	Moderate	23.7%	47.2%	47.1%	54.5%	81.5%
	Major	22.7%	43.0%	42.0%	53.3%	80.9%
	Average	21.9%	40.3%	41.0%	49.3%	75.8%

Another issue that could contribute to the low accuracy of the system is that multiple exposure cases can consist of more than two substances. Since the system only considers the primary and secondary contributors, any additional substances involved could affect the clinical effects in a manner not normally predicted in a case only involving two substances. To improve the quality of the training data, a system was created based solely on cases where exactly two substances are involved. The system accuracy is reported in Table 6-3 under the column titled "double exposures." Although this approach improves data quality, it also reduces the amount of training cases from 8,901 to 5,149, a data reduction of over 40%. The end results yield a nominal increase in the average accuracy of only 0.7%.

Further attempts to improve accuracy resulted in two more variations of the system. The original system requires the correct identification of both primary and secondary contributors for a diagnosis to be considered successful. The first variation relaxes the constraints of the original system by allowing the order of the primary and secondary contributing substances to be reversed. Thus, diagnosing a test case with a primary contributor of A and a secondary contributor of B as having a primary contributor of B and a secondary contributor of A is considered an accurate diagnosis. As seen in Table 6-3 under the column labeled "order reversed," the relaxed diagnosis criteria increase accuracy by an average of 8.9%. Unfortunately, the resulting system is still not viable, having only achieved a maximum accuracy of 56.0%. The second variation on the original system attempted to improve accuracy by allowing the system to count any diagnosis as a correct match if the primary contributor matched the primary contributor of the test case, regardless of the secondary contributors involved. As shown in Table 6-3 under the column labeled "primary correct," this increases the system's accuracy drastically, yielding a maximum accuracy of 82.9%. It should be noted that these

results are falsely optimistic because the most common substances involved in multiple exposures are the primary contributors for many different substance combinations. As a result, a number of different possible diagnoses could be considered "correct" diagnoses for any single test case. Additionally, diagnosing multiple exposures by substance has a maximum accuracy of 64.8%, which is not an outstanding number. In spite of these shortcomings, the final system test seems to indicate that the primary contributor might be the dominating force in most multiple exposure cases. For that reason, the research presented in the following section focuses on diagnosing the primary contributor.

Diagnosing Multiple Exposures with Single Exposure Cases

The findings in the previous section seem to indicate that the clinical effects observed in most multiple exposure cases are dominated by the signs and symptoms associated with the primary contributor. To determine the truth of the matter, a system trained entirely on single exposures was tested to see if it could accurately diagnose the primary contributor in multiple exposure cases. The first column of Table 6-4 shows the accuracy of the system when diagnosing the primary contributor for every multiple exposure case. The next column shows the results when the test cases are limited to double exposures. With accuracies reaching as high as 84.9%, the results confirm that the clinical effects observed in most multiple exposure cases are indeed dominated by those associated with the primary contributor. Furthermore, the evidence indicates that the poor performance observed in the system trained solely on multiple exposure cases was not due to non-linear interactions between multiple toxins. As discussed in the previous section, the remaining explanation for the system failure is lack of sufficient data.

Table 6-4. Accuracy diagnosing primary contributors using single exposures[1]

Diagnosed by	Severity	Singles diagnosing multiples	Singles diagnosing doubles	Combined diagnosing multiples	Combined diagnosing doubles
Substance	Minor	75.4%	75.2%	79.1%	77.5%
	Moderate	77.2%	77.3%	81.1%	79.3%
	Major	78.7%	81.8%	83.5%	83.1%
Major & Minor categories	Minor	77.8%	76.4%	80.4%	78.2%
	Moderate	81.3%	80.4%	83.3%	81.6%
	Major	84.9%	84.9%	86.9%	86.2%
Major category	Minor	74.4%	74.9%	77.7%	76.9%
	Moderate	75.5%	76.2%	78.7%	78.5%
	Major	75.8%	78.3%	79.9%	80.5%
	Average	77.9%	78.4%	81.2%	80.2%

To test whether lack of data caused the poor performance observed in the system trained solely on multiple exposures, a system was trained using a combination of multiple exposures and single exposures to diagnose the primary contributor in multiple exposure cases. For training purposes, each multiple exposure was treated as a single exposure case with the primary contributor as the correct diagnosis. All single exposure cases were used for training along with approximately 90% of the multiple exposure cases. The remaining 10% of the multiple exposures were tested against the system to see if it could identify the primary contributor. The training and testing was repeated ten times to thoroughly test the system against every available multiple exposure case. In a similar manner, a system trained on a combination of double exposures and single exposures was tested to see if it could identify the primary contributor in double exposure cases. The results of these two tests are displayed in the last two columns of Table 6-4. On average, the accuracy increased by 3.3% when diagnosing multiple exposures and 1.8% when diagnosing only double exposures. These results indicate that valuable information

[1] To enable maximum comparability, minor restrictions were instated to ensure that all test runs within the same diagnosis level contained exactly the same number of trained substances on every test cycle. Explicitly, there were exactly 431 possible diagnoses for diagnosing by substance, 129 possible diagnoses for diagnosing by major and minor categories, and 60 possible diagnoses for diagnosing by major category alone.

capable of yielding greater than 80% accuracy is contained in the multiple exposure cases. Moreover, these results are consistent with the explanation that the system failure when training on multiple exposures alone was due to lack of sufficient data. It is also interesting to note that the system performed slightly better diagnosing multiple exposures, which generally should contain more extraneous clinical effects, than when diagnosing double exposures. The explanation is that training with multiple exposures included the information from approximately 8,011 cases per diagnosis cycle, whereas, training with double exposures included approximately 4,634 cases per diagnosis cycle. Presumably, having the same number double exposures as multiple exposures would result in the double exposures performing better. A similar observation can be made of the data presented in Table 6-5.

Table 6-5. Accuracy diagnosing secondary contributors using single exposures[2]

Diagnosed by	Severity	Singles diagnosing multiples	Singles diagnosing doubles	Combined diagnosing multiples	Combined diagnosing doubles
Substance	Minor	69.6%	68.6%	77.6%	75.7%
	Moderate	70.5%	69.5%	79.7%	77.6%
	Major	69.5%	70.0%	81.6%	77.0%
Major & Minor categories	Minor	67.8%	63.2%	78.3%	76.8%
	Moderate	73.0%	69.5%	82.4%	80.9%
	Major	77.6%	76.2%	86.2%	83.9%
Major category	Minor	62.1%	57.1%	71.4%	69.0%
	Moderate	64.4%	59.7%	72.9%	70.2%
	Major	67.4%	63.0%	74.3%	69.6%
	Average	69.1%	66.3%	78.2%	75.6%

The first two columns in Table 6-5 display the accuracies of a system trained solely on single exposure cases and tested against the secondary contributor for both multiple and double disorder cases. With average accuracies of 69.1% and 66.3%, the system performance is not

[2] To enable maximum comparability, minor restrictions were instated to ensure that all test runs within the same diagnosis level contained exactly the same number of trained substances on every test cycle. Explicitly, there were exactly 431 possible diagnoses for diagnosing by substance, 129 possible diagnoses for diagnosing by major and minor categories, and 60 possible diagnoses for diagnosing by major category alone.

stellar, however, it is high enough to raise a question: If the clinical effects in multiple exposure cases are dominated by the primary contributor, why is the accuracy in diagnosing the secondary contributor so high? Recall that during data cleaning all multiple exposure cases involving only products with the same generic substance code are removed from the dataset. This cleaning is only performed at the substance level. It is still likely that many multiple exposure cases consist of primary and secondary substances that share the same major and minor categories. Belonging to the same category makes it much more likely that the two substances exhibit similar clinical effects. Examining the data, it was determined that 21.0% of the primary and secondary contributors in all multiple exposure cases belonged to the same major category and 11.6% belonged to the same minor category as well. Likewise, 21.9% of all primary contributors in double exposure cases belonged to the same major category and 11.1% belonged to the same minor category. Because these cases are more likely to be diagnosed correctly based on the primary contributor, the accuracies are falsely optimistic.

The last two columns in Table 6-5 show the accuracies of a system trained on a combination of single exposures and the secondary contributors for either multiple exposures or double exposures. The addition of the secondary contributors improves the average system accuracy by 9.1% for multiple exposure diagnosis and 9.3% for double exposure diagnosis. Such a significant jump in accuracy attests that, although dominated by the primary contributor's clinical effects, secondary contributors do produce enough clinical effects that the system can be trained to at least recognize the most common multiple exposure combinations. Although some of the accuracy can be accounted for by prior probabilities, the results give hope that further research might enable reasonably accurate identification of secondary contributors.

The final step necessary to fully explore the impact of combining multiple exposure cases with single exposure cases was to train a system with the combined data and use it to diagnose only single exposure cases (Table 6-6). The first column shows the accuracy of a system trained on single exposures alone when diagnosing single exposures. The second and third columns display the accuracies for systems trained on single exposures along with the primary contributors for either multiple or double exposures. The last two columns contain the accuracies of systems trained on single exposures along with the secondary contributors for either multiple or double exposures. Interestingly, those systems trained with the primary contributors increased the average system accuracy from 74.6% to 74.9% when including multiple exposures and 75.1% when including double exposures. Although a minor increase, it is an increase nonetheless and lends further support to the conclusion that the clinical effects in multiple exposure cases are dominated by the primary contributor. Furthermore, the average accuracy for systems trained with secondary contributors decreased from 74.6% to 74.2% when including multiple exposures and 74.4% when including double exposures. A lower accuracy is to be expected since training on the secondary contributor would associate clinical effects caused by the primary contributor with the secondary contributor instead. The minimal change in accuracy can be partially explained by the multiple and double exposures that involve closely related substances from the same major and minor categories, as discussed above. Additionally, on average 33,855.3 single exposure cases were used to train the system on each cycle. The added 8,901 multiple exposure cases or 5,149 double exposure cases only account for approximately 20.8% and 13.2% of the training cases.

Table 6-6. Comparison of system accuracies when diagnosing single exposure cases[3]

Diagnosed by	Severity	Single exp alone	Singles & Multiples (primary)	Singles & Doubles (primary)	Singles & Multiples (secondary)	Singles & Doubles (secondary)
Substance	Minor	68.3%	68.2%	68.4%	68.1%	68.2%
	Moderate	77.5%	78.2%	78.0%	77.4%	77.4%
	Major	80.7%	81.4%	81.4%	80.6%	80.8%
Major & Minor categories	Minor	69.0%	68.9%	69.0%	68.6%	68.8%
	Moderate	77.6%	77.7%	78.0%	77.2%	77.5%
	Major	79.8%	80.6%	81.0%	80.6%	80.3%
Major category	Minor	68.8%	68.4%	68.9%	67.6%	67.9%
	Moderate	73.9%	74.3%	74.3%	73.4%	73.3%
	Major	76.2%	75.9%	76.8%	74.7%	75.0%
	Average	74.6%	74.9%	75.1%	74.2%	74.4%

Conclusions

This dissertation presents research performed to create a prototype knowledge-based system for diagnosing toxic exposures. A major goal of the research is to bypass the knowledge acquisition bottleneck of traditional knowledge-based systems by using data mining to automatically generate the system. Because system generation assumes no knowledge about the field of toxicology, lower accuracy percentages are to be expected; however, future research can build off this foundation and intelligently modify substance groupings to improve performance. Another important aspect of the system is the use of adjusted likelihood ratios. Likelihood ratios are mathematical calculations that are commonly known and used throughout the medical field. In this research, traditional likelihood ratios are adjusted by adding a fractional possibility to every potential outcome. The result is a robust equation that mitigates multiply-by-zero and divide-by-zero errors while rapidly converging to the same value as non-adjusted likelihood ratios. Ultimately, the system is intended to serve as a diagnostic consultant by providing

[3] To enable maximum comparability, minor restrictions were instated to ensure that all test runs within the same diagnosis level contained exactly the same number of trained substances on every test cycle. Explicitly, the average number of possible diagnoses for each testing cycle was 414.4 when diagnosing by substance, 125.6 when diagnosing by major and minor categories, and 58.1 when diagnosing by major category alone.

differential diagnoses for toxic exposure cases based on observed clinical effects. The system enables physicians to tap into the knowledge stored in poison control center databases, giving decision support information in a simple, understandable format.

Chapter 5 presented the development of the system and its subsequent testing on single exposure cases. The research explored the effects of two different filters for refining diagnosis based on a minimum number of exposure cases and a minimum number of clinical effects. System accuracy reached as high as 79.8% and increased above 80% when test cases were required to involve more than one clinical effect. Furthermore, the user interface and system operation received a positive response from two toxicologists and the diagnostic process was found to be simple and fast enough to make implementation on personal digital assistants (PDA's) a reality.

Chapter 6 continued the research by applying the system approach to multiple exposure cases. Although initial tests yielded a poor performance, further examination determined that the low accuracy was primarily due to lack of multiple exposure training cases. Further testing revealed that the clinical effects observed in multiple exposures tend to be dominated by a single substance called the primary contributor. Systems generated from a combined training set of both single exposures and primary contributors from multiple exposure cases yielded performances as high as 86.9% accuracy when diagnosing primary contributors. More specifically, 86.9% of the cases were diagnosed in the top 13 out of 129 possible major and minor category combinations.

The research performed on this system offers a number of contributions to both the field of knowledge-based systems and medicine. First, being automatically generated, the system bypasses the knowledge acquisition bottleneck of traditional knowledge-based systems. Second,

the system implements an approach to the unsolved problem of diagnosing multiple exposures. Although lack of data inhibited the diagnosis of more than one substance at a time, the system demonstrates effective diagnostic capabilities in identifying the primary contributor in multiple exposure cases. Being able to diagnose the disorder causing the most detrimental clinical effects is certainly valuable. Once the primary contributor is treated, it becomes easier to identify the other contributors in a multiple exposure case. Furthermore, there is hope that with the collection of more data the accuracy when simultaneously diagnosing multiple exposures will improve. A third contribution is the application of intelligent systems to the field of toxicology. At the present time, no American diagnostic systems exist for the field of clinical toxicology. Although systems have been implemented for France, Bulgaria, and Russia, they use different methods and are not readily available to assist American physicians. Finally, the creation of the adjusted likelihood ratio serves as a method to bridge the gap between intelligent systems and the medical field. Too often, intelligent systems fail because they use methods that are unknown and distrusted by the medical community. The adjusted likelihood ratio utilizes mathematics commonly accepted in medicine with a slight modification that creates a robust calculation without losing the essence of the original equation.

Future Work

The system presented in this dissertation is a prototype. Although the results show great promise, there is much to be done before a final system can be implemented in the real world. Recently, poison control centers (PCC's) around the United States have converted their databases from TESS standards to a new system known as the National Poison Data System (NPDS). To enable long-term growth and development of the knowledge-based consultant, the system must also be converted to the NPDS standard. Additionally, more data must be acquired through a petition to the FPIC's in Tampa and Miami and a proposal written to the national

repository. With more data in hand, the system can be thoroughly tested for diagnosing secondary contributors, both individually as well as in combination with primary contributors.

From the outset, a major objective of the research was to bypass the knowledge acquisition bottleneck by generating a knowledge-based system capable of producing meaningful and useful results without the need for an active, overseeing expert. This design principle inherently limited the designer from making any changes that required even a fundamental knowledge of toxicology. Now that the prototype is complete, several changes can be implemented for the betterment of the system. First, useless substance diagnoses, such as the "unknown drug" diagnosis, should be removed. Second, redundant substances, such as "aspirin: pediatric formulation," "aspirin: unknown if adult or pediatric formulation," and "aspirin: adult formulation," should be consolidated into a single diagnosis. Third, the category divisions should be examined by a toxicologist to create groupings based primarily on clinical effects. For example, most opioids tend to exhibit similar clinical effects, whereas, the effects associated with spider bites vary greatly depending on the species of spider. Intelligently restructuring diagnosis groupings could greatly increase the accuracy and utility of the knowledge-based consultant.

After refining the system, the next step is to field test the system in a PCC. Based on these results, further improvements can be implemented. One possible concern is that, although the system may perform well on toxic exposure cases as a whole, it may be more beneficial for the system to specialize on more difficult and deadly problems. In other words, it may be better to sacrifice accuracy on simple, routine exposures to increase the accuracy of the system on exposures that are dangerous and difficult to diagnose.

Once the system is fully tested, there will be freedom to expand in various directions. The general system approach could be applied to other domains, particularly those in the medical

field. The consultant could be implemented as a program on a PDA that physicians can carry with them at all times. Beyond simply diagnosing disorders, the system could be expanded by the addition of recommended treatments for each type of exposure. Once a physician makes a diagnosis, the program could serve as a reference for the treatment of the patient. Finally, the system could be converted into a program for knowledge discovery within toxicology. When training on cases in the database, the system identifies relationships between specific exposures and their clinical effects. While many of these relationships are already known, it is quite possible that the system is discovering new relationships that were previously undocumented. This is particularly true when characterizing multiple exposure cases, many of which have little documentation. Examining the relationships within a trained system could lead to new discoveries in the field of toxicology.[4]

[4] For examples of systems created to discover unknown relationships within a field, refer to Breault et al. (2002) and Brossette et al. (1998).

LIST OF REFERENCES

Abidi, S. & Manickam, S. (2002). Leveraging XML-based electronic medical records to extract experiential clinical knowledge: An automated approach to generate cases for medical case-based reasoning systems. *International Journal of Medical Informatics*, **68**, 187-203.

Althoff, K., Bergmann, R., Wess, S., Manago, M., Auriol, E., Larichev, O., Bolotov, A., Zhuravlev, Y. & Gurov, S. (1998). Case-based reasoning for medical decision support tasks: The Inreca approach. *Artificial Intelligence in Medicine*, **12**, 25-41.

Atzmueller, M., Baumeister, J. & Puppe, F. (2003a). Evaluation of two strategies for case-based diagnosis handling multiple faults. In M. Nick & K. Althoff, Eds., *Proceedings of the 2nd German Workshop on Experience Management, GWEM 2003*, Luzern, Switzerland. http://CEUR-WS.org/Vol67/, February, 2006. CEUR Workshop Proceedings.

Atzmueller, M., Baumeister, J. & Puppe, F. (2003b). Inductive Learning of Simple Diagnostic Scores. In P. Petra, R. Brause & H. Holzhutter, Eds., *Medical Data Analysis: 4th International Symposium, ISMDA 2003*, Berlin, Germany, pp. 23-30. Berlin: Springer-Verlag.

Atzmueller, M., Baumeister, J. & Puppe, F. (2004a). Quality measures for semi-automatic learning of simple diagnostic rule bases. In D. Seipel, M. Hanus, U. Geske & O. Bartenstein, Eds., *15th International Conference on Applications of Declarative Programming and Knowledge Management, INAP 2004 and 18th Workshop on Logic Programming, WLP 2004*, Potsdam, Germany, pp. 65-78. Berlin: Springer-Verlag.

Atzmueller, M., Baumeister, J., Puppe, F., Shi, W. & Barnden, J. (2004b). Case-based approaches for diagnosing multiple disorders. In V. Barr & Z. Markov, Eds., *Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004*, Miami Beach, Florida, pp. 154-159. Menlo Park, California: AAAI Press.

Au, W. & Chan, K. (2003). Mining fuzzy association rules in a bank-account database. *IEEE Transactions on Fuzzy Systems*, **11**, 238-248.

Baumeister, J., Seipel, D. & Puppe, F. (2001). Incremental development of diagnostic set-covering models with therapy effects. In G. Kern-Isberner, T. Lukasiewicz & E. Weydert, Eds., *Proceedings of the KI-2001: Workshop on Uncertainty in Artificial Intelligence*, Vienna, Austria.

Baumeister, J., Atzmueller, M., & Puppe, F. (2002). Inductive Learning for Case-Based Diagnosis with Multiple Faults. In S. Craw & A. Preece, Eds., *Advances in Case-Based Reasoning: Proceedings of the 6th European Conference on Advances in Case-Based Reasoning*, Aberdeen, Scotland, pp. 28-42. Berlin: Springer-Verlag.

Ben-Bassat, M., Carlson, R., Puri, V., Davenport, M. Schriver, J., Latif, M., Smith, R., Portigal, L., Lipnick, E. & Weil, M. (1980). Pattern-based interactive diagnosis of multiple disorders: The MEDAS system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2**, 148-160.

Ben-Bassat, M. Campell, D., MacNeil, A. & Weil, M. (1983). Evaluating multimembership classifiers: A methodology and application to the MEDAS diagnostic system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **5**, 225-229.

Bradley, P., Fayyad, U. & Reina, C. (1998). Scaling clustering algorithms to large databases. In R. Agrawal, P. Stolorz & G. Piatetsky, Eds., *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, KDD-98*, New York, pp. 9-15. Menlo Park, California: AAAI Press.

Breault, J., Goodall, C. & Fos, P. (2002). Data mining a diabetic warehouse. *Artificial Intelligence in Medicine*, **26**, 37-54.

Brossette, S., Sprague, A., Hardin, J., Waites, K., Jones, W. & Moser, S. (1998). Association rules and data mining in hospital infection control and public health surveillance. *Journal of the American Medical Informatics Association*, **5**, 373-381.

Buchanan, B. & Shortliffe, E. (1984a). *Rule-Based Expert Systems*. Reading, Massachusetts: Addison-Wesley Publishing Company.

Buchanan, B. & Shortliffe, E. (1984b). Uncertainty and evidential support. In B. Buchanan & E. Shortliffe, Eds., *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, pp. 209-232. Reading, Massachusetts: Addison-Wesley Publishing Company.

Cios, K. & Moore, G. (2002). Uniqueness of medical data mining. *Artificial Intelligence in Medicine*, **26**, 1-24.

Darmoni, S., Massari, P., Droy, J., Mahe, N., Blanc, T., Moirot, E. & Leroy, J. (1994). SETH: An expert system for the management on acute drug poisoning in adults. *Computer Methods and Programs in Biomedicine*, **43**, 171-176.

Darmoni, S., Massari, P., Droy, J., Blanc, T. & Leroy, J. (1995). Functional evaluation of Seth: An expert system in clinical toxicology. In P. Barahona, M. Stefanelli & J. Wyatt, Eds., *Artificial Intelligence in Medicine: 5th Conference on Artificial Intelligence in Medicine Europe, AIME '95 Proceedings*, Pavia, Italy, pp. 231-238. Berlin: Springer-Verlag.

Delgado, M., Sanchez, D., Martin-Bautista, M. & Vila, M. (2000). Mining association rules with improved semantics in medical databases. *Artificial Intelligence in Medicine*, **21**, 241-245.

Dempster, A. (1967). Upper and lower probabilities induced by a multi-valued mapping. *Annals of Mathematical Statistics*, **38**, 325-399.

Duda, R., Hart, P. & Stork, D. (2001). *Pattern Classification*, 2nd Ed. New York: John Wiley & Sons, Inc.

Florida Poison Information Center Network. (2005). FPIN statewide annual reports, calendar year (Jan-Dec) 2004: General call summary report. http://fpicn.jax.ufl.edu/Data/Reports/Calls_state_2004.pdf, December 2005. Florida Poison Information Center Jacksonville.

Gonzalez, A. & Dankel, D. (1993). *The Engineering of Knowledge-based Systems Theory and Practice*. Englewood Cliffs, New Jersey: Prentice Hall.

Graefe, G., Fayyad, U. & Chaudhuri, S. (1998). On the efficient gathering of sufficient statistics for classification from large SQL databases. In R. Agrawal, P. Stolorz & G. Piatetsky, Eds., *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, KDD-98*, New York, pp. 204-208. Menlo Park, California: AAAI Press.

Han, J. & Kamber, M. (2001). *Data Mining Concepts and Techniques*. San Francisco: Morgan Kaufmann Publishers.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, Michigan: University of Michigan Press.

Holsheimer, M., Kersten, M., Mannila, H. & Toivonen, H. (1995). A perspective on databases and data mining. In U. Fayyad & R. Uthurusamy, Eds., *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining, KDD-95*, Montreal, Canada, pp. 150-155. Menlo Park, California: AAAI Press.

Kluza, A. (2004). Veterinary toxicology information system. *TASK Quarterly*, **2**, 297-301.

Kolodner, J. (1993). *Case-Based Reasoning*. San Mateo, California: Morgan Kaufmann Publishers.

Kononenko, I., Bratko, I. & Kukar, M. (1998). Application of machine learning to medical diagnosis. In R. Michalski, I. Bratko & M. Kubat, Eds., *Machine Learning and Data Mining: Methods and Applications*, pp. 389-428. New York: John Wiley & Sons, Inc.

Koton, P. (1988). Reasoning about evidence in causal explanations. In *Proceedings of the 7th National Conference on Artificial Intelligence*, *AAAI-88*, St. Paul, Minnesota, pp. 256-261. Los Altos, California: Morgan Kaufmann Publishers.

Kusiak, A., Kern, J., Kernstine, K. & Tseng, B. (2000). Autonomous-decision making: A data mining approach. *IEEE Transactions on Information Technology in Biomedicine*, **4**, 274-284.

Kusiak, A., Law, I. & Dick, M. (2001). The G-algorithm for extraction of robust decision rules—Children's postoperative intra-atrial arrhythmia case study. *IEEE Transactions on Information Technology in Biomedicine*, **5**, 225-235.

Lavrac, N. (1999). Selected techniques for data mining in medicine. *Artificial Intelligence in Medicine*, **16**, 3-23.

Liu, Z. & Yan, F. (1997). Fuzzy neural network in case-based diagnostic system. *IEEE Transactions on Fuzzy Systems*, **5**, 209-222.

Medical University of South Carolina. (2000). Sensitivity and Specificity. http://www.musc.edu/dc/icrebm/sensitivity.html, February 2006. Medical University of South Carolina.

Miller, R., Pople, H & Myers, J. (1982). INTERNIST-I, an experimental computer-based diagnostic consultant for general internal medicine. *The New England Journal of Medicine*, **307**, 468-476.

Monov, A., Iordanova, I., Zagorchev, P., Vassilev, V., Nissimov, M., Kojuharov, R., Tconev, R. & Damianov, V. (1992). MEDICOTOX CONSILIUM – An expert system in clinical toxicology. In K. Lun, P. Degoulet, T. Piemme & O. Rienhoff, Eds., *Proceedings of the 7th World Congress on Medical Informatics, MEDINFO 92*, Geneva Palexpo, Switzerland, pp. 610-614. Amsterdam: Elsevier Science Publishers.

Nechyba, M. (2003). Introduction to feedforward neural networks. http://www.mil.ufl.edu/courses/eel5840/classes/intro_neural_networks.pdf, February 2006. Machine Intelligence Laboratory, University of Florida.

Nilsson, M. & Sollenborn, M. (2004). Advancements and trends in medical case-based reasoning: An overview of systems and system development. In V. Barr & Z. Markov, Eds., *Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004*, Miami Beach, Florida, pp. 178-183. Menlo Park, California: AAAI Press.

Nilsson, N. (1998). *Artificial Intelligence: A New Synthesis*. San Francisco, California: Morgan Kaufmann Publishers.

Ohmann, C., Franke, C. & Yang, Q. (1999). Clinical benefit of a diagnostic score for appendicitis: Results of a prospective interventional study. *Archives of Surgery*, **134**, 993-996.

Onisko, A., Druzdzel, M. & Wasyluk, H. (2000). Extension of the HEPAR II model to multiple-disorder diagnosis. In M. Klopotek, M. Michalewicz & S. Wierzchon, Eds., *Intelligent Information Systems*, pp. 303-313. Heidelberg: Physica-Verlag.

Onisko, A., Druzdzel, M. & Wasyluk, H. (2001). Learning Bayesian network parameters from small sets: Applications of Noisy-OR gates. *International Journal of Approximate Reasoning*, **27**, 165-182.

Owens, D. & Sox, H. (2001). Medical decision-making: Probabilistic medical reasoning. In E. Shortliffe, L. Perreault, G. Wiederhold & L. Fagan, Eds., *Medical Informatics: Computer Applications in Health Care and Biomedicine*, pp. 76-131. New York: Springer-Verlag.

Pawlak, Z. (1982). Rough sets. *International Journal of Computer & Information Sciences*, **11**, 341-356.

Peng, Y. & Reggia, J. (1986). Plausibility of diagnostic hypotheses: The nature of simplicity. In *Proceedings of the 5th National Conference on Artificial Intelligence, AAAI-86*, Philadelphia, Pennsylvania, pp. 140-145. Los Altos, California: Morgan Kaufmann Publishers.

Peng, Y. & Reggia J. (1987). A probabilistic causal model for diagnostic problem solving – Part I: Integrating symbolic causal inference with numeric probabilistic inference. *IEEE Transactions on Systems, Man, and Cybernetics*, **2**, 146-162.

Peng, Y. & Reggia, J. (1989). A comfort measure for diagnostic problem solving. *Information Sciences*, **47**, 149-184.

Pople, H. (1977). The formation of composite hypotheses in diagnostic problem solving: An exercise in synthetic reasoning. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence, IJCAI-77*, Cambridge, Massachusetts, pp. 1030-1037. Pittsburgh, Pennsylvania: Carnegie-Mellon University.

Pople, H. (1985a). CADUCEUS: An experimental expert system for medical diagnosis. In P. Winston & K. Prendergast, Eds., *The AI Business: The Commercial Uses of Artificial Intelligence*, pp. 67-80. Cambridge, Massachusetts: The MIT Press.

Pople, H. (1985b). Evolution of an Expert System: From Internist to Caduceus. In I. de Lotto & M. Stefanelli, Eds., *Proceedings of the International Conference on Artificial Intelligence in Medicine*, Pavia, Italy, pp. 179-208. Amsterdam: Elsevier Science Publishers.

Portinale, L. & Torasso, P. (1995). ADAPtER: An integrated diagnostic system combining case-based and abductive reasoning. In M. Veloso & A. Aamodt, Eds., *Proceedings of the 1st International Conference on Case-Based Reasoning Research and Development, ICCBR-95*, Sesimbra, Portugal, pp. 277-288. Berlin: Springer-Verlag.

Quinlan, J. R. (1996). Bagging, boosting, and C4.5. In *Proceedings of the 13th National Conference on Artificial Intelligence, AAAI-96*, Portland, Oregon, pp. 725-730. Menlo Park, California: AAAI Press.

Reggia, J., Nau, D. & Wang, P. (1983). Diagnostic expert systems based on a set covering model. *International Journal of Man-Machine Studies*, **19**, 437-460.

Rumelhart, D. & McClelland, J. (1986). *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*. Cambridge, MA: MIT Press.

Tsumoto, S. (2000). Automated discovery of positive and negative knowledge in clinical databases. *IEEE Engineering in Medicine and Biology*, **19**, 56-62.

Valdes-Perez, R. (1999). Principles of human-computer collaboration for knowledge discovery in science. *Artificial Intelligence*, **107**, 335-346.

van der Gaag, L. & Wessels, M. (1994). Efficient multiple-disorder diagnosis by strategic focusing. Technical report. UU-CS-1994-23.

Vinterbo, S. & Ohno-Machado, L. (2000). A genetic algorithm approach to multi-disorder diagnosis. *Artificial Intelligence in Medicine*, **18**, 117-132.

Wang, L. (2003). The WM Method completed: A flexible fuzzy system approach to data mining. *IEEE Transactions in Fuzzy Systems*, **11**, 768-782.

Watson, W., Litovitz, T., Rodgers, G., Klein-Schwartz, W., Reid, N., Youniss, J., Flanagan, A. & Wruk, K. (2004). 2004 annual report of the American Association of Poison Control Centers Toxic Exposure Surveillance System. *The American Journal of Emergency Medicine*, **23**, 589-666.

Wu, T. (1990). Efficient diagnosis of multiple disorders based on a symptom clustering approach. In *Proceedings of the 8th National Conference on Artificial Intelligence*, *AAAI-90*, Boston, Massachusetts, pp. 357-364. Menlo Park, California: AAAI Press.

Wu, T. (1991). A problem decomposition method for efficient diagnosis and interpretation of multiple disorders. *Computer Methods and Programs in Biomedicine*, **35**, 239-250.

Yu, V., Fagan, L., Bennett, S. Clancey, W., Scott, A., Hannigan, J., Buchanan, B., & Cohn, S. (1984). An Evaluation of MYCIN's advice. In B. Buchanan & E. Shortliffe, Eds., *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, pp. 589-596. Reading, Massachusetts: Addison-Wesley Publishing Company.

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, **8**, 338-353.

Zhou, Z. (2003). Three perspectives of data mining. *Artificial Intelligence*, **143**, 139-146.

BIOGRAPHICAL SKETCH

Joel Daniel Schipper was born in 1979 to W. Thomas and Harriet Anne Schipper. He grew up in the suburbs of Los Angeles with his two older brothers, Tom and James. Although an excellent student, he much preferred spending his time on the athletic field than studying. Joel attended Loyola Marymount University as a Presidential Scholar and graduated summa cum laude with a Bachelor of Science in Electrical Engineering. He continued his studies as an Alumni Fellow at the University of Florida where he received a Master of Science in Electrical Engineering. During his time at the University of Florida, he met and married Alice Eileen Brown. He is currently pursuing his doctorate by writing this dissertation, though he would much rather be outside playing.

Upon completion of his doctoral degree, Joel will join the faculty of Bradley University as an Assistant Professor of Electrical and Computer Engineering.

+ AMDG +