CONSTRUCTING PREDICTIVE MODELS TO ASSESS THE IMPORTANCE OF VARIABLES IN EPIDEMIOLOGICAL DATA USING A GENETIC ALGORITHM SYSTEM EMPLOYING DECISION TREES

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

ANAND TAKALE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

AUGUST 2004

UNIVERSITY OF MINNESOTA

This is to certify that I have examined this copy of master's thesis by

Anand Takale

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

_____

Name of Faculty Adviser(s)

_____

Signature of Faculty Adviser(s)

_____

Date

GRADUATE SCHOOL

Acknowledgments

I would like to take this opportunity to thank those who helped me immensely throughout this thesis work. People whose efforts, I will always appreciate and remember.

Dr. Rich Maclin, for his guidance - be it technical or otherwise, valuable suggestions, tireless energy to review the work, and for inspiring and helping in all possible ways and at any time.

Dr. Tim Van Wave for sharing his domain expertise, sparing his time to review the work, and for providing constructive feedback of our work.

The members of my thesis committee, Dr. Hudson Turner and Dr. Mohammed Hasan for their suggestions and for evaluating this thesis work.

All the faculty and staff of Computer Science department at University of Minnesota Duluth for their assistance throughout my stay at Duluth.

All my family members and friends for their support.

Abstract

With an ever-growing amount of data produced in a wide range of disciplines, there is an increasing need for effective, efficient and accurate algorithms to discover interesting patterns in the data. In many datasets, not all the features contain useful information. In this work, we attempt to build a system using genetic algorithms and decision trees to construct a predictive model which identifies good, small subsets of features with high classification accuracy and establishes relationships within a dataset. Our system uses a decision tree based preprocessing technique to discard likely irrelevant features.

The system that we have created, which uses survey datasets, employs a genetic algorithm combined with decision trees. In our testing, it effectively addresses the problem of identifying predictors of cardiovascular disease risk factors and mental health status variables and discovering interesting relationships within the data, especially between cardiovascular disease risk factors and mental health status variables. We also identify a set of parameters of genetic algorithms for which accurate data models are obtained. We believe that our system can be applied to various epidemiological datasets to construct meaningful predictive data models. We believe that the results obtained from our system may enable physicians and health professionals to intervene early in addressing cardiovascular disease risk factors and mental health issues and reduce risks of both conditions effectively and efficiently.

Contents

List of Tables:

List of Figures:

Chapter 1

Introduction

In recent years, we have seen a dramatic increase in the amount of data collected. Data is collected by numerous organizations and institutes to extract useful information for decision making using various data-mining algorithms. The ever-growing amount of data has prompted a search for effective, efficient and accurate algorithms to extract useful information from the data. Generally a dataset consists of large number of features, but only some of the features in the dataset contain useful information. The feature subset selection process is an useful process to find an optimal subset of features that contain useful information. In this thesis, we construct easily comprehensible predictive models using a genetic algorithm system employing decision trees, which assesses importance of features in the dataset, identifies optimal subsets of features and establishes relationships within the data. In particular, we apply our system to epidemiological datasets as they contain a large number of features that contain irrelevant information..

In order to increase public awareness about health risks, several collaboratives and organizations have collected large amounts of data using surveys. A lot of interesting information can be inferred from the collected medical survey data. Statistical methods are frequently used to find patterns in the data [Manilla, 1996] as they are easy to implement. However, they do not reflect every aspect of relationships in the data, especially relationships involving multiple features. Currently, machine learning techniques, which make use of statistical properties, are used increasingly for analyzing large datasets [Manilla, 1996]. Using machine learning techniques we can view data from several perspectives and find complex relationships within the data.

According to statistics released by the American Heart Association [AHA, 2004], cardiovascular diseases have been the main cause of deaths in United States since 1900 except for the year 1918. Statistics [NIMH, 2004] also indicate depression is one of the most commonly diagnosed psychiatric illnesses in United States. Studies [Davidson et al., 2001, Ford et al., 1998, O'Connor et al., 2000, Pratt et al., 1996] have linked clinical depression with chronic

illnesses and life threatening diseases. Understanding the relationship between risk factors associated with cardiovascular disease and poor mental health can lead to prevention and early intervention for both conditions.

In this thesis we present a machine learning system which attempts to identify predictors of cardiovascular disease risk factors and mental health status variables. We attempt to derive a predictive model which draws inferences from survey data to identify predictors of cardiovascular disease risk factors and mental health in a regional population. In this research we further attempt to establish relationships within the data. We especially try to identify the relationship between cardiovascular disease risk factors and mental health which might help physicians and health professionals to intervene early in addressing mental health issues and cardiovascular disease risks.

1.1 Machine Learning and Knowledge Discovery in Databases

Learning is a process of improving knowledge through experience. Humans learn from their experience so that they perform better in similar situations in the future and do not repeat their mistakes. This experience comes either from a teacher or through self-study. A more concrete definition of learning which is widely accepted by psychologists is as follows:

> *Learning is a process in which behavior capabilities are changed as the result of experience, provided the change cannot be accounted for by native response tendencies, maturation, or temporary states of the organism due to fatigue, drugs, or other temporary factors* [Runyon, 1977].

Machine learning is a process of attempting to automate the human learning process using computer programs. The goal of a machine learning algorithm is to learn from experience and build a model that can be used to perform similar tasks in the future. A machine learning system is a system capable of autonomous acquisition and integration of knowledge. In recent years many successful machine learning algorithms have been developed ranging from learning to perform credit risk assessment [Gallindo and Tamayo, 1997] to autonomous vehicles that learn to drive on highways [Pomerleau, 1995].

Recently, the capacity for generating and collecting data has increased rapidly. Millions of databases are available in business management, government administration, scientific and engineering data management, traffic control, weather prediction, diagnosis of patients and many other applications. The number of such databases is increasing rapidly because of the availability of powerful and affordable database systems. This explosive growth in data and databases has generated urgent need for new techniques and tools that can intelligently and automatically process the data into useful information [Chen et al., 1996]. Knowledge discovery in databases (sometimes called data-mining) makes use of concepts from artificial intelligence to extract useful information from the databases and in combination with machine learning techniques produce efficient predictive models.

1.2 Thesis Statement

In this research we propose to build a system employing genetic algorithms to perform feature subset selection to reduce the number of features used in constructing the learning model while maintaining the desired accuracy. In this predictive model, we use decision trees as a base classifier to construct the learning model and reflect the relationship within the data. We further attempt to identify the factors affecting the performance of the genetic algorithms. In this thesis, we also compare the results obtained from our proposed model with the results obtained from traditional statistical methods. As a test of our method we attempt to construct a predictive model to identify predictors of cardiovascular disease risk factors and mental health status in a regional population from a survey dataset.

The aims of the study of cardiovascular disease risk factors and mental health status are:

- Identification of predictors of cardiovascular disease risk factors and predictors of mental health status of a regional population.
- Examination of relationships between cardiovascular disease risk factors and mental health status in a regional population.
- Identification of an optimal set of parameters of the genetic algorithms for the survey dataset.

1.3 Thesis Outline

The thesis is organized as follows. Chapter 2 presents background information for our system, introduces decision tree learning and genetic algorithms, and various related concepts. Chapter 3 introduces the dataset used to find the predictors of cardiovascular disease risk factors and mental health status of a regional population. Chapter 4 presents the proposed solution and the design of the system. It discusses the preprocessing technique, the feature subset selection procedure and the construction of the predictive model. Chapter 5 describes various experiments that were conducted and the results obtained from these experiments. It further presents a comparison of the results obtained by our proposed system with the results obtained from traditional statistical methods. Chapter 6 discusses research related to this work. Chapter 7 discusses future improvements that can be done to the proposed system. Finally, Chapter 8 summarizes the main findings of this work and concludes the thesis.

Chapter 2
Background

This chapter discusses background for this research. In the first section we introduce decision trees, a method used in supervised machine learning. We discuss the representation of decision trees, the decision tree learning algorithm and related concepts. Finally we give an illustrative example showing the decision tree learning process. In the second section we discuss genetic algorithms, which are an effective technique in machine learning to perform randomized stochastic search. In this section we first discuss the theory of natural evolution as proposed by Charles Darwin [Darwin, 1859] and how the theory of artificial evolution borrows concepts from natural evolution. Then we describe the genetic algorithm learning process, genetic operators and different selection mechanisms. Finally we discuss the parameters of genetic algorithms.

2.1 Decision Trees

Decision tree learning [Breiman et al., 1984, Quinlan, 1986] is one of the most popular and widely used algorithms for inductive learning [Holland et al., 1986]. Decision trees are powerful and popular tools for classification and prediction. A decision tree is used as a classifier for determining an appropriate action from a set of predefined actions. Decision trees are an efficient technique to express classification knowledge and to make use of the learned knowledge. Decision tree learning algorithms have been applied to a variety of problems ranging from the classification of celestial objects in satellite images [Salzberg et al., 1995] to medical diagnosis [Demsar et al., 2001, Kokol et al., 1994] and credit risk assessment of loan applicants [Gallindo and Tamayo, 1997].

2.1.1 Decision Trees Representation

As shown in Figure 2.1, a decision tree is a *k-ary* tree where each of the internal nodes specifies a test on some feature from the input features used to represent the dataset. Each branch descending from a node corresponds to one of the possible values of the features specified at that node. Decision trees classify instances by sorting them down from the root node to a leaf node.

An instance is classified by recursively testing the feature value of the instance for the feature specified by that node starting from the root node and then moving down the corresponding branch until a leaf node is reached. A classification label is one of the possible values of the output feature (i.e., the learned concept). Every leaf node is associated with a classification label and every test instance receives the classification label of the corresponding leaf. All the internal nodes are represented by *ovals* and all the leaf nodes are represented by *rectangles.*

For better understanding of decision trees, consider a hypothetical case where you have to predict the outcome of a basketball match played by the Moonriders basketball team. The decision tree in Figure 2.1 can be used to try to predict the outcome of the game. The decision tree can be constructed if you have sufficient data pertaining to the previous performances of the team and the outcomes of the previous games.

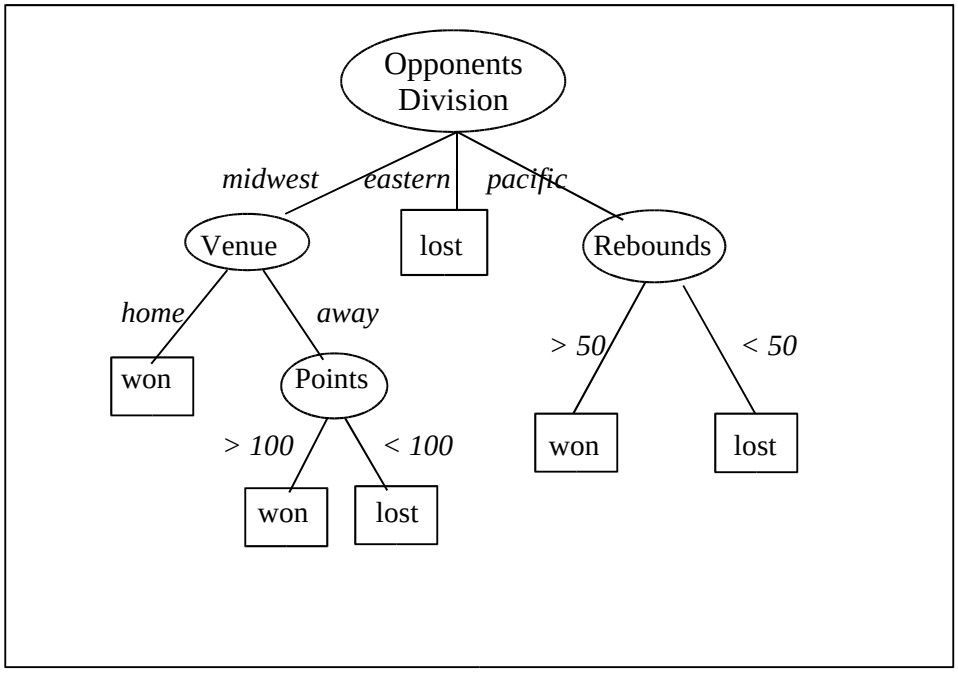

Figure 2.1: A decision tree to predict the outcome of a Moonriders basketball game.

Consider the following example:

(*Opponents Division =midwest, Points>100=yes, Venue=home, Rebounds>50=no, Opponents Points>100=yes,Opponents Rebounds>50=no)*

Classification starts at the root node of the decision tree. At the root node the feature *Opponents Division* is tested and sorted down the branch corresponding to *midwest* in the decision tree. Then at the second level of the decision tree the feature *Venue* is tested and is sorted down the left branch (corresponding to *Venue = home*) to a leaf node. When an instance is classified down to a leaf node, it is assigned the classification label associated with the leaf node (*Outcome = won*). Thus the decision tree predicts that if Moonriders are playing a home game against opponents from Midwest division, then they will win the game.

The decision tree can also be represented as a set of *if-then* rules. The learned decision tree represented in Figure 2.1 can also be represented as *if-then* rules as shown in Table 2.1. A decision tree can also be represented as a disjunction of conjunctive expressions. Each path in the decision tree from the root node to the leaf node is a conjunction of constraints on feature values and all such paths from the root to a leaf node form a disjunction of conjunctions.

Table 2.1: The *If-then* rules corresponding to learned decision tree in Figure 2.1.

if *Opponents Division=eastern* then *Outcome=lost*
if *Opponents Division = midwest* and *Venue = home* then *Outcome = won*
if *Opponents Division=midwest* and *Venue=away* and *Points>100* then *Outcome=won*
if *Opponents Division=midwest* and *Venue=away* and *Points<100* then *Outcome=lost*
if *Opponents Division=pacific* and *Rebounds>50* then *Outcome=won*
if *Opponents Division=pacific* and *Rebounds<50* then *Outcome=lost*

Table 2.2: Decision tree rules from the decision tree in Figure 2.1 represented as a disjunction of conjunctive expressions.

(Opponents Division=midwest ∧ Venue=home)
∨ *(Opponents Division=midwest ∧ Venue=away ∧ Points>100)*
∨ *(Opponents Division=pacific ∧ Rebounds>50)*

Thus decision trees represent a disjunction of conjunctions on the feature values of instances. For example, the paths from the learned decision tree that represent *winning* outcomes in Figure 2.1 can be represented as shown in Table 2.2.

2.1.2 Decision Tree Learning Algorithms

Most decision tree learning algorithms employ a top-down, greedy search through the space of possible decision trees. The ID3 decision tree learning algorithm [Quinlan, 1986] is a basic decision tree learning algorithm around which many of the variants of decision tree learning algorithms have been developed. However, due to various limitations of the ID3 learning algorithm, it is seldom used. The C4.5 decision tree learning algorithm [Quinlan, 1993] replaced the ID3 algorithm by overcoming many of the limitations of the ID3 decision tree learning algorithm. The C5.0 learning algorithm [Quinlan, 1996] is a commercial version of this family of algorithms. C5.0 incorporates newer and faster methods for generating learning rules, provides support for boosting [Freund and Schapire, 1996] and has the option of providing non-uniform misclassification costs. C5.0 and CART [Breiman et. al., 1984] are the most popular decision tree algorithms in use. The CART (Classification And Regression Trees) algorithm is used on a large scale in statistics. In this section we briefly describe the ID3 decision tree learning algorithm.

Table 2.3 summarizes the ID3 decision tree learning algorithm. First, we will define some of the terms used in the algorithm. The features of the dataset can take two or more distinct values (e.g., color could be red, green or blue) or can be continuous (e.g., the weight of a person in pounds). The *target class* is the label given to each training example by a teacher (e.g., in predicting the outcome of a game, the target labels are *won* or *lost*.). The task of the algorithm is to learn the decision tree from the training data and predict the labels of the examples from the test data. *Information gain*, which is based on *entropy* (discussed in the next sub-section), is the statistical measure used in ID3 to determine how useful an feature is for classifying the training examples. When a decision tree is constructed, some feature is tested at each node and a classification label is attached to every leaf node.

The ID3 algorithm follows a greedy approach by constructing the tree in a top-down manner by choosing the 'best' possible feature from those remaining to classify the data. The

Table 2.3: An outline of the ID3 decision tree learning algorithm.

ID3 (S)

- If all examples in *S* are labeled with the same class, return a leaf node labeled with that class.
- Find information gain of every feature and choose the feature (A_T) with highest information gain.
- Partition set into *S* disjoint subsets S_1, S_2, ..., S_n. where *n* is the the number of discrete values the chosen feature can take.
- Call the tree construction process ID3(S_1), ID3(S_2) ....ID3(S_N) on each of the subsets recursively and let the decision trees returned by these recursive calls be T_1, T_2, ..., T_N.
- Return a decision tree *T* with a node labeled A_T as the root and T_1, T_2, ..., T_n as descendant of T.

ID3 algorithm chooses the 'best' feature from the remaining set of features by performing a statistical test (*information gain* in ID3) on each feature to determine how well the chosen feature alone classifies the data. The feature is 'best' in the sense that it classifies the data most accurately amongst the set of features specified in the dataset if we stop at that feature, but it is not necessarily the optimal feature for the tree. Initially a decision tree is empty. The root node is created by choosing the best feature by calculating the information gain of every feature and choosing the feature with the largest information gain. Once the root node is created, internal descendant nodes are created for every possible value of the feature tested at the root node. The training data is split at the root node depending upon the value of the feature tested. The decision tree process continues recursively and the entire procedure is repeated using the training examples associated with each descendant node to select the best feature at that point and split the data further. This process continues until the tree perfectly classifies the training examples or until all the features have been used.

The algorithm represented in Table 2.3 is a simplified ID3 algorithm. The ID3 algorithm can be extended to incorporate continuous values as well as handle missing data values. Continuous values are handled by the ID3 algorithm by dynamically defining new discrete valued features that partition the continuous feature values into a discrete set of intervals. The ID3 algorithm handles missing data values by probabilistically assigning a value based on observed frequencies of various values for the missing feature amongst the data present at the

node. The ID3 algorithm takes as input a set $S = \{<X_1,c_1>,<X_2,c_2>,.....<X_n,c_n>\}$ of training samples. The set *S* consists of training samples of the form $<X,c>$ where X is vector of features describing some case and *c* is the classification label for that training sample. The output of the algorithm is a learned decision tree.

2.1.2.1 Entropy

The main aspect of building a decision tree is to choose the 'best' feature at each node in the decision tree. Once the best feature is chosen at each node, the data is split according to the different values the feature can take. The selected feature at each node should be the most effective single feature in classifying examples. The ID3 algorithm uses a statistical measure called *information gain* to determine how effective each feature is in classifying examples. The information gain of an feature determines how well the given feature separates the training examples according to the target classification. The concept of information gain is based on another concept in information theory called *entropy*. Entropy is a measure of the expected amount of information conveyed by an as-yet-unseen message from a known set. The expected amount of information conveyed by any message is the sum over all possible messages, weighted by their probabilities.

Consider a set *S* of training examples. For simplicity we assume that the target function is boolean valued. As the decision tree has to classify the training examples into two classes, we consider these two classes of training examples as positive and negative. Hence set *S* contains positive and negative examples of some target concept. The entropy of the set *S* relative to this boolean classification is:

$$Entropy(S) = -p_+ log_2 p_+ - p_- log_2 p_- \qquad (2.1)$$

where p_+ is the proportion of positive examples in the set *S* and p_- is the proportion of negative examples in set *S*. For all calculations involving entropy we define $\log_2 0 = 0$.

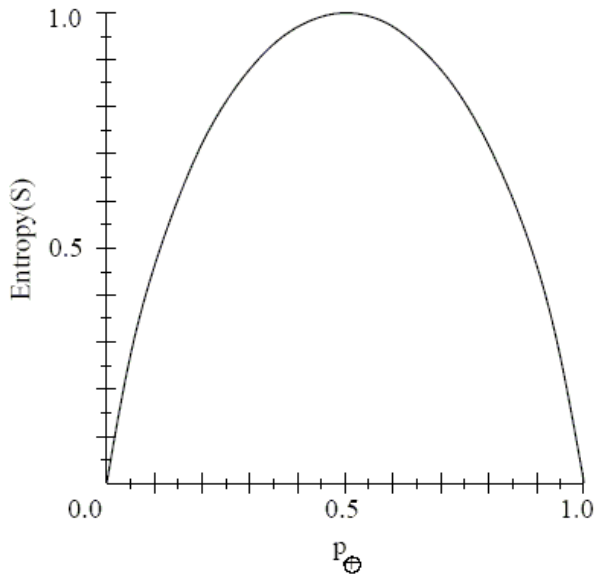

Figure 2.2: A graph showing variations in value obtained by entropy function relative to a boolean classification as the proportion of positive examples p_+ varies between 0 and 1.

Figure 2.2 shows a graph of the entropy function as the proportion of positive examples varies between 0 and 1. The entropy is 0 if all members of *S* belong to the same class. The entropy is 1 if the set *S* contains equal number of positive and negative examples. If the set *S* contains an unequal number of positive and negative examples, the entropy of the set S varies between 0 and 1. Entropy specifies the minimum number of bits of information needed to encode the classification of an arbitrary training example of set *S*. Equation (2.1) gives the expression for calculating entropy of set *S* assuming that the target function is boolean valued. The formula for calculating entropy can be easily extended to learn a target function that takes more than two values. Equation (2.2) is an extension of Equation (2.1) for calculating the entropy of a set *S* whose target function can take on *N* different values.

$$Entropy(S) = \sum_{i=1}^{N} -p_i \log_2 p_i \qquad (2.2)$$

2.1.2.2 Information Gain

Information gain is the expected reduction in entropy caused by partitioning a set of examples according to a particular feature. Information gain is used to measure the effectiveness of an feature in classifying the training data. Consider a set *S* of training examples. The information gain *Gain(S,A)* of an feature *A*, relative to a collection of examples *S* is defined as:

$$Gain(S,A) = Entropy(S) - \sum_{V \in values(A)} \frac{|S_V|}{|S|} \times Entropy(S_V) \qquad (2.3)$$

where *Entropy(S)* is the entropy of set *S, values(A)* is the set of all possible values for feature *A,* and S_V is the subset of *S* for which feature A has value *V*. *Gain(S,A)* is the information provided about the target function value, given the value of some feature A. The ID3 algorithm uses information gain as a measure to choose the best feature at each step while constructing the decision tree. The feature with the highest value of information gain is chosen and data is split into various subsets based on the values of the training examples for the chosen feature. Next, we show an example illustrating the computation of information gain of an feature.

2.1.3 An Illustrative Example

This section illustrates the working of the ID3 algorithm for an example. We continue the task of predicting the outcome of a basketball game played by the Moonriders. For predicting the outcome of the basketball game, we need some previous records of the games played by the Moonriders to construct the decision tree. The input features are *Opponents, Points, Opponents Points, Rebounds, Opponent Rebounds, Opponents Division* and *Venue.* The task is to predict the outcome of the basketball game. This is represented by the target feature *Result.* Table 2.4 shows the training examples used to train the decision tree. The ID3 algorithm determines the information gain for each of the input features. At each point it chooses the feature with maximum value of information gain. A node is created with the chosen feature, the data is tested for the chosen feature and is split according to the value of the chosen feature.

Table 2.4: Training examples for predicting the *Result* of Moonriders basketball game.

ID	Opponent	Opp. Points > 100	Venue	Opp. Rebounds > 50	Opp Division	Rebounds > 50	Points > 100	Result
1	Wolves	Yes	Home	Yes	Midwest	No	Yes	Won
2	Lakers	No	Home	No	Pacific	Yes	No	Won
3	Mavs	No	Home	No	Pacific	Yes	No	Won
4	Suns	No	Home	Yes	Midwest	No	Yes	Won
5	Magic	Yes	Home	Yes	Eastern	No	Yes	Lost
6	Sixers	Yes	Away	Yes	Midwest	No	No	Lost
7	Celtics	Yes	Away	Yes	Midwest	Yes	No	Lost
8	Spurs	Yes	Home	No	Eastern	No	No	Lost
9	Clippers	No	Away	Yes	Midwest	No	No	Lost
10	Sonics	Yes	Away	Yes	Midwest	Yes	Yes	Lost
11	Pistons	No	Away	Yes	Midwest	No	Yes	Won
12	Nuggets	Yes	Home	No	Eastern	Yes	Yes	Lost
13	Grizzlies	Yes	Home	Yes	Eastern	No	Yes	Lost
14	Hawks	Yes	Home	No	Pacific	No	No	Lost
15	Bobcats	No	Home	No	Pacific	No	No	Lost

To illustrate the computation of entropy consider our set *S* containing all 15 training examples. 5 examples of this training set are labeled positive *(when the outcome is won)* while 10 examples are labeled negative *(when the outcome is lost).* Then the entropy of the set *S* is computed as follows:

$$Entropy(S) = Entropy([5+,10-]) = -(5/15)\log_2(5/15) - (10/15)\log_2(10/15)$$
$$= 0.9182$$

To illustrate the computation of the information gain of an feature, consider the feature *Rebounds*, which has two possible values (less than 50, and greater than 50). We calculate the gain for feature *Rebounds* as follows:

$$Values(Rebounds) = \{less\ than\ 50\ ,\ greater\ than\ 50\}$$
$$S = [\,5+,\ 10-\,]$$
$$S_{<50} = [\,3+\ ,\ 7-\,]$$
$$S_{>50} = [\,2+\ ,\ 3-\,]$$
$$Gain(S,Rebounds) = Entropy(S) - (10/15)\,Entropy\,(S_{<50}) - (5/10)\,Entropy\,(S_{>50})$$
$$= 0.9182 - (10/15) * (0.8812) - (5/15) * (0.9708)$$
$$= 0.0072$$

Similarly the ID3 algorithm determines the information gain for all the features. The information gain for the features is as shown:

Gain(S, Venue) = 0.0853
Gain(S, Points) = 0.0065
Gain(S, Opponent Points) = 0.0165
Gain (S, Rebounds) = 0.0072
Gain(S, Opponent Rebounds) = 0.0000
Gain(S, Opponent Division) = 0.1918

As the information gain for *Opponents Division* is maximum, it is chosen as the best feature. The root node is created with *Opponent Division* as the test condition. As *Opponent Division* can take three distinct values, three descendants are created and the same process is repeated at each node until all of the training examples have the same target class or until all the features have been used up. The decision tree corresponding to the training examples in Table 2.4 is shown in Figure 2.1.

2.1.4 Overfitting and Pruning

The ID3 algorithm can suffer from overfitting of data if, for example, the training examples contain noise in the data. Mitchell [1997] defines overfitting as:

> *Given a hypothesis space H, a hypothesis $h \in H$ is said to overfit the training data if there exists some alternative hypothesis $h' \in H$, such that h has smaller error than h' over the training examples, but h' has a smaller error than h over the entire distribution of instances.*

Often overfitting occurs if the training examples contain random errors or noise. To overcome the limitations of the ID3 algorithm, Quinlan [1987] suggested a reduced-error pruning to prevent overfitting of data. The C4.5 algorithm often uses a technique called rule post-pruning proposed by Quinlan [1993]. In reduced error pruning, a decision node may be pruned by removing the subtree rooted at that node and making it a leaf node by assigning it the most common classification of the training examples associated with that node. Pruning focuses on

those nodes which result in the pruned tree performing at least as well as the original tree. Reduced-error pruning can take place while the decision tree is built, whereas rule post-pruning prunes the decision tree after it is built entirely. Rule post-pruning allows the decision tree to be built completely and sometimes prevents overfitting from occurring. Then it converts the learned decision tree into an equivalent set of rules by creating one rule for each path from the root to the leaf node. The algorithm then tries to prune the rules by generalizing each rule, removing any preconditions that result in improving the accuracy of the data. We use decision trees in this work to evaluate the solutions obtained by the genetic algorithms, which are discussed next.

2.2 Genetic Algorithms

Genetic algorithms [Holland, 1975] are stochastic search algorithms loosely based on ideas underlying the theory of evolution by natural selection [Darwin, 1859]. Genetic algorithms provide an approach to learning that is based loosely on simulated evolution and are random search methods that follow the principle of "survival of the fittest."

Genetic algorithms are useful in solving optimization problems [Michalewiz, 1992], scheduling problems [Mesman, 1995] and function-approximation problems [Hauser and Purdy, 2003]. Genetic algorithms are currently used in chemistry, medicine, computer science, economics, physics, engineering design, manufacturing systems, electronics and telecommunications and various related fields. Harp et al. [1990] used genetic algorithms in the design of neural networks to be applied to a variety of classification tasks. Schnecke and Vorberger [1996] designed a genetic algorithm for the physical design of VLSI chips. Galindo and Tamayo [1997] have applied genetic algorithms to credit risk assessment problem.

2.2.1 Natural Evolution and Artificial Evolution

Darwin in his theory of natural evolution states that evolution is a process by which populations of organisms gradually adapt themselves over time to better survive and reproduce in conditions imposed by their surrounding environment. An individual's survival capacity is determined by various features (size, shape, function, form and behavior) that characterize it. Most of these variations are heritable from one generation to the next. However, some of these

> *As many more individuals of each species are born that can possibly survive, and as consequently there is a frequently recurring struggle for existence, it follows that any being, if it vary in any manner profitable to itself, under the complex and sometime varying conditions of life, will have a better chance of survival and thus be naturally selected. From the strong principle of inheritance, any selected variety will tend to propagate its new and modified form.* Charles Darwin on origin of species [Darwin, 1859].

heritable traits are more adaptive to the environment than others, thus improving the chances of surviving and reproducing. These traits become more common in the population, making the population more adaptive to the surrounding environment. The underlying principle of natural evolution is that more adaptive individuals will win the competition for scanty resources and have better chance of surviving. According to Darwin, the fittest individuals (those with most favorable traits) tend to survive and reproduce, while the individuals with unfavorable traits would die out gradually. Over a long period of time, entirely new species are created having traits suited to particularly ecological niches.

In artificial evolution, genetic algorithms are based on the same principle as that of natural evolution. Members of a population in artificial evolution represent the candidate solutions. The problem itself represents the environment. Every candidate solution is applied to the problem and a fitness value is assigned for every candidate solution depending upon the performance of the candidate solution on the problem. In compliance with the theory of natural evolution, more adaptive hereditary traits are carried over to the next generation. The features of natural evolution are maintained by ensuring that the reproduction process preserves many of the traits of the parent solution and yet allows for diversity for exploration of other traits.

2.2.2 Working of a Genetic Algorithm

The genetic algorithm approach is a robust and efficient approach for problem solving as it represents natural systems and can adapt to wide variety of environments. A simple prototypical genetic algorithm is depicted in Table 2.5. Genetic algorithms search through a space of candidate solutions to identify the best solutions. Genetic algorithms operate iteratively

Table 2.5: An outline of a prototypical genetic algorithm.

1.[Start] Generate random population of n chromosomes (suitable initial solutions for the problem).

2.[Fitness] Evaluate the fitness function fitness(X) of each chromosome X in the population.

3.[New population] Create a new population by repeating the following steps until the new population is complete.

- **1.[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected).
- **2.[Crossover]** With some probability crossover the parents to form new offspring. If no crossover was performed, offspring is the exact copy of parent.
- **3.[Mutation]** With some probability mutate new offspring at each locus (position in chromosome).
- **4.[Accepting]** Add new offspring to the new population.

4.[Replace] Use new generated population for the next iteration of the algorithm.

5.[Test] If the end condition is satisfied, **stop**, and return the best solution in the current population.

6. [Loop] Go to step **2.**

over a set of solutions, evaluating each solution at every iteration on the basis of the fitness function and generating a new population probabilistically at each iteration.

Genetic algorithms operate on a set of candidate solutions which are generated randomly or probabilistically at the beginning of evolution. This set of candidate solutions are generally bit streams called chromosomes. The set of current chromosomes is termed a population. Genetic algorithms operate iteratively on a population of chromosomes, updating the pool of chromosomes at every iteration. On each iteration, all the chromosomes are evaluated according to the fitness function and ranked according to their fitness values. The fitness function is used to evaluate the potential of each candidate solution. The chromosomes with higher fitness values have higher probability of containing more adaptive traits than the chromosomes with lesser fitness values, and hence are more fit to survive and reproduce. A new population is then generated by probabilistically selecting the most fit individuals from the current population using a selection operator which is discussed later in the section. Some of the selected individuals may be carried forward into the next generation intact to prevent the loss of the current best solution. Other selected chromosomes are used for creating new offspring individuals by

applying genetic operators such as crossover and mutation described later in the section. The end result of this process is a collection of candidate solutions which contain members that are often better than the previous generations.

In order to apply a genetic algorithm to a particular search, optimization or function approximation problem, the problem must be first described in a manner such that an individual will represent a potential solution and a fitness function (a function which evaluates the quality of the candidate solution) must be provided. The initial potential solutions (i.e., the initial population) are generated randomly and then the genetic algorithm makes this population more adaptive by means of selection, recombination and mutation as shown in Table 2.5. Table 2.5 shows a simple genetic algorithm framework which can be applied to most search, optimization and function approximation problems with slight modifications depending upon the problem environment. The inputs to the genetic algorithm specify the population size to be maintained, the number of iterations to be performed, a threshold value defining an acceptable level of fitness for terminating the algorithm and the parameters to determine successor population. The parameters of a genetic algorithm are discussed in detail in the Section 2.2.6.

The genetic algorithm process often begins with a randomly generated population, while in some cases the initial population is generated from the training dataset. Most genetic algorithm implementations use a binary encoding of chromosomes. Different types of chromosome encodings are discussed later in Section 2.2.3. The first real iterative step starts with the evaluation of the candidate solutions. Every individual solution is evaluated by a fitness function, which provides a criteria for ranking the candidate solutions on the basis of their quality. The fitness function is specific to a problem domain and varies from implementation to implementation. For example, in any classification task, the fitness function typically has a component that scores the classification accuracy of the rule over a set of provided training examples. The value assigned by the fitness functions also influences the number of times an individual chromosome is selected for reproduction. The candidate solutions are evaluated and ranked in descending order of their fitness values. The solutions with higher fitness values are superior in quality and have more chances of surviving and reproducing.

After the candidate solutions are ranked, the selection process selects some of the top solutions probabilistically. The selection operator and various types of selection schemes used

are discussed later in the section. A certain number of chromosomes from the current population are selected for inclusion in the next generation. This process is called *elitism*; it ensures that the best solutions are not lost in the recombination process. Even though these chromosomes are included directly in the next generation, they are also used for recombination to achieve preservation of the adaptive traits of the parent chromosomes and also allow exploration of other traits. Once these members of the current generation have been selected for inclusion in the next generation population, additional members are generated using a crossover operator. Various crossover operators are discussed later in the section. In addition to crossover, genetic algorithms often also apply a mutation operator to the chromosomes to increase diversity.

The combined process of selection, crossover and mutation produces a new population generation. The current generation population is destroyed and replaced by the newly generated population (though some individuals may be carried over). This newly generated population becomes the current generation population in the next iteration. So, a random population generation is required only once, at the start of first generation, and otherwise the population generated in n^{th} generation becomes the starting population for the $n+1^{th}$ generation. The genetic algorithm process terminates at a specified number of iterations or if the fitness value crosses a specified threshold fitness value. The outcome of a genetic algorithm is a set of solutions that hopefully have a fitness value significantly higher than the initial random population. There is no guarantee that the solution obtained by genetic algorithms is optimal, however, genetic algorithms will usually converge to a solution that is very good. The following sections describe four main elements for implementing a genetic algorithm: encoding hypotheses, operators to affect individuals of population, fitness function to indicate how good the individual is, and the selection mechanism.

2.2.3 Representing a Hypothesis

To apply genetic algorithms to any problem, the candidate solutions must be encoded in a suitable form so that genetic operators are able to operate in an appropriate manner. Generally the potential solution of the problem is represented as a set of parameters and this set of

Table 2.6 : Binary encoding in bits for the *Opponent Division* feature for representing a condition predicting outcome of Moonriders basketball game.

Bit String	Corresponding condition
001	*Opponent = pacific division*
010	*Opponent = midwest division*
100	*Opponent = eastern conference*
011	*Opponent = midwest division OR pacific division*
111	*do not care condition*

parameters is encoded as chromosomes. In the traditional genetic algorithm, solutions are represented by bit strings. Binary encodings are used commonly because of their simplicity and because of the ease with which the genetic operators crossover and mutation can manipulate the binary encoded bit streams. Integer and decision variables are easily represented in binary encoding. Discrete variables can also be easily encoded as bit strings. Consider the feature *Opponent Division* from the example given in decision trees section. The feature *Opponent Division* can take on any of three values *eastern, midwest* or *pacific*. The easiest way to encode any feature into a bit stream is to use a bit string of length *N*, where *N* is the number of possible values the feature can take. For example, the feature *Opponent Division* can take on three different values, hence we use a bit string of length three. Table 2.6 shows some possible values of the encoded bit string and the corresponding conditions. Consider the following instance

Opponent Division = midwest, Points>100 = yes, Venue=home, Rebounds>50 = no,
Opponents points>100 = no, Opponents rebounds > 50 = no

The feature *Opponent Division* can be encoded as shown in Table 2.6. The feature *Venue* can take two values: *home* and *away*, hence it is encoded into a binary string using two bits represents 10 represents *home* and 01 represents *away*). The remaining features also take on two values, *yes* or *no*, so they can be encoded in the similar way as feature *Venue* is encoded (i.e. 10 represents *yes* and 01 represents *no*). Table 2.7 shows an example of a binary encoded chromosome. From the Table 2.7 we can see that the chromosome 0101010010101 represents the instance specified above.

Table 2.7: An example of binary encoded chromosome representing an example with feature values encoded as binary numbers.

Opponent, Points>100, Venue, Rebounds>50, Opp. Points>100, Opp Rebounds>50					
010	10	10	01	01	01

Continuous values are harder to encode into binary strings. In some cases continuous values are discretized by classifying the values into classes (e.g., the variable *points scored* is a continuous variable, it can be discretized into classes by assigning a 0 if *points scored* is less than 100 and 1 if *points scored* is greater than 100. It can also be classified into a larger number of classes depending upon the requirements.) In some cases continuous values are encoded directly into binary strings by actually converting the number into binary format. However to maintain fixed length strings, the precision of continuous values is restricted.

Although binary encoding is widely used in genetic algorithms, various other encodings have been proposed. Some other types that have been used thus far are permutation encoding [Mathias and Whitley, 1992], value encoding and tree encoding. Permutation encoding is used in ordering problems where every chromosome is a string of numbers that represents a position in a sequence. Mathias and Whitley [1992] used permutation encoding in solving the traveling salesman problem using genetic algorithms. Value encoding [Geisler and Manikas, 2002] is used when the solution contains real numbers which are hard to encode in binary strings. In value encoding every chromosome is a sequence of some values directly encoded in a string. Tree encoding [Koza, 1992] is used in genetic programming where every chromosome is represented as tree of objects such as functions or commands in programming language.

For our problem we are interested in finding small but accurate subsets of features. We call this the feature subset selection task. Binary encodings are used widely in feature subset selection tasks. In the feature subset selection task, the main aim is to find an optimal combination of subset of features from a set of candidate features. A binary encoding can be used to represent the subset of features. The chromosome is a binary string of length equal to the number of candidate features. A '0' in bit position *n* in the chromosome represents that the corresponding feature is not included in the subset of features, whereas a '1' in bit position *n* represents that the corresponding feature is included in the subset of features.

2.2.4 The Fitness Function

A fitness function quantifies the optimality of a solution. It evaluates all the candidate solutions and evaluates the quality of all individual solutions. It gives a criterion to rank candidate solutions which is the basis of making a decision as to whether a particular individual solution is fit to survive and reproduce. A fitness function must be devised for each problem. The fitness function takes in one chromosome at a time as input and returns a single numeric value, which is indicative of the ability or utility of the candidate solution represented by the input chromosome. The fitness function should be smooth and regular so that there is not much disparity in the fitness values of chromosomes. An ideal fitness function should neither have too many local maxima, nor a very isolated global maximum. The fitness function should correlate closely with the algorithm's goal, and should be executed quickly, as genetic algorithms must be iterated numerous times to produce useful results. For example, if the task is to learn classification rules, then the function has a component that scores the classification accuracy of the rule over a set of training examples. Fitness functions can be as simple as evaluating the distance traveled in traveling salesman problem or can be as complex as finding predictive accuracies using a classifier. In our system we compute the fitness value by computing classification accuracy and penalizing it for missing features as discussed in Section 4.3.3.4.

2.2.5 Genetic Operators

Genetic algorithms are stochastic, iterative algorithms. Thus the candidate solutions should get better with more iterations. Genetic algorithms attempt to preserve individuals with good traits (i.e., preserving individuals having high fitness values) and to create better individuals with new traits by combining fit individuals. Genetic algorithms employ genetic operators to preserve fit individuals (selection) and to explore new traits by recombining fit individuals (crossover and mutation). The function of a genetic operator is to cause chromosomes created during reproduction to differ from those of their parents in order to explore any missing traits. The recombination operators must be able to create new configurations of genes that never existed before and are likely to perform well. Below, we discuss the basic genetic operators and their variants.

2.2.4.1 Selection

At every iteration, chromosomes are recombined to create new chromosomes in an attempt to find better chromosomes. As genetic algorithms follow the theory of natural evolution, better individuals should be able to survive and reproduce. The selection operator is used to select such fit individuals from the population for recombination. Before any recombination takes place, the fittest individual solutions are selected and promoted to the next generation in an attempt to ensure that the best solution is not lost. Then the selection operator is applied again for choosing chromosomes to act as parents and produce new offspring. The selection operator is solely responsible for choosing better individuals for preservation and recombination. The selection process is one of the key factors affecting the overall performance of the genetic algorithms. If the selection mechanism selects fit individuals for elitism and recombination, then the solution converges faster. The selection process controls which fit individuals should be preserved and which individuals should be used for recombination. A bad selection mechanism could hamper the performance of genetic algorithm in terms of quality and also in terms of convergence rate. We discuss some of the popular methods for selecting a chromosome for preservation or for recombination.

Fitness Proportional Selection [Goldberg, 1989]

In fitness proportional selection, parents are selected according to their fitness value. The probability of selecting a chromosome is directly proportional to the fitness value of the chromosome. Imagine a roulette wheel where all the chromosomes in the population are placed. The size of the section in the roulette wheel is proportional to the value of the fitness function of every chromosome. The chromosomes with larger fitness values are assigned larger sections of roulette wheel and have greater probability of being selected.

Ranked Selection [Bäck and Hoffmeister, 1991, Whitley, 1989]

In ranked selection, chromosomes are sorted in descending order on the basis of their fitness function value. Once they are sorted they are assigned new fitness values based on their rankings. Fitness proportional selection does not perform as expected if there are very few chromosomes with very high fitness value and the rest of the chromosomes have very low fitness

values, because the chromosomes with high fitness values get selected very often and many traits are left unexplored. In such a situation, ranked selection performs better than fitness proportional selection, as ranked selection assigns probability of selection to every chromosome based on the ranking of the chromosome and not on the basis of the fitness value of the chromosome.

Boltzmann Tournament Selection [Blickle and Thiele, 1995, Goldberg and Deb, 1991]

In tournament selections, tournaments are conducted between sets of competing chromosomes. The competing chromosomes are chosen randomly and, once chosen, the best chromosome amongst the set of randomly chosen chromosomes is selected based on the fitness value of the chromosomes. The important parameter in tournament selection is the tournament size. If the tournament size is equal to one, then tournament selection reduces to random selection. However if the tournament size is very close to the population size, then tournament selection produces results similar to ranked selection as the chromosomes with higher rank have high probability of being selected. Tournament selection can produce good results with appropriate tournament size.

2.2.4.2 Crossover

The crossover operator produces two new offspring from two parent strings by copying selected bits from each parent. The bit at position *i* in each offspring is copied from the bit at position *i* in one of the two parents. The choice of which parent contributes the bit for position *i* is determined by an additional string called the crossover mask. In this section, we discuss some of the methods for performing crossover.

Table 2.8: An example of single-point crossover for two chromosomes.

Chromosome 1	1101100100110110
Chromosome 2	1101111000011110
Crossover Mask	1111100000000000
Offspring 1	1101111000011110
Offspring 2	1101100100110110

Table 2.9: An example of two-point crossover for two chromosomes.

Chromosome 1	110110010011011
Chromosome 2	110111100001111
Crossover Mask	111111000001111
Offspring 1	110110100001011
Offspring 2	110111010011111

Single-point crossover

In single-point crossover, the crossover point is selected randomly. The binary string from the beginning of the chromosome to the crossover point is copied from the first parent, the rest is copied from the other parent. The crossover mask is always constructed so that it begins with a string containing *n* contiguous 1's followed by the necessary number of 0's to complete the chromosome string. This results in offspring in which the first *n* bits are contributed by one parent and the remaining bits by the second parent. Table 2.8 shows an example of single-point crossover. It shows two parent chromosomes represented by bit strings and the crossover point. The crossover operator creates two offspring using the crossover mask to determine which parent contributes which bit.

Multi-point crossover

The most widely used form of multi-point crossover is two-point crossover. In two-point crossover, two bit positions are randomly selected. The binary string from one of the parent chromosome is copied from the first bit position to the second bit position, while the remaining bits (i.e., the bits from start of the string to the first bit position and from the second bit position until the end of the string) are copied from the other parent. This concept can be further extended to implement multi-point crossover by generating bit positions randomly and copying the strings from parents alternately until the next bit position is reached. Table 2.9 shows an example of two-point crossover. It shows two parent chromosomes represented by bit strings and the crossover mask. The crossover operator creates two offspring using the crossover mask to determine which parent contributes which bit.

Table 2.10: An example of uniform crossover for two chromosomes.

Chromosome 1	1101100100110110
Chromosome 2	1101111000011110
Crossover Mask	0010100010100010
Offspring 1	1101100100010110
Offspring 2	1101111000111110

Table 2.11: An example of mutation applied to a chromosome.

Original offspring	1101111000011110
Mutated offspring	1100111000011110

Uniform Crossover

Uniform crossover combines bits sampled uniformly from the two parents and is illustrated in Table 2.10. In this case the crossover mask is generated as a random bit string with each bit chosen at random and independent of the others.

2.2.4.3 Mutation

Mutation is intended to prevent early convergence of all solutions in the population into a local optimum of the solved problem. The mutation operation randomly changes the offspring resulted from crossover. The mutation operator produces small random changes to the bit string by choosing a single bit at random, then changing its value. Table 2.11 shows how some chromosomes have random mutations just as they occur in genes in nature.

2.2.5 Parameters of a Genetic Algorithm

A genetic algorithm operates iteratively and tries to adapt itself progressively over the iterations. At every iteration the genetic algorithm evaluates the population of chromosomes on the basis of it's fitness function and ranks them according to the fitness value. Genetic algorithms also apply the crossover and the mutation operator to explore new traits in chromosomes. There are various parameters defining a genetic algorithm. These parameters can be varied to obtain better performance. In this section we discuss some of the parameters that are provided as input

to the genetic algorithm. The extent to which each of the factors affects the performance of the genetic algorithm and the optimum values of the parameters for the research dataset are discussed in Chapter 5.

Crossover rate: The crossover rate specifies how often a crossover operator would be applied to the current population to produce new offspring. The crossover rate, mutation rate and selection rate determine the composition of the population in the next generation.

Mutation rate: The mutation rate specifies how often mutation would be applied after the crossover operator has been applied.

Selection rate: This parameter comes into play if elitism is applied to the genetic algorithms. When creating a new population there is a chance that the best chromosome might get lost. Elitism is a method which prevents the best chromosome from getting lost by copying a fixed percentage of the best chromosomes directly into the next generation. The selection rate specifies how often the chromosomes from the current generation would be carried over to the next generation directly by means of elitism.

Population size and generation: This parameter sets the number of chromosomes in the population at a given instance of time (i.e., in one generation). It also determines whether the initial population is generated randomly or by heuristics.

Number of iterations: This parameter dictates the stopping criteria for the genetic algorithm. Generally the stopping criteria used in genetic algorithms is the number of iterations. In some cases genetic algorithms are halted if the average fitness value crosses a certainthreshold value.

Selection type: This parameter dictates the type of selection mechanism to be used.

Crossover type: This parameter dictates the type of crossover to be used.

Chapter 3

The Bridge To Health Dataset

This chapter briefly discusses the data used in this research to test our method. We discuss the data in general, the data collection process, data design, and the statistical weighting of the dataset. We further discuss some of the salient features of the dataset and groups of features in the dataset.

3.1 Bridge to Health Dataset (BTH 2000)

The data used for this research is the Bridge to Health Survey Dataset (BTH 2000) [Block et. al., 2000]. The data was collected by the Bridge to Health Collaborative with the help of 118 organizations and individuals from the study region. The data was collected by conducting surveys of randomly chosen households in a sixteen-county region in Northwestern Wisconsin and Northeastern Minnesota. The purpose of the survey was to gather population-based health status data about the adult residents in the study region to assist health professionals in understanding the health and well-being of regional residents.

3.2 Data Collection

The BTH 2000 data was collected using computer aided telephone interviews conducted by the Survey Research Center, Division of Health Services Research and Policy located in the School of Public Health at the University of Minnesota. One adult (age 18 or older) from each sampled household was selected to participate in the survey. Proper care was taken in sampling the households and choosing the respondents to ensure representation of various age groups, economic backgrounds, ethnicities, races and educational levels. The survey was carried out between November 1999 and February 2000 and included interviews of 6,251 individuals.

3.3 Data Representation

The survey was designed to gather information pertaining to the perceived health,

diagnosed diseases, general information (such as age, sex, height, weight, education, income, health insurance status etc.) and life-style of the respondent (such as amount of food intake, amount of physical activity, amount of drinking etc). The survey consisted of 101 questions and included all questions from the Short-Form 12-Item survey (SF12) [Ware et al., 1996]. Some questions were to be answered *yes* or *no*, but generally respondents were provided with more options to answer the questions. Respondents also had a choice to refuse to answer a question or choose the option of *don't know / not sure* if the respondent was not sure about the answer to any of the questions. The data was originally represented in a SPSS data format in the form of a 2-dimensional table, consisting of 6251 data points, with each data point corresponding to the responses of an individual. Each data point is made up of 334 features, representing direct responses of an individual, recoded variables, combined variables and maintenance variables. The dataset was converted to C4.5 data format for effective and efficient usage of the data by our proposed system.

3.4 Data Description

Each data point in the The BTH 2000 dataset represents responses of individual respondents. Each data point is made up of three categories of variables: (1) direct responses of the individual respondents, (2) new and recoded variables and (3) maintenance variables. The first category of variables contain direct responses of the respondents to the 101 questions presented in the survey. The new and recoded variables are constructed from the direct responses of the respondents by discretizing a continuous variable. For example, BMI is a continuous variable representing the body mass index of an individual. BMMICUTS is a recoded variable which discretizes BMI into three classes: not overweight, overweight and obese. Variables are also constructed by combining two or more direct responses. For example, the variable MCS12 represents mental composite score of an individual. The score is computed using the values of various mental health variables. Maintenance variables consist of variables to maintain the integrity of the data. For example, the variable ID is used to index the data points. It provides unique identification of every data point. Maintenance variables also include those variables used for statistical weighting of the data. For example variables like STATEWT and COUNTYWT which represent state weight and county weight depending upon the population distribution. In general, the 334 features can be classified in groups with each group of features conveying different information. The features can be classified as follows:

- *Demographic Information*: information such as age, sex, weight, height, race, marital status, education, income and where the individual lives.
- *Life Style*: information such as general health, physical activities, eating habits, seat belt usage, and frequency of health doing checkups and drinking/smoking habits.
- *Medical History*: information such as whether an individual suffers from allergies, asthma, cancer, diabetes, back problems, high blood pressure, high cholesterol, the insurance status of an individual, and prescriptions for medicines of an individual.
- *Physical Health Status*: features such as whether the person is overweight, whether the person is suffering from any diseases and how frequently does an individual exercise.
- *Mental Health Status*: information such as whether the person was diagnosed with depression or anxiety or has other indications of poor mental health status.
- *Recoded Variables*: some variables are recoded either to make them more specific or to make them more general. For example, BMI is a continuous variable representing the body mass index of an individual. OVERWGT2 is a recoded variable which discretizes BMI into two classes: not overweight and overweight.
- *Combination of direct responses*: some of the variables were combined to form a composite variable. For example, all the mental health status variables were combined to give a mental health status composite score (MCS12). Another example is, the variable ALCWDBNO which is constructed by combining the accomplished less, careless work, downhearted/blue, depression and anxiety variables.
- *Maintenance Variables*: Maintenance variables are used to maintain the data integrity, to order the data points, and to statistically weigh the data. For example variables like ID, SRVYMODE, STATEWT and COUNTYWT.

3.5 Statistical Weighting of Data

Statistical weighting of the BTH 2000 dataset is necessary due to differences in household sizes, differences in population distribution in different counties and differences in the response rates between men and women of different ages [Block et al., 2000]. In the first step, the BTH 2000 dataset is weighted by the inverse of selection probability within the household to remove any bias created due to different household sizes. Then the BTH 2000 dataset is weighted by the ratio of adult population size in a county to the number of adults interviewed in that county, which ensures that the respondents from every county are counted towards the

results in same proportion as the population from that county contributed towards the overall population of the survey area. The third step weighs the data by a factor based on the 1998 U.S. Census to ensure that the respondents from every age and/or gender group contribute in same proportion as that of the overall population distribution. In the last step, the weights are divided by a numeric constant to ensure that the total sample size is equal to the total number of respondents in the survey area.

3.6 Salient Features

The dataset contains responses from individuals with diverse demographic profiles (i.e., individuals from different age groups, gender, education levels, poverty status, race and geographic area of residence). Almost half the respondents in the BTH 2000 dataset were male and half were female (49% were males and 51% were females). The age of the respondents varied from 18 to 99 with a mean age of 48.4 years, a median age of 46.0 years and a standard deviation of 17.9 years. The response rate of the survey was 74%, as some of the respondents refused to answer specific questions of the survey. The overall educational level of the study area was high, with 91.2% of the people having received at least a high school diploma. Another major characteristic of the survey was that 77.1% of the people in the study area lived in rural regions and 22.9% of the people lived in urban regions (mainly in the cities of Duluth, Minnesota and Superior, Wisconsin) [Block et al., 2000].

Table 3.1:Variables related to mental health status of individuals in the BTH 2000 dataset.

Variable	***Description***	***Response***
Q5_15NEW2	Diagnosed depression	1: Yes 10: No
Q5_16NEW2	Diagnosed anxiety	2: Yes 20: No
DEPRANX2	Diagnosed depression and diagnosed anxiety	2: Yes 5: No
ALDPANX2	Accomplished less without depression or anxiety	31: Yes 33: No
CWDPANX2	Careless work without depression or anxiety	31: Yes 34: No
DBDPANX2	Downhearted blue without depression or anxiety	31: Yes 35: No

3.7 Features of Interest

In this research, we are interested in identifying subsets of the features which predict cardiovascular disease risk factors or mental health status of a regional population. Table 3.1 lists the mental health status variables and Table 3.2 lists the cardiovascular disease risk factors. Our learning task is to try to find factors affecting the outcome of the variables described in Table 3.1 and Table 3.2.

Table 3.2: Variables related to cardiovascular disease risk factors of individuals in the BTH 2000 dataset

Variable	***Description***	***Response***
Q5_11	Diagnosed high blood pressure	1: Yes 2: No 7: Don't know/not sure 9: Refused
Q5_12	Diagnosed elevated cholesterol	
Q48REC	Blood pressure checked in last 2 years	1: Within the past year 2: Within the past two years 3: Within the past 5 years 4: 5 or more years ago 5: Never 7: Don't know/not sure 9: Refused
Q49REC	Cholesterol checked in last 2 years	
OVERWGT2	Overweight or not overweight based on BMI	0: Not overweight 1: Overweight
BMICUTS	Normal weight, overweight, obese based on BMI	0: Not overweight 1: Overweight 2: Obese
EXCERREC	Moderate or vigorous exercise #X per week	1: Exercise less than 3 times per week 2: Exercise more than 3 times per week
Q69A	Current smoker	1: Yes 2: No 7. Don't know/not sure 9: Refused
CHRONIC	60+ drinks per month	0: Less than 60 drinks per month 1: 60+ drinks in the past month

Chapter 4

A Genetic Algorithm System For Feature Selection

In this research, we implemented a genetic algorithm based system employing decision trees to identify small, good subsets of data with high classification accuracy. To evaluate our system we constructed a predictive model from a survey dataset to identify predictors of cardiovascular disease risk factors and mental health status in a regional population. The predictive models also establish relationships within the data, especially between cardiovascular disease risk factors and mental health status in a regional population. This chapter discusses our proposed method, which involves preprocessing and modeling of data to construct accurate predictive models. First we discuss our proposed system in brief. Then we discuss the preprocessing technique used to eliminate features carrying redundant or irrelevant information. Finally, we discuss the modeling done by our genetic algorithm system, which makes use of decision trees.

4.1 Machine Learning Modeling of Medical Variables

Large amounts of epidemiological data have been collected by various organizations and collaboratives to assist health professionals in understanding the health and well-being of individuals. Our proposed system delves into the problem of finding data patterns in epidemiological datasets and provides a solution to examine such problems and establish relationships among various features of such datasets. This section describes the modeling performed by our system. Predictive models are constructed to identify predictors of the variables in question, in this case cardiovascular disease risk factors and mental health status in a regional population. Our predictive models also explore the relationships between variables, in this case between cardiovascular disease risk factors and mental health status variables. In an attempt to build a predictive model which identifies the predictors and also determines the relationships in epidemiological data, we used a genetic algorithm system that builds decision trees as shown in Figure 4.1. The system is divided into two stages: the preprocessing stage and the feature subset selection stage.

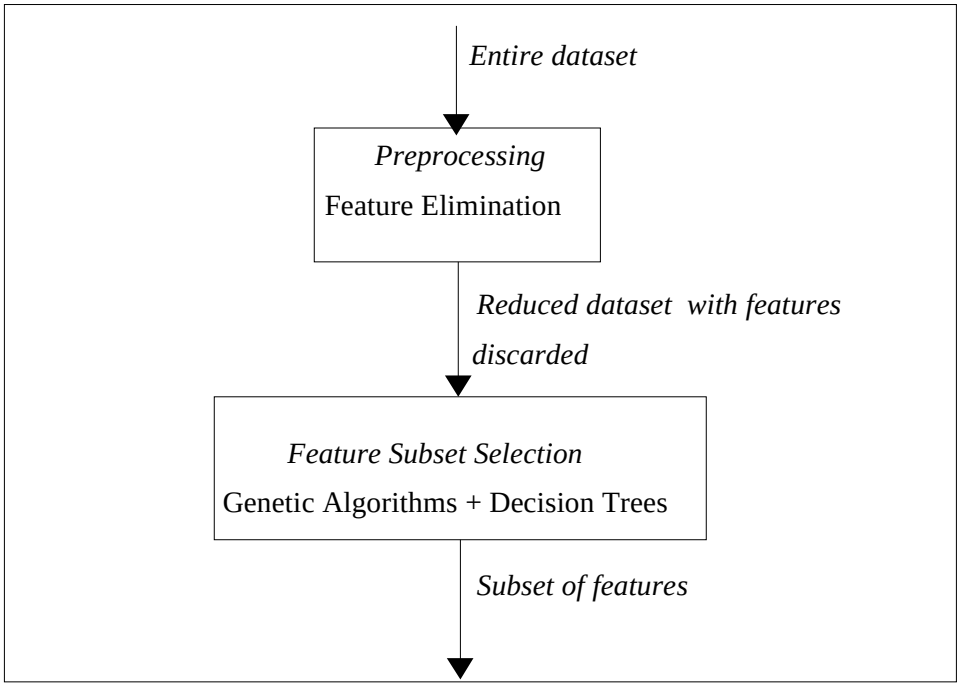

Figure 4.1: The two stages of data analysis in our system

Epidemiological datasets such as the the BTH 2000 dataset consist of a large number of features. Machine learning algorithms are time intensive for datasets with a large number of features and/or large number of data points. In addition, many of the features in the dataset convey very little or no additional information. Finally, a large number of features can lead to overfitting of data when the decision trees are constructed, resulting in higher accuracy for the training dataset, but poor overall performance. Hence we apply a preprocessing technique which discards some of the features and presents the genetic algorithm based system with a dataset containing a reduced number of features. We attempt to eliminate features in such a way that no information is lost in the feature elimination process.

To better describe our system we focus on the problem of identifying the predictors of cardiovascular disease risk factors and mental health status, though our system can be applied to any set of survey data. We propose to build a system which accurately predicts the value of one of the variables listed in Table 3.1 and Table 3.2. Our proposed system determines the predictors

for the cardiovascular disease risk factors and mental health status variables individually. The variable for which the system identifies the predictors is regarded as the output variable. The system focuses on determining small subsets of features which accurately predict the class of the output variable. Our system implements a genetic algorithm based feature subset selection method. The subsets of features are evaluated by growing decision trees, which also provide a set of rules to predict the class of the output feature. The following sections describe the two stages of the system in detail.

4.2 Preprocessing the Data

This section describes in detail the preprocessing method used. A preprocessing technique employing decision trees was used to narrow down the list of features to enable a fast and efficient feature subset selection process to construct accurate data models.

4.2.1 Motivation for feature elimination

Preprocessing of data is necessary for several reasons. The main motivating force behind reducing the number of features in the dataset is preventing overfitting of the data in the decision tree learning process. The decision tree learning algorithm can suffer from overfitting of data and implements pruning to avoid overfitting of data. Overfitting of data results in higher accuracy over the training examples, but results in poor overall future performance. The decision tree learning process is an important factor of constructing the predictive model, as it is the process which establishes relationships among features in the dataset. Hence in order to avoid overfitting of data, the system focuses on narrowing down the excessive features by discarding the irrelevant features and keeping only the useful features.

The BTH dataset consists of a large number of features. Several features in the dataset convey very little or no additional information. The BTH 2000 dataset contains some features which were used to assist the data collection and the data integration process. The dataset includes maintenance features such as ID, which represents the record number of an individual response in the dataset, SRVYMODE that indicates how the survey was carried out and numerous other features that have no bearing by definition in relation to cardiovascular disease risk factors and mental health status variables. The BTH 2000 dataset also contains some

features that are used for statistical weighting of the data (such as STATEWT, COUNTYWT, etc. to represent state weight and county weight). Such variables are based on population distribution of the survey area and have limited bearing by definition with relation to cardiovascular disease risk factors and mental health status variables. The BTH 2000 dataset also contains several recoded features. These recoded variables contain almost the same information as that of the original feature and may not convey any additional information. Hence some variables can be discarded from the dataset to avoid overfitting of data in the decision tree learning process to produce accurate learned models. Narrowing down the number of features also accelerates the hybrid decision tree based genetic algorithm system. Hence there is a need to incorporate the feature elimination process in the system to make it more efficient and accurate.

4.2.2 Feature elimination process

The feature elimination process tries to determine the importance of individual features and discards less important features while maintaining the desired accuracy. The feature elimination process uses C4.5 decision trees as a classification mechanism to predict the class of the output feature. The output feature is one of the cardiovascular disease risk factors listed in Table 3.2 or mental health status variable listed in Table 3.1. In this approach we try to classify the individual features from the dataset into three classes: very likely irrelevant features, possibly irrelevant features and possibly relevant features.

- *Very Likely Irrelevant Features:* Very likely irrelevant features are those features that convey no information about the class of the output feature that the decision tree classifier is predicting. For example, if we are trying to predict whether a person has diagnosed depression, then the feature representing the record number of an individual does not convey any information about the individual being depressed.

- *Possibly Irrelevant Features*: Possibly irrelevant features are those input features which carry very little additional information about the class of the output feature that the decision tree classifier is predicting. For example, if our learned decision tree is predicting the class of the output feature CHRONIC (whether an individual is a chronic drinker), then the feature DRPERMO (number of drinks consumed by an individual in the past month) conveys very little additional information about the class of output variable CHRONIC because of the

definition of the feature CHRONIC. The feature CHRONIC has a class '0' if DRPERMO<60 and has a class '1' if DRPERMO>60 (i.e. an individual is considered as a chronic drinker if that individual consumes more than 60 drinks in the previous month).

- *Possibly Relevant Features:* Possibly relevant features have the potential to be one of the predictors of the output feature as they contain useful information about the output feature. Features from this class may convey useful additional information about the class of the output variable.

Many epidemiological datasets such as the BTH 2000 contain large number of features. Many of these features might contain redundant or irrelevant information. In classification tasks, using all the available features in the dataset might have detrimental effect on the classification accuracy because some of the features are dependent on others while some of the features contain irrelevant information and act as noisy data. When we are using all the available features of the dataset there is more chance that some feature will randomly fit the data increasing the probability of overfitting. The dependent features contain useful information about the class of output feature, but the information they carry is redundant as they contain almost the same information about the class of the output feature as the information contained in the feature on which they depend. Hence there is a need to eliminate the dependent and noisy features from the dataset to improve the comprehensibility of the learned classification tree. Eliminating the dependent and noisy features can result into improved accuracy and clearer descriptions of learned concepts. Hence, we can construct better predictive models if all of the dependent and noisy features are removed from the dataset. In this research work, we use a preprocessing technique which assesses the importance of individual features and classifies them into one of the three categories of features discussed above.

To determine the importance of individual features and to classify the features into three distinct classes we employ a decision tree based heuristic that performs reasonably well in assessing importance of individual features and locating dependent and noisy features. The decision tree classifier uses a 10-fold cross-validation technique (discussed in Section 5.1) to compute the accuracy of classification of a data instance into an appropriate class of the output feature. Initially features such as ID, SRVYMODE, STATEWT, etc. that have no bearing by definition with relation to cardio-vascular disease risk factors and mental health status variables

were discarded from the dataset. Then the accuracy of the reduced dataset is computed by learning a decision tree. To determine the importance of a variable in terms of information conveyed by the variable, the feature is dropped from the dataset and again the accuracy is computed for the dataset with the feature dropped. The change in the accuracy is observed by taking the difference between the accuracy values before and after dropping the feature. Similarly, changes in the accuracy value is observed for every feature. Every feature is assigned a score reflecting the difference between overall accuracy and accuracy value obtained by dropping the corresponding feature. Features are ranked in descending order of their scores.

Once the changes in the accuracy values are observed and a score is assigned to every feature, the features are divided into three classes depending upon the change observed in the accuracy value. If there is no change or a very small increase in the accuracy, then the feature is classified as *very likely irrelevant feature* because even after dropping the feature, there was no impact or a negative impact on the classification accuracy indicating that the feature contains random data and provides no useful information about the class of the output feature. Features with a slight drop (we set a threshold of 0.1% drop in accuracy) in the accuracy are classified as *possibly irrelevant* features because dropping the feature does not have a significant impact on the classification accuracy. The *possible irrelevant* category of features contains dependent features which do not contain any additional information about the class of the output variable. All the features classified as *very likely irrelevant* and *possibly irrelevant* features were examined by a medical expert (Dr. Tim Van Wave, School of Medicine, University of Minnesota Duluth) to verify that those features were not meaningful from medical perspective. All of the remaining features which have significant accuracy drops (i.e., more than the threshold value of 0.1%) are classified as *possibly relevant* features. The features in the *possibly relevant* category of features contain useful information about the class of the output feature, and have significant impact on the classification accuracy. The feature elimination approach reduces the number of features used to construct the predictive models (discussed in Section 4.3) by discarding *very likely irrelevant* and *possibly irrelevant* features. All the categories of features were inspected and confirmed by a domain expert, which is mandatory, as the heuristic used by the system is purely a machine learning component and does not have any first-hand knowledge about the importance of features from medical perspective.

Out of 334 features present in the BTH 2000 dataset, 76 features were discarded because they had no bearing by definition with relation to the cardiovascular disease risk factors or mental health status variables. Out of the remaining 258 features, 86 features were eliminated by the feature elimination process. The remaining 172 variables still contained some recoded features. With the help of a domain expert (Dr. Tim Van Wave, School of Medicine, University of Minnesota Duluth) we discarded 60 more features to give a reduced dataset of 112 features. The feature subset selection method operates on the reduced dataset containing 6,251 data points and 112 features.

4.3 Building a predictive model using hybrid system

This section describes how the predictive models are built using our proposed system. We use a hybrid system which combines two machine learning techniques: genetic algorithms and decision trees to construct the predictive model. We use genetic algorithms that employ decision trees in their fitness function to construct the predictive model. Our aim is to identify predictors of cardiovascular disease risk factors and mental health status variables. This is achieved by performing feature subset selection using genetic algorithm. We also attempt to establish relationship between the output variable and the predictors. The decision tree learning algorithm is used to grow decision trees from good subsets of data to establish relationships between the input features and the output feature. The following subsections describe how the feature subset selection is carried out and how decision trees play an important role in establishing relationships between the predictors and the output variable.

4.3.1 Feature subset selection using genetic algorithms

We attempt to identify predictors of cardiovascular disease risk factors and mental health status variables. To identify the predictors we first need to identify good small subsets with high accuracy. We use genetic algorithms to perform the task of features subset selection. Table 2.5 depicts the general genetic algorithm process and explains the genetic algorithm learning process in brief. In most of the genetic algorithm based systems, the basic algorithm is implemented with slight variations. In this research, we use the same backbone of the genetic algorithm as depicted in Table 2.5. However the encoding of chromosome, selection mechanisms used, crossover mechanisms used, population recombination scheme used and the fitness function used vary

from problem to problem. Sections 4.3.2 and 4.3.3 describes in detail how the various aspects of genetic algorithm were handled to identify good subsets of data.

4.3.2 Establishing relationships using decision trees

The genetic algorithms are used to find the subset of features predicting the class of the output features listed in Table 3.1 and Table 3.2. Genetic algorithms play an important role in identifying a subset of relevant features. Decision trees also play an equally important role in the system by determining the quality of the subsets. Decision trees are used to calculate the fitness value of all the chromosomes in the population. Decision trees are used to evaluate the quality of subsets identified by the feature subset selection process. Decision trees further play an important role in interpreting the results obtained from the genetic algorithm. The results obtained from the genetic algorithms indicate only the subset of features which most accurately predict the class of the output variable. We are also interested in finding out the relationship between the subset of features and the output variable. The decision tree rules (discussed in Chapter 5) provide a complete relationship between the subset of features and the class of the output feature.

4.3.3 Encoding of feature subset selected problem as genetic algorithm.

This section describes how the feature subset selection is encoded as a genetic algorithm. We discuss how the various aspects of genetic algorithms are modified for identifying good subsets of features containing minimal number of features and having high classification accuracy.

Table 4.1 : A binary encoded chromosome for the feature subset selection. 0's indicate that the corresponding features are not included in the subset. 1's indicate that the corresponding features are included in the subset.

Chromosome	00010000.....11011(112 bits)

4.3.3.1 Encoding of chromosomes

The main aim of the feature subset selection stage is to identify small subsets of features which have high classification accuracy. Hence the candidate solution for the feature subset selection problem is a subset of data containing some of the features of the dataset. In genetic algorithms, candidate solutions are represented as chromosomes. In this case, given a set of features we need to find good subsets of the features, hence the candidate solution should represent which features are included in the subset and which features are not included in the subset. We use binary encoding to represent the candidate solution as a chromosome.

The chromosome is represented as a binary string of length equal to the number of features in the dataset. Each bit in the chromosome corresponds to a feature in the dataset. A '0' in the chromosome represents that the corresponding feature is not included in the subset, and a '1' in the chromosome represents that the corresponding feature is included in the subset. Hence the total number of features included in the subset is equal to the number of 1's in the chromosome. The reduced BTH 2000 dataset that is used to perform all the experiments contains 112 features. Hence, every chromosome contains 112 bits and represents one candidate solution to the problem. Table 4.1 shows an example of an encoded chromosome. The chromosome would contain 112 bits, with each bit representing whether the corresponding feature from the dataset in included in the subset or not. Only those features having the corresponding bit in the chromosome set to '1' are included in the subset.

4.3.3.2 Selection mechanisms

As discussed in Chapter 2, the selection procedure is one of the most important factor affecting the performance of genetic algorithms. It is essential that fit individuals be selected for preservation and recombination to ensure that new traits are explored without losing the current best solutions. In our system we have implemented four different selection mechanisms: random, ranked, fitness proportional and tournament as described in Chapter 2. In this section we describe their implementations in our system in brief.

- *Random selection:* A random number between 0 and the size of population (i.e., number of chromosomes in the population) is generated and the chromosome at the index corresponding

to the random number is returned.

- *Ranked selection:* Chromosomes are selected probabilistically from the population depending upon the rank of chromosomes in the population. The chromosomes are ranked in descending order of their fitness values.

- *Fitness proportional selection:* Chromosomes are selected probabilistically from the population depending upon their fitness value. The larger the fitness value of a chromosome, the higher the chance of being selected.

- *Tournament selection:* We use the standard Boltzmann tournament selection with the *tournament size* set to 25% of the population size. (i.e., *tournament size* = 25)

The effect of each selection mechanism on the overall performance of the genetic algorithm is discussed in Chapter 5.

4.3.3.3 Crossover and Mutation Operators

The crossover operator is responsible for recombination of individual chromosomes. The main aim of applying crossover operator is to explore new adaptive traits by combining two fit individual chromosomes. In this genetic algorithm system we implement single-point, two-point and uniform crossover. As discussed in Chapter 2, the crossover operator operates upon two parent chromosomes and produces two offspring. This section describes the crossover operator in brief.

- *Single-point crossover*: The crossover-point is obtained by generating a random number. The crossover mask is then set by inserting all 0's till the crossover-point and inserting all 1's after the crossover point.

- *Two-point crossover:* Two random numbers are generated to act as crossover-points. The crossover mask is then set by putting 0's till the first crossover point, 1's from the first crossover point to the second crossover point and 0's again from second crossover point till the end of the chromosome.

- *Uniform crossover:* The entire crossover is generated randomly by generating a random binary string of length equal to the number of features.

The mutation operator is implemented in our system by generating a random number to choose the bit position in the chromosome and then flipping the chosen bit.

4.4.3.4 The fitness function

Every chromosome in the population represents a candidate solution. The quality of the solution represented by a chromosome is determined by the fitness value of the chromosome. The fitness function evaluates individual chromosomes of the population and assigns a fitness value to each chromosome. In this section we discuss the fitness function used to evaluate the subsets of features.

As discussed in section 4.3.3.1, our system uses binary encoding to encode the feature subset selection problem as genetic algorithm. A '1' in the chromosome indicates that the corresponding feature is included in the subset and a '0' in the chromosome indicates that the corresponding feature is not included in the subset. Thus, every chromosome represents a set of predictors, identifying the class of the output feature. To determine how accurately each of these sets of predictors predict the class of the output feature, we build a decision tree from the subset of features encoded by the chromosome. Once the decision tree is constructed, it is applied to the entire dataset to compute the number of examples which are correctly classified by the learned decision tree. As we are interested in identifying good subsets of features which contain minimal number of features, we penalize the component of the classification accuracy. We compute the fitness value of the chromosome as shown in Equation 4.2.

Classification accuracy of the decision tree is computed as shown in Equation 4.1

$$\text{accuracy} = \frac{\text{number of examples correctly classified}}{\text{number of total examples}} \qquad (4.1)$$

$$\text{adjAcc} = \frac{\text{numCorrectEx} - \text{factor} * \max(0, \text{numFeatures} - \text{minFeatures})}{\text{numTotalEx}}$$

$$\text{Fitness Value} = \text{adjAcc}^2 \qquad (4.2)$$

where,

adjAcc: adjusted accuracy

numCorrectEx: number of examples or data points correctly classified by the learned decision tree

numFeatures: number of features included in the subset i.e., number of 1's in the chromosome

minFeatures: minimum number of features used in any subset. i.e., minimum number of 1's in chromosome in the population

numTotalEx: total number of examples or data points

We can directly use the classification accuracy to evaluate the subsets of features represented by the chromosomes. However we are interested in finding small subsets of features having high classification accuracy. To find smaller subsets, we penalize the classification accuracy depending on the number of features included in the subset. We compute the adjusted accuracy as shown in Equation 4.2, which is based on the minimum description length theory. We identify the chromosome with least number of 1's and set a threshold value for the number of features to be included in the subset. If a chromosome contains more features then it gets penalized by a factor obtained by subtracting the threshold value from the number of features in the subset and multiplying it by some constant as shown in Equation 4.2. The constant factor is determined experimentally and is set to 5. The fitness value is obtained by taking the square of the adjusted accuracy so that the fitness values are bit more spread out in the overall fitness region.

If two chromosomes have the same classification accuracy, but one chromosome contains a smaller number of features as compared to the other, then the fitness value of the chromosome with less number of features is higher than the fitness value of the chromosome with large number of features due to the penalization term used in computing the adjusted accuracy. By using the fitness function as shown in Equation 4.2 we ensure that our system identifies small subsets of features with high classification accuracies.

4.4.3.5 Population generation and recombination

One of the factors affecting the convergence of the genetic algorithms is how the initial population is generated. In this implementation we generate the initial population randomly. Another factor affecting the convergence of the genetic algorithms is how the population is recombined at the end of each iteration. In this implementation we maintain the size of the population over generations. The population carried over to the next generation is same size as the initial population size. The population size is kept constant throughout the entire evolution process. We first perform crossover on the chromosomes depending upon the specified crossover rate. The crossover rate specifies how many chromosomes should be selected and used for recombination. For example, if the crossover rate is 60%, then 60 chromosomes out of 100 are selected with the help of the selection mechanism for recombination. In the crossover operation, two parent chromosomes are combined to produce two offspring. Both the parents and offspring are evaluated by computing fitness values. Only two chromosomes out of the four chromosomes (i.e, two parent chromosomes and two offspring chromosomes) are promoted to the next generation on the basis of their fitness values.

We then perform the mutation operation on the newly formed offspring depending upon the specified mutation rate. The mutation rate specifies the number of chromosomes which undergo mutation. For example if the mutation rate is 5%, then 5 chromosomes out of 100 are selected for mutation. The chromosomes undergo mutation to produce 5 offspring which are included in the population of the next generation.

The remaining slots of the population are filled by promoting the best chromosomes in the current generation to the next generation. This process of promoting the best chromosomes in the current generation to the next generation is called *elitism* and it ensures that the best chromosomes are not lost due to changes in the chromosome by the crossover and the mutation operations. For example, if the population size is 100, the crossover rate is 60% and the mutation rate is 5%, then 60 chromosomes are formed by the crossover operation, 5 of the newly formed children chromosomes are selected for mutation and the remaining 40 slots are filled by promoting the top 40 chromosomes of the current generation to ensure that the best solution is not lost in the recombination process.

Chapter 5
Experiments and Results

This chapter discusses various experiments carried out to evaluate the research work. First, we discuss a performance estimation technique used to cross-validate the results obtained from various experiments. Then we discuss the results obtained from the preprocessing technique to narrow down the number of features. All of the other experiments carried out were performed on the subset of data obtained from the feature elimination process. We discuss the results obtained by constructing our predictive model. We further present the predictors of cardiovascular disease risk factors and mental health status variables as identified by our predictive model. Next we present comparison of results obtained by varying different parameters of genetic algorithms. Finally we compare the results obtained from the genetic algorithm based system with results obtained from statistical methods and with results obtained from decision trees. In this chapter we present detailed results for only one feature: Q5_11 (diagnosed high blood pressure). We present predictors for the rest of the cardiovascular disease risk factors and mental health status features in Appendix A and Appendix B.

5.1 N-fold cross-validation technique

This section describes the N-fold cross-validation technique used to measure the performance of the system throughout the research (generally with N set to 10). We use the C4.5 decision tree learning algorithm as the classifier for the predicting the class of the output feature. The decision tree classifier divides every data point into one of the output classes. The *accuracy* is the proportion of the total number of predictions made by our system that were correct (i.e., the prediction matched the teacher label). The *accuracy* is calculated from a confusion matrix as shown in Table 5.1. The confusion matrix contains the information about the actual and predicted value of the class of the output feature as predicted by the classifier system. Consider an output feature which can take two possible values: positive and negative. The confusion matrix for predicting the class of the output feature is shown in Table 5.1.

Table 5.1: A confusion matrix to compute accuracy of the base classifier.

		Predicted	
		Negative	Positive
Actual	Negative	A	B
	Positive	C	D

where,

A is the number of correct predictions that an instance is negative.

B is the number of incorrect predictions that an instance is positive (false positives)

C is the number of incorrect predictions that an instance is negative (misses)

D is the number of correct predictions that an instance is positive.

Accuracy is calculated by performing the computation shown in Equation (5.1)

$$\text{Accuracy} = \frac{A+D}{A+B+C+D} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.1)$$

N-fold cross-validation is done by dividing the dataset into N equal parts. The N equal parts of the dataset are created by random sampling of the data instances. For predicting the accuracy, (N-1) parts of the data are used as training data and used for learning decision tree rules. The learned decision tree is then tested on the N^{th} part and accuracy is computed. A similar process is carried out for computing accuracy for each set of the data. The total accuracy is obtained by totaling the predictions obtained from each of the N folds of the data. The N-fold cross-validation technique is used because it allows for more accurate estimates and it guarantees testing on every single data instance.

5.2 Feature Elimination Results

As discussed in Chapter 4, we applied a preprocessing technique to eliminate features that might contain irrelevant or redundant information from the BTH 2000 dataset. The feature elimination process categorizes all the features into three classes: *very likely irrelevant, possibly irrelevant* and *possibly relevant.* To ensure that the features were not misclassified, we performed a 10-fold cross-validation test on each of the three classes. Table 5.2 summarizes the results of the 10-fold cross-validation applied to each of the three classes.

Table 5.2 Accuracy obtained from 10-fold cross-validation for different classes of features.

Features included in the dataset	***Accuracy obtained by 10-fold cross-validation***
Entire dataset	93.96
Reduced subset (subset of data used to perform experiments)	93.39

We performed all of our experiments only on the reduced dataset obtained from the *possibly relevant features.* We can see from Table 5.2 that our feature elimination process eliminates significantly large number of features without affecting the desired accuracy. We can observe from the Table 5.2 that the reduced dataset containing 112 features used for conducting the experiments has comparable accuracy (93.39%) compared to the entire dataset (93.96%) although it has significantly fewer features. To observe the effect of eliminating each class of features, we performed an experiment in which features were dropped progressively from the dataset. Initially all the features were assigned a score by the feature elimination process depending upon the change in accuracy observed by eliminating the corresponding feature. Then the features were classified into three different classes as discussed in Section 4.2.2. The features were discarded from the dataset one at a time and the classification accuracy for the remaining set of features was computed using the 10-fold cross-validation technique. Figure 5.1 shows a graph of the accuracy obtained from 10-fold cross-validation against the features dropped.

From Figure 5.1 we can observe that as the features from the *possibly relevant features* class are dropped, there is a steady drop in accuracy. After the features from *possibly relevant features* class are dropped, the features from the *possibly irrelevant features* class are dropped. The feature elimination process classifies very few features in the *possibly irrelevant features* class. As the features from the *possibly irrelevant features* class are dropped first, the accuracy drops relatively steadily. Although some of the features from the *possibly irrelevant* class contain useful information, they do not contain any additional information about the class of the output feature. From the graph in Figure 5.1 we can see that the accuracy starts dropping rapidly when all the features from *possibly relevant* and *possibly irrelevant* class of features are dropped (around the point when approximately 190-200 features are dropped). The accuracy fluctuates when the features from *very likely irrelevant* class are dropped. We observe a random pattern at the end of the process, as the features remaining in the dataset are not predictive of the output

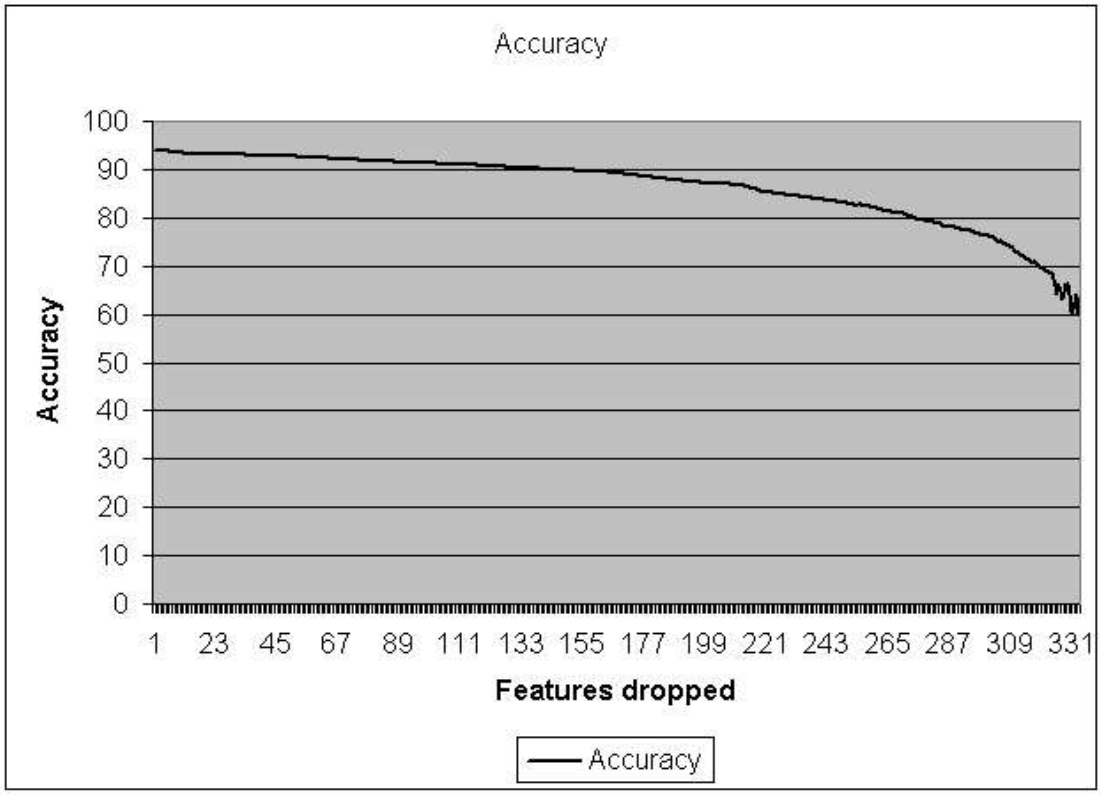

Figure 5.1 A graph of accuracy computed by 10-fold cross-validation as features are dropped from the dataset.

feature and the accuracy fluctuates due to the random data patterns in the remaining subset of the data.

5.3 Predictive model for Q5_11 (Diagnosed high blood pressure)

In this section we describe in detail the predictive model that is constructed by our learning system. We describe the learned data model for the feature Q5_11 (Diagnosed high blood pressure). Similar data models were learned for all the features listed in Table 3.1 and Table 3.2. We discuss the data models for the remaining features in brief in Appendix A and Appendix B..

Our main goal is to identify the predictors of diagnosed high blood pressure, Q5_11 (i.e., we want to identify features in the dataset which contain information about the class of the

output feature Q5_11). In order to construct the data model, we use genetic algorithms to identify good, small subsets of features having high classification accuracy. The system uses decision trees to evaluate the quality of the subsets. Once the subsets of features are identified, decision trees are further used to establish relationships between the subset of features and the output feature. The learned data model thus consists of good, small subsets of features with high classification accuracy and the relationships between the features in the subsets with the output feature. Table 5.3 indicates the parameters of the genetic algorithm used to construct the predictive model. The parameters of the genetic algorithm that best suit the BTH 2000 dataset are set by executing the genetic algorithm with various settings. We describe the experiments to set the parameters in sections 5.4 to 5.9.

After the genetic algorithms are executed with the parameters indicated in Table 5.3, all of the 100 chromosomes in the population represent good, small subsets of features. They are arranged in descending order of their fitness values. Table 5.4 indicates the top 10 feature subsets and the corresponding accuracy. The fitness value is computed by observing the number of correct predictions made by our system using 10-fold cross-validation and penalizing for any additional features that the subset contains.

Table 5.3: Setting of the parameters of genetic algorithms to construct the predictive model.

No. of chromosomes in randomly generated population: 100
Number of iterations: 1000
Crossover Rate: 60%
Mutation Rate: 5%
Method of Parent Selection: Ranked
Crossover Mechanism: Uniform Crossover
Fitness Function: penalizing classification accuracy for missing features as shown in Formula (4.1)
Dataset: Reduced Dataset (112 variables)
Other Constraints: Population carried over to next generation (population size maintained)

Table 5.4: Top 10 feature subsets obtained from the feature subset selection process for variable Q5_11

(1) AGE, Q5_12, PCS12, BMI, Q27, Q5_9, Q5_5, Q24, Q5_4, (92.86)
(2) AGE, Q5_12, PCS12, BMI, Q3, Q27, Q4, Q5_10, Q49REC, Q56, Q5_9, DOLLARS (92.79)
(3) Q5_12, Q5_9, PCS12, BMI, Q5_10, Q36, ALDPANX2, Q5_6, DBDPANX2, Q5_1, RACE, DOLLARS, (92.75)
(4) AGE, Q5_12, PCS12, BMI, Q5_9, Q27, Q5_6, Q5_10, Q65A, EXCERREC, Q7A, DEPRANX2, Q40, (92.68)
(5) AGE, BMI, PCS12, Q5_9, Q3, MCS12, Q5_10, Q39, DBDPANX2, GENHLTH, Q5_9, DOLLARS, (92.70)
(6) Q5_12, Q5_9, BMI, Q34, Q27, Q7A, Q27, Q43, EXCERREC, Q41, RACE, Q54NEW, Q56, (92.62)
(7) PCS12, Q27, Q39, Q66A, Q42, ALDPANX2, Q49, Q24, Q4, Q49REC3, Q25, Q57, Q5_15NEW2, Q5_5, Q58, Q56, (92.68)
(8) AGE, PCS12, Q5_12, BMI, Q34, Q27, EXCERREC, Q7A, Q4, Q5_10, DBDPANX2, RACE, (92.57)
(9) Q5_12, PCS12, BMI, Q5_10, MCS12, Q7B, Q5_16NEW2, Q4, Q5_6, Q6, Q26_11, RACE, (92.56)
(10) Q5_12, Q5_9, PCS12, BMI, Q34, Q40, Q5_6, MARITAL, Q5_10, MCS12, Q5_6, RACE, (92.52)

Once the genetic algorithm has performed the task of feature subset selection, we get the set of predictors identifying the appropriate class of the output feature Q5_11. The solution obtained from the feature subset selection task just lists a subset of features that most accurately predict the class of the output variable Q5_11, but do not convey any information regarding importance of each individual feature in the subset or any kind of relationship between the subset of features and the output feature. To determine the importance of each feature, we use the same approach used in the feature elimination process. The feature is dropped from the subset and the change in accuracy is observed. If a feature occurs in more than one feature subset, then the average change is accuracy is computed for that feature over the subsets in which it occurs. All the features are ranked in descending order of their score, which is computed by observing the change in accuracy after the feature is dropped from the subset. Once the importance of the features is determined, we build a decision tree with all the features that occur in the top 10 chromosomes. The learned decision tree is then converted to a set of *if-then* rules to establish the relationship between input features and Q5_11. Table 5.5 gives the list of predictors of Q5_11 (diagnosed high blood pressure) arranged in descending order of their importance assesses on the basis of change in accuracy after dropping the corresponding feature from the subset of data. It lists a set of features which predict that an individual in the regional population has diagnosed high blood pressure and states the relationships between the individual features

and Q5_11. Table 5.6 lists the top five rules which predict the correct class of the output feature Q5_11. The rules are ranked on the basis of number of correct examples classified by the rule. The predictors obtained from our system were inspected by a medical expert and were tagged with three classes as shown in Table 5.5. The predictors were tagged as: (1) predictors make sense and have support in the medical field for that predictor being meaningful (2) variable makes sense but is interesting (something which the medical field has not seen before and (3) variable does not make sense from medical perspective. The predictive model constructed by our system gives useful information about the data from various perspectives as can be seen from Table 5.4, Table 5.5 and Table 5.6. We construct similar predictive models for all of the mental

Table 5.5 Top predictors identifying that an individual in the regional population has diagnosed high blood pressure (Q5_11 = 1) ranked in order of their importance and the relationship between top predictors and Q5_11

Feature	***Rule***	***Description***	***Tag***
AGE	AGE > 53.5	Age of the individual is greater than 53.5 years	Has support
Q5_12	Q5_12 < 1.5	Has diagnosed high cholesterol	Has support
Q5_9	Q5_9 < 1.5	Has diagnosed heart related problems	Has support
PCS12	PCS12 < 44.618	Physical health score less than 44.618	Has support
BMI	BMI > 26.019	Body mass index greater than 26.019	Has support
Q34	Q34 < 1.5	Limited: moderate activities = limited a lot	Has support
Q27	Q27 >= 1.5	Prescriptions written and filled for medicines	Has support
Q3	Q3 >= 2.5	General health = good, fair or poor	Has support
MCS12	MCS12 < 55.8	Composite mental health score less than 55.8	Interesting
Q39	Q39 < 1.5	Limited in kind of work	Has support
Q5_10	Q5_10 < 1.5	Has diagnosed stroke related problem	Has support
Q40	Q40 < 1.5	Accomplished less due to mental health	Interesting
ALDPANX2	ALDPANX2 < 32.5	Accomplished less w/o depression or anxiety	Interesting
Q24	Q24 > 1.5	Health insurance status (partially or uninsured)	Interesting
DBDPANX2	DBDPANX2 = 31	Downhearted and blue w/o depression or anxiety	Interesting
Q7B	Q7B <=2	Servings of fruits/veg per day (less than 1 serving)	Has support

Table 5.6 Top 5 rules predicting high blood pressure (Q5_11=1) in the regional population

(1) If AGE>43.5 and BMI>29.74 and DBDPANX2=31 and Q5_9=1 then Q5_11=1

(2) If Q5_12=1 and PCS12<46.085 and Q40=1 and Q5_10=1 and MCS12<63.5 then Q5_11=1

(3) If Q5_12=1 and Q5_9=1 and ALDPANX2=31 and DBDPANX2=31 then Q5_11=1

(4) AGE>53.5 and Q5_12=1 and BMI>30.49 and Q4>=4 and Q5_10=1 and EXCERREC=1 then Q5_11=1

(5) If Q5_12=1 and PCS12<35.41 and Q5_10=1 and Q5_16NEW2=2 then Q5_11=1

health features listed in Table 3.1 and all of the cardiovascular disease risk factors listed in Table 3.2. The results for all the other variables are listed in Appendix A. The following sections discuss the effect of the parameters of genetic algorithms on the overall performance of the genetic algorithms.

5.4 Effect of Elitism on performance of genetic algorithms

Figure 5.2 shows a graph of accuracy of the subsets of features obtained by our system against the number of iterations performed by the genetic algorithm process. As discussed in

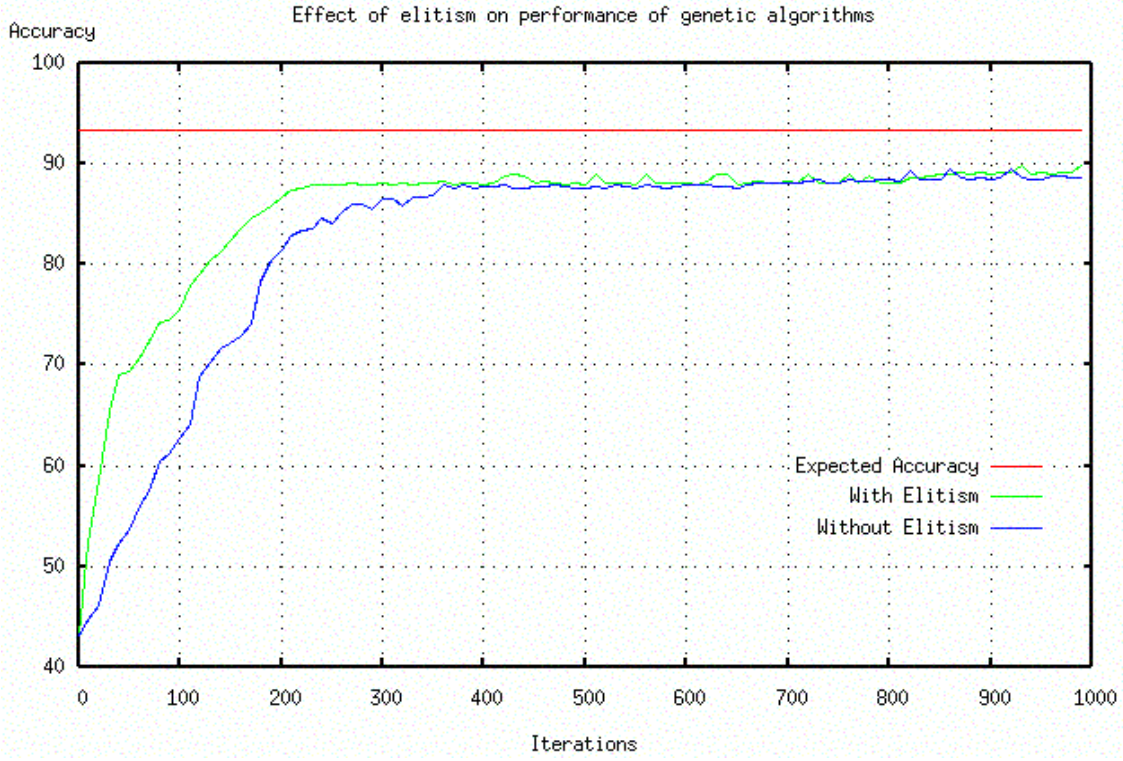

Figure 5.2 Results measuring the effect of *elitism* on performance of genetic algorithm.

Chapter 2, *elitism* is a process in which few of the top chromosomes in the current generation are promoted to the next generation to ensure that the best current solution is not lost. To observe the effect of *elitism* on the overall performance of the genetic algorithms, we constructed a predictive model employing *elitism* and a predictive model without employing *elitism* keeping all the parameters of genetic algorithms constant. We compute average accuracy of all the chromosomes after every iteration. We can see from the graph in Figure 5.2 that though both the implementations have comparable accuracy after 1000 iterations, the genetic algorithm converges faster if we employ *elitism*, which is unsurprising given the fact that *elitism* preserves the best solution by carrying over few of the top chromosomes to the next generation.

5.5 Effect of selection mechanism on performance of genetic algorithms

Figure 5.3 shows a graph of accuracy of subsets of features obtained by our system against the number of iterations performed by the genetic algorithm. We have implemented four selection mechanisms: random, ranked, fitness proportional and tournament selection. We can observe from the graph in Figure 5.3, that ranked selection has faster convergence rate than

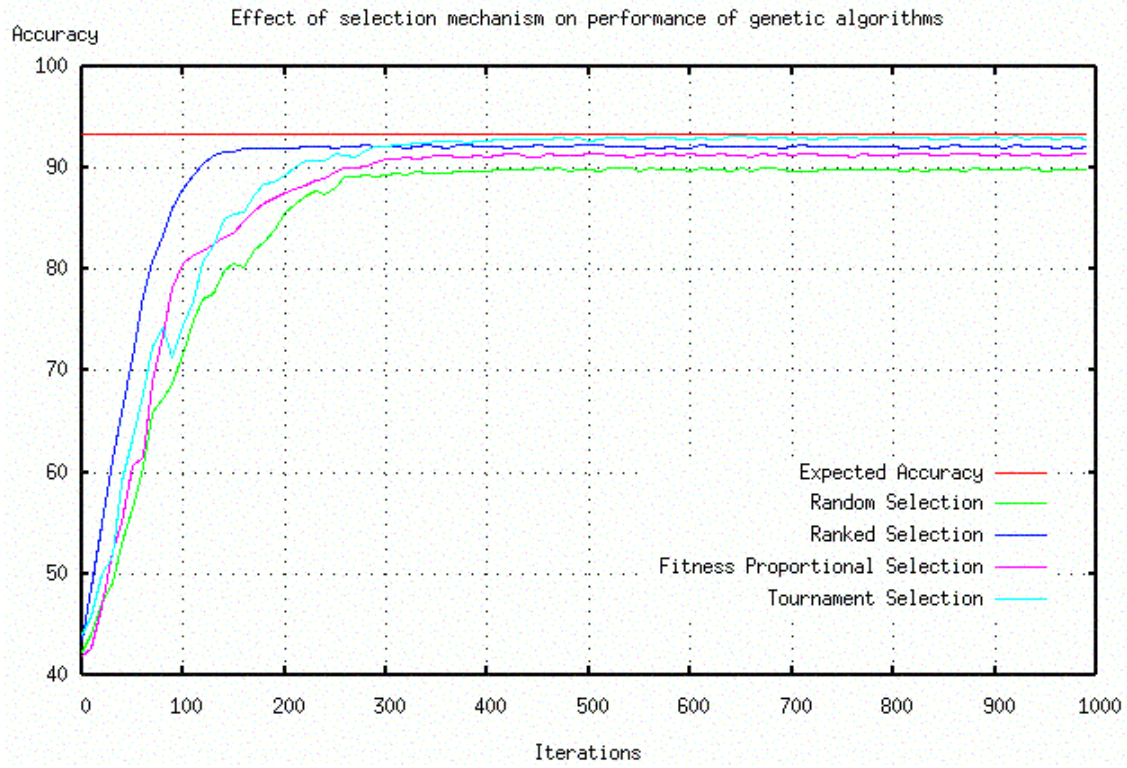

Figure 5.3: Results measuring the effect of selection mechanisms on performance of genetic algorithms.

any other selection mechanisms. This is because of the fact that in ranked selection, higher ranked chromosomes have more probability of getting selected. We can also observe that ranked selection and tournament selection perform better than fitness proportional and random selection. In our system, the fitness proportional selection does not work well, as the fitness values of the chromosomes are very close to each other. As expected the random selection performs the worst and has slower convergence rate. Tournament selection performs better over time than ranked selection. However proper care needs to be taken to set the value of tournament size. The tournament size in the tournament selection was set to 25. We observed experimentally that the tournament selection performed reasonably well if the tournament size was set between 25% to 50% of the number of individuals in the population.

5.6 Effect of crossover mechanism on performance of genetic algorithms

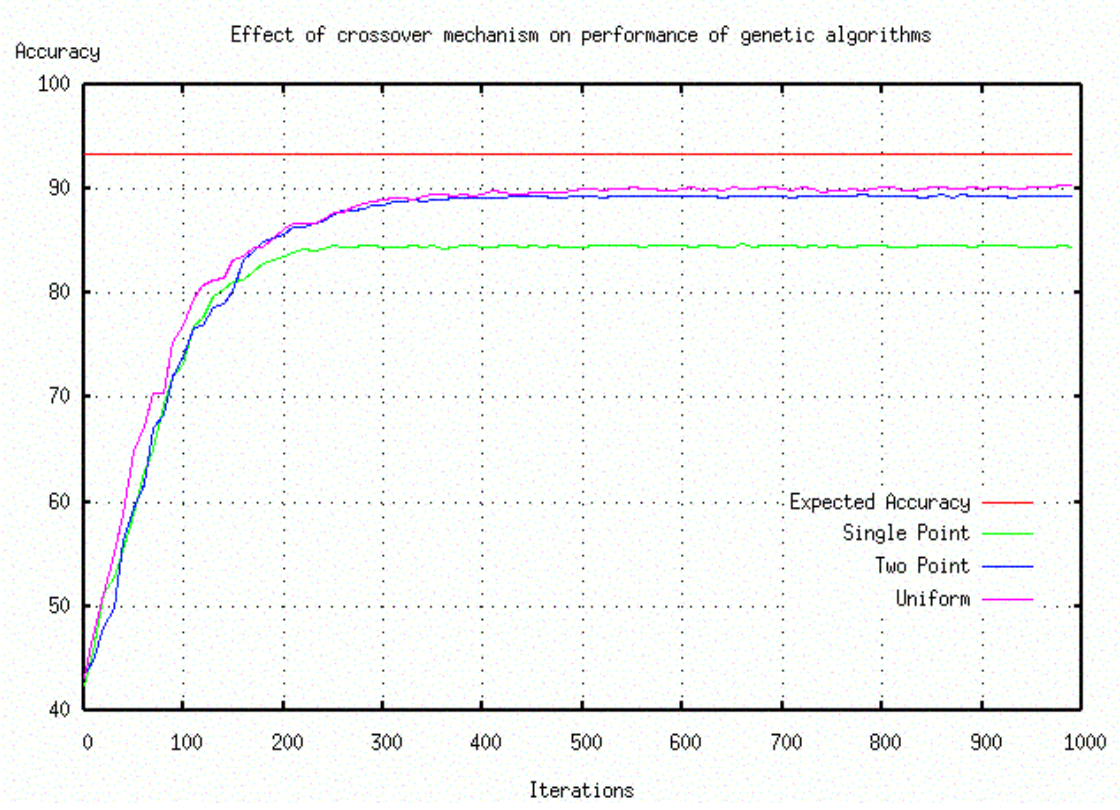

Figure 5.4 Results measuring the effect of crossover mechanisms on performance of genetic algorithms.

Figure 5.4 shows a graph of average accuracy of subsets of features obtained by our system against the number of iterations performed by the genetic algorithm. We have implemented three crossover mechanisms: single-point, to-point and uniform crossover. The graph indicates the average accuracy obtained from all the chromosomes in a generation when each of the three crossover mechanisms are used. All the other parameters are kept constant. We can see that although uniform crossover performs better than the rest of the crossover mechanisms, we obtain comparable results using two-point crossover. Uniform crossover and two-point crossover tend to perform better than the single-point crossover as more traits are explored in two-point and uniform crossover than in single-point crossover.

5.7 Effect of population size on performance of genetic algorithms

Figure 5.5 shows a graph of average accuracy of subsets of features obtained by our system against the number of iterations performed by the genetic algorithm. The graph shows average accuracies of all the chromosomes in the population. All the parameters are kept

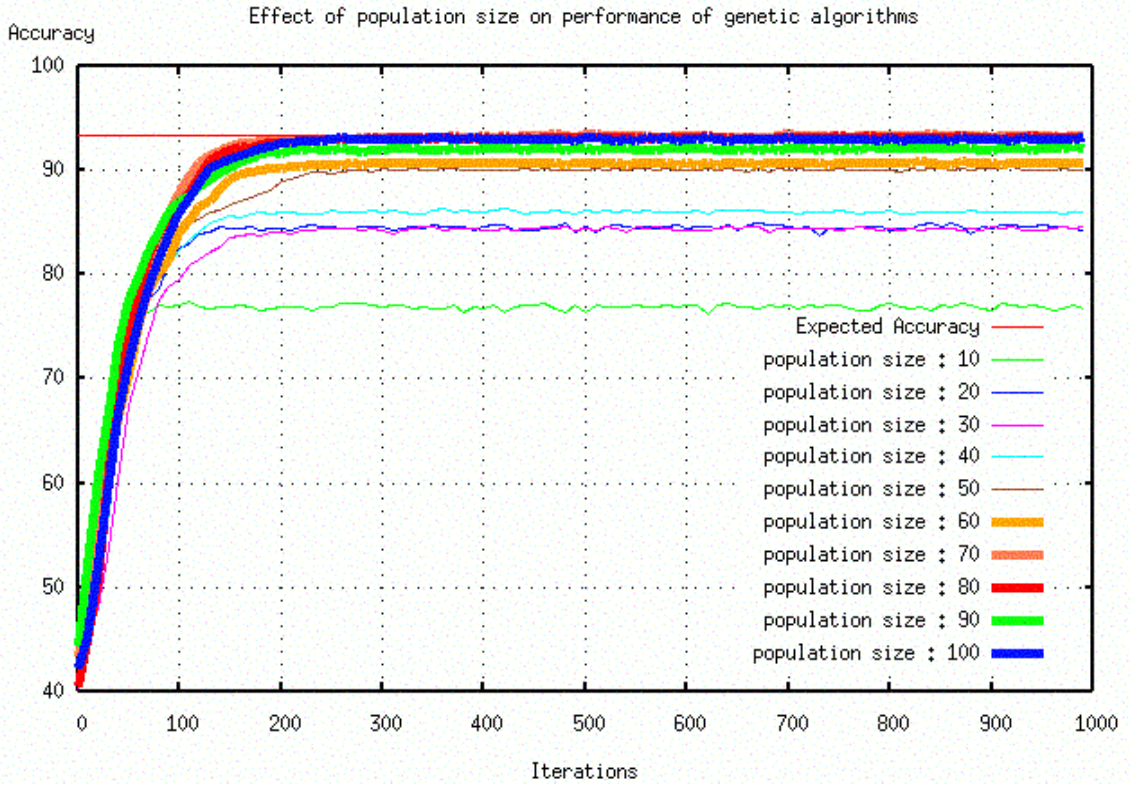

Figure 5.5: Results measuring the effect of population size on performance of genetic algorithms.

constant and only the population size is varied from 10 to 100. We can see that for a small value of population size, the accuracies are significantly lower. This is due to the fact that the lower the population (i.e., less the number of chromosomes), the fewer the traits observed in the population. Hence with a small population size not all the adaptive traits are explored (i.e., all the possible combinations of good features are not explored) and as a result the average accuracy remains significantly lower than the desired accuracy. From the graph we observe that we get good accuracies with a population size of 60 to 70 chromosomes. Although large population size enables more traits to be explored, the performance is slightly lower because the graph shows the average accuracy of the population.

5.8 Effect of crossover rates on performance of genetic algorithms

Figure 5.6 shows a graph of average accuracy of subsets of features obtained by our system against the number of iterations when crossover rate is varied. Crossover rate is varied from 0% to 100% while all the other parameters of genetic algorithm are kept constant. From

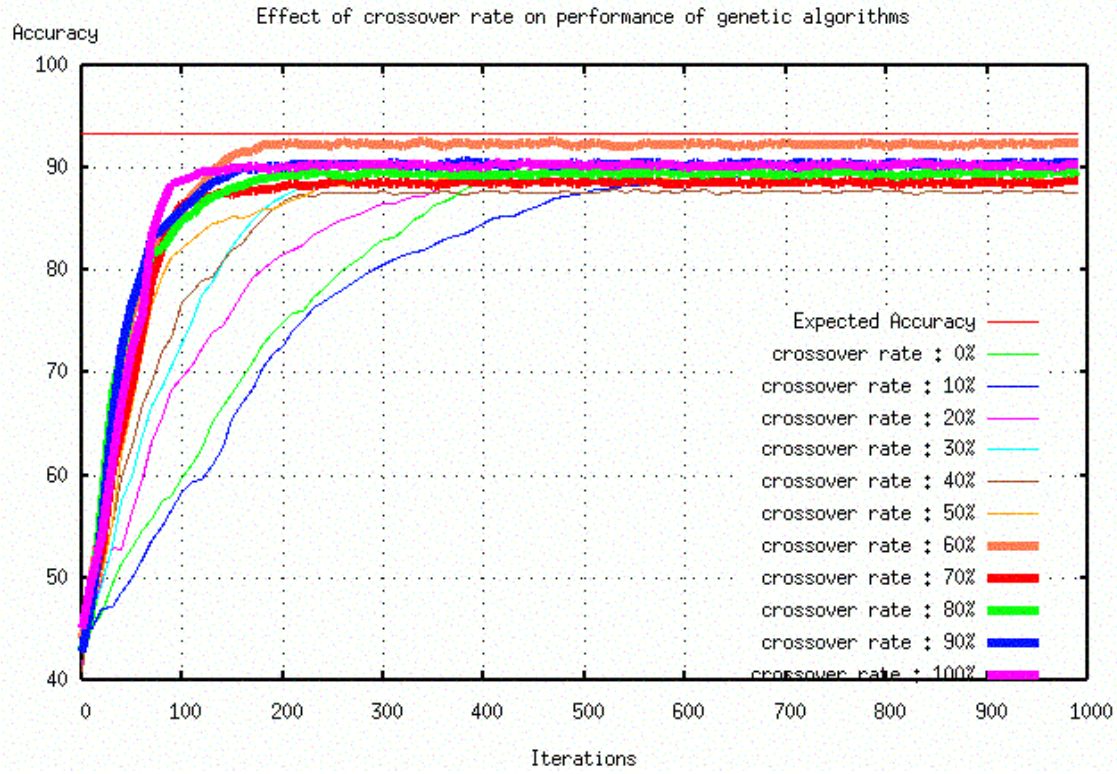

Figure 5.6: Results measuring the effect of crossover rate on performance of genetic algorithms.

graph we can observe that high accuracy is obtained if crossover rate is set to 60%. For lower crossover rates, lower accuracy is observed as fewer chromosomes are used for recombination and as a result many adaptive traits are left unexplored. As the crossover rates approach 100%, the performance of the system degrades as all the chromosomes are used up for recombination and the chromosomes with high fitness values may be lost.

5.9 Effect of number of iterations on performance of genetic algorithms

Figure 5.7 shows a graph of accuracy of subsets of features obtained by our system against the number of iterations when the number of iterations is varied. The graph shows the average accuracy of all the chromosomes in a generation and the accuracy of the best subset of features. We can see from the graph that the average accuracy obtained from our system increases rapidly initially, highlighting the adaptive nature of genetic algorithms. The genetic algorithms are based on artificial evolution that follows the principle of natural evolution, and they make the population (i.e., set of chromosomes which are candidate solutions to the problem) more adaptive over time.

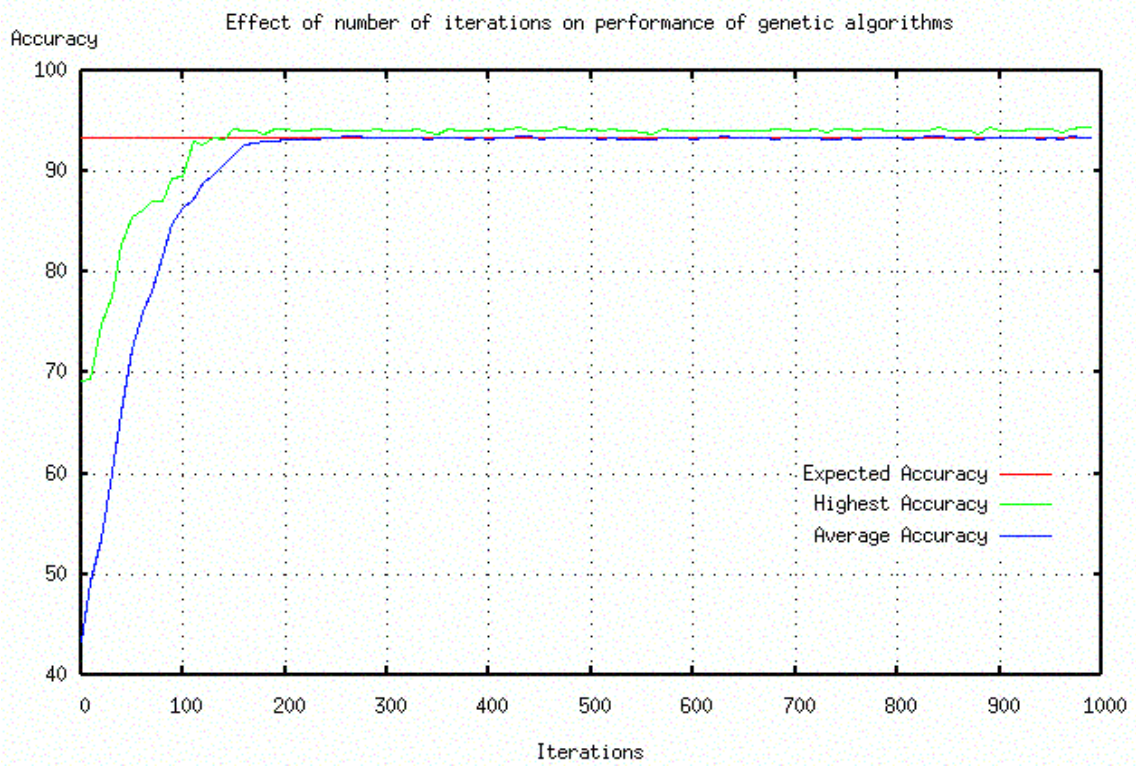

Figure 5.7 Results measuring the effect of number of iterations on performance of genetic algorithms.

Table 5.7: A comparison between results obtained by statistical methods and results obtained by our system.

Top 10 Correlations	*Relationships in our system*
EXCERREC, ALDPANX2 = 2.0	If ALDPANX2=31 then EXCERREC=1
Q5_12, DEPRANX2 = 2.0	If Q5_12<1.5 then DEPRANX2=2
Q5_11, DEPRANX2 = 1.8	If Q5_11<1.5 then DEPRANX2=2
Q5_12, Q15NEW2 = 1.8	No relationship determined.
Q5_12, Q16NEW2 = 1.7	If Q5_12<=1.5 then Q16NEW2=2
EXCERREC, Q15NEW2 = 1.7	If Q15NEW2=1 then EXCERREC=1
EXCERREC, DEPRANX2 = 1.7	If DEPRANX2=2 then EXCERREC=1
EXCERREC, DBDPANX2 = 1.7	No relationship determined.
BMI, DEPRANX2 = 1.6	If BMI>28.35 then DEPRANX2=2
BMI, Q15NEW2 = 1.6	If BMI>=29.23 then then Q15NEW2=1

5.10 Comparison between our system and statistical methods.

The results obtained from our system are summarized in Sections 5.2 to 5.9 and in Appendix A and Appendix B. Dr. Tim Van Wave [Van Wave, 2004] used statistical methods to establish relationships between cardiovascular disease risk factors and mental health status variables in the BTH 2000 dataset. He used odds ratio and chi square to find correlation between cardiovascular disease risk factors and mental health status variables. Table 5.7 summarizes the comparison between the results obtained by statistical methods and the results obtained by our system. The results obtained using odds ratios are arranged in descending order of their values. We consider only the top ten correlations obtained by Odds ratio. Correlation between cardiovascular disease risk factors and mental health status features was found using odds ratio. The table indicates the top ten correlated features with the value obtained from odds ratio using contingency tables. The result space of the two features was observed to find a rule suggesting relationship between the two features. From Table 5.7 we can observe that our system identifies relationship between the two features for 8 out of the 10 top correlations obtained by the traditional statistical methods. Table 5.7 lists the relationships between the two correlated features. We can see that in addition to identifying the predictors of cardiovascular disease risk factors and mental health status variables our system also establishes relationships between them. Hence from the comparison we can conclude that our system identifies predictors of

cardiovascular disease risk factors and mental health status variables accurately and also suggests the nature of the relationships between them. The predictive models constructed by our system may provide useful information to health professionals in addressing cardiovascular disease risk factors and mental health issues. We also believe that our predictive model may be used to derive predictive models from other survey datasets using the specified parameters.

Chapter 6
Related Work

Machine learning and knowledge discovery in databases are closely related fields. Many machine learning techniques are employed to discover patterns in data for decision making. Numerous methods have been proposed to extract useful information from the datasets. In this work, we employed the feature subset selection process using genetic algorithms to filter features containing useful information from a large set of features. In the first section we summarize the findings of research related to the feature subset selection task and genetic algorithms. A large amount of research work has been carried out to explore the patterns in epidemiological datasets using machine learning techniques. In the second section, we summarize the work and findings of selected related research in exploiting the patterns in epidemiological datasets.

6.1 Genetic algorithms and feature subset selection.

In this section we discuss some of the research work carried out in the field of genetic algorithms. We summarize the work and findings of some research work concentrated on applying genetic algorithms to the feature subset selection task and improving performance of the genetic algorithms.

Oliveira et al. [2001] applied genetic algorithms for handwritten digit recognition. They applied genetic algorithms for the feature subset selection task for handwritten digital recognition through a modified wrapper based multi-criterion approach in conjunction with multilayer perceptron neural network. They implemented two approaches: a simple genetic algorithm and an iterative genetic algorithm for practical pattern recognition. They used a binary encoding of chromosome and implemented fitness proportional selection mechanism. They also attempted to find the optimal parameter setting for the digit recognition problem. They found out experimentally that their system performed most accurately for the following set of parameters: population size – 30, number of generations – 100, crossover probability – 0.8, mutation probability – 0.007. They also found out that although the performance of simple genetic algorithms and iterative genetic algorithms were comparable and both approaches reduced the

complexity of the classifier, the iterative genetic algorithm converged a lot faster.

Matsui et al. [1999] used genetic algorithms in conjunction with neural networks to select an optimal combination of features to classify gray matter and white matter in MRI segmentation. They applied genetic algorithms to carry out the feature subset selection task, and used neural networks to test the predictive model. They used a new approach called the vector quantized conditional class entropy to evaluate the combination of features rapidly without testing the actual classifier. They used the following set of parameters to conduct their experiments: population size – 30 , number of iterations – 30, selection mechanism – fitness proportional selection.

Both of these research works - Oliveira et al. [2001] and Matsui et al. [1999] - use a combination of genetic algorithms and neural networks to construct the predictive models. Our system uses a combination of genetic algorithms and decision trees to identify good predictors of the output variable and to establish relationships between the predictors and the output variable. We also use a fitness function that penalizes the classification accuracy based on minimum description length theory.

Ever since the evolution of genetic algorithms in late 1970's, many researchers have concentrated their work on identifying optimal parameters of genetic algorithms. Many sets of parameters have been identified, however none of the parameter sets serve as a benchmark. Optimal parameters vary for different problem domains. However, DeJong's settings [DeJong and Spears, 1990] and Grefenstette's settings [Grefenstette, 1986] are regarded as standard settings for most of the genetic algorithm problems. Both of these research efforts were mainly concentrated on identifying an optimal set of parameters for genetic algorithms applied to various problem domains. In our research work we mainly concentrate on applying genetic algorithms to a survey dataset and find a set of parameters that could be used to apply the genetic algorithm learning process to any epidemiological dataset. We use the standard settings to estimate the performance of our predictive model and compare our settings with the standard settings.

6.2 Exploring patterns in epidemiological datasets.

Demsar et al. [2001] attempted to construct outcome prediction models from retrospective data of severe trauma patients after the first surgery. They used decision trees and naive Bayesian classifiers to induce the predictive models making it easier for trauma surgeons to decide the eligibility of patients for damage control. Damage control requires a massive investment of medical resources which are limited and expensive. Their constructed prognostic model helps optimize the use of limited medical resources. They used a preprocessing technique based on RELIEFF [Kira and Rendell, 1992, Kononenko, 1994] to narrow down the list of features used, by eliminating irrelevant features, and used different statistical measures to estimate the performance of the derived prognostic model. However the model was built from a small dataset (data consisted of 68 data points and 174 features collected from Ben Traub General Hospital, Houston, TX) and is regarded as a pilot model to guide further studies and researches. We build our predictive model from a large dataset consisting of 334 features and 6,251 data points. We also used genetic algorithms to identify good subsets of features from a relatively large set of candidate features to construct the predictive model as opposed to feature subset selection by RELIEFF done by Demsar et al. In addition our predictive model is constructed from a hybrid system of genetic algorithms and decision trees whereas Demsar et al. initially perform the feature subset selection and then construct the predictive model.

Inza et al. [2001] applied genetic algorithms and an estimation of distribution algorithm to predict the survival of cirrhotic patients treated with transjugular intrahepatic portosystemic shunt. They constructed a predictive model using feature subset selection and standard machine learning classifiers. They found subsets with the best predictive accuracies by applying genetic algorithm and estimation of distribution algorithm. They used four classifiers (naive Bayesian, decision trees, rule learning and nearest neighbor) in addition to the feature subset selection to construct the predictive model. They got promising results from both genetic algorithm as well as estimation of distribution algorithm. However they used a small dataset to construct the predictive model. They used a dataset containing 107 cases with each case having 77 features. The constructed predictive model helped in building compact models which could be easily understood and applied by the medical staff. Inza et al. used a hybrid genetic algorithm based on estimation of distribution of the data whereas we use a hybrid genetic algorithm combining genetic algorithms with decision trees. In our research work, we use a relatively larger dataset

making it important to use a preprocessing technique to filter out unwanted features. However, the main variation is in the fitness function used to evaluate the subsets of features. Inza et al. use the classification accuracy to evaluate the subsets of features. In our research work we use a fitness function that uses a component of classification accuracy and penalizes it with a component based on number of missing features. We concentrate on identifying smaller yet good subsets of features as compared to Inza et al.

Gamberger et al., [2003] examined patient groups at high risk for coronary heart diseases. Using a data model consisting of data gathering, data cleaning, data transformation, subgroup discovery and statistical characterization tasks. They constructed the data model by combining machine learning techniques and statistical measures. They used a combination of decision trees and statistical measures including sensitivity (true positive rate) and false alarm (false positive rate). They performed subset selection using a heuristic expert guided subgroup discovery algorithm, achieving promising results. Their dataset was collected from institute for Cardiovascular Prevention and Rehabilitation in Zagreb, Croatia. The dataset consisted of 238 records with each record having 22 features. Gamberger et al. used a heuristic expert guided subgroup discovery algorithm, whereas our work concentrates on identifying good subsets of features having high predictive accuracies using genetic algorithms.

Dr. Timothy Van Wave [Van Wave, 2004] performed a secondary analysis of the BTH 2000 dataset [Block et al., 2000] using Pearson's chi-square ratio to estimate independence between mental health status and cardiovascular disease risk factors. In his work, he used two-way contingency table analysis to evaluate statistical relationship between mental health status and cardiovascular disease risk factors. His work suggests that self-assessed and physician diagnosed mental health status is significantly associated with cardiovascular disease risk factors. However, his work only identifies association between cardiovascular disease risk factors and mental health status variables and does not establish any relationships between the two categories of variables, as he used traditional statistical methods. In our work we use machine learning techniques to address the same problem and establish the relationships between cardiovascular disease risk factors and mental health status variables. He further suggests that early intervention addressing poor mental health and recognized risk factors for heart diseases may work together to reduce heart disease risks more effectively. We use the results obtained from his work as a benchmark to compare the results obtained from our predictive model.

Kankaria [2004] has implemented a Bayesian network model to determine predictors of cardiovascular disease risk factors and predictors of mental health status of a regional population. In her thesis, she uses a Bayesian network structure learning to develop a web-based tool box to view the Bayesian structure of data and to construct a predictive model from the BTH 2000 dataset [Block et al., 2000]. The tool box is based on a Construct-TAN method put forth by Friedman et al. [Friedman et al., 1997] and tries to identify relationship between various features of the data. This research work provides yet another perspective of examining the BTH 2000 dataset in addition to statistical methods [Van Wave, 2004] and our hybrid predictive model constructed using genetic algorithm and decision trees.

Chapter 7
Future Work

In our research, we built a system which constructed predictive models using genetic algorithms and decision trees. The system identifies small subsets of features having high classification accuracy. We further addressed the issue of identifying predictors of cardio-vascular disease risk factors and mental health of an individual in a regional population by constructing a predictive data model. In this chapter we discuss possible improvements that can be done to enhance our work in future. We discuss some factors that would make our predictive data model more accurate and efficient.

In this work we are interested in inferring relationships between variables in survey datasets. The data in the survey datasets is collected by conducting surveys in a representative population. In most of the surveys there is a high probability that the respondents did not answer all the questions. Such unanswered responses are dealt by our system by treating them as missing values in data which is then handled by the C4.5 decision tree learning algorithm. Hence the missing values are handled from the machine learning point of view. However to build even more accurate models we need to tackle the problem of handling missing responses and need to build a predictive model which would also learn from the missing responses. The respondents also had a choice to refuse to answer a question or choose the option of *don't know / not sure* if the respondent was not sure about the answer to any of the question. In future we would like to incorporate some technique in our system that would construct a predictive model for the missing responses and perhaps help in designing future surveys.

In this work we focus on finding patterns in epidemiological datasets. There are many organizations and collaboratives undertaking surveys in United States for collecting population based information. Many population-based surveys have been carried out and the data collected. Some of the other epidemiological datasets are: NHIS (National Health interview Survey), BRFSS (Behavioral Risk factor Surveillance Systems), NHANES (National Health and Nutrition Examination Survey). NLAES (National Longitudinal Alcohol Epidemiological Survey) and YRBS (Youth Risk Behavior Surveillance System). These datasets are collected by government

agencies and many of them are freely available. We would like to test our proposed system on various epidemiological datasets and test the performance of the system applied to various survey datasets. We would concentrate of applying our system to epidemiological datasets, however our system is generalized and can be used to find interesting data patterns in all types of datasets by making slight variations in the system.

We implement a system which is a combination of genetic algorithms and decision trees. Genetic algorithms are an iterative process and take time to converge. Hence we need to run the genetic algorithm for several iterations before any useful results are obtained. Our system grows decision trees for every chromosome in the population in every iteration. For datasets with large number of features and/or data points, our system requires significant runtime. Genetic algorithms are naturally suited for parallel implementation. Tanese [1989] and Cohoon et. al. [1987] have explored approaches for parallelization of genetic algorithms. In future we would like to explore the possibilities of parallelizing our system to reduce the runtime and increase the efficiency.

We would also like to incorporate our learning system in web-based tool, which can be made freely available to explore patterns in various datasets. We would also like to explore various fitness functions to obtain good, small subsets of data. We believe that our predictive model constructed from the BTH 2000 dataset might help physicians and health professionals by identifying predictors of cardio-vascular disease risk factors and mental health status variables. In general, the system can be applied to any epidemiological dataset to determine important features of the dataset and relationships within the dataset.

Chapter 8

Conclusions

In this research work we attempted to build a system using genetic algorithms and decision trees to construct a predictive model which identifies good, small subsets of features with high classification accuracy and establishes relationships within the dataset. Our system incorporates a preprocessing technique which categorizes features of dataset based on the variations observed in the classification accuracy when the feature was dropped from the dataset. In this work, we also observe the effect of various parameters on the performance of the genetic algorithms. We also determine a set of parameters of the genetic algorithms for which predictive models with high accuracy are obtained.

In this thesis work, we tested our system on an epidemiological dataset. The predictive model that we have created using the BTH 2000 dataset effectively addresses the problem of identifying predictors of cardiovascular disease risk factors and mental health status variables and discovering interesting relationships within the data, especially between cardiovascular disease risk factors and mental health status variables. All the results obtained from the system and their reasonableness from a medical perspective were inspected and confirmed by the expert.

From the results obtained from our system we can conclude that small subsets of features are sufficient to build meaningful predictive models from the dataset. We can also conclude that the feature elimination preprocessing technique implemented in our system is an effective technique to discard likely irrelevant features. The reduced number of features prevents overfitting of data while building decision trees and makes the system more effective, efficient and accurate. Our system produced results which were useful in machine learning domain as well as medical domain. Our system will be useful in exploring interesting patterns in data, especially in epidemiological datasets, and would be useful to construct meaningful predictive models.

We believe that our predictive model built from the BTH 2000 dataset may provide useful information to enable physicians and health professionals to intervene early in addressing mental health issues and cardiovascular disease risks. The results obtained from our predictive

model would be helpful for planning and evaluating health-related programs and services.

Bibliography:

[AHA, 2004] Available at http://www.americanheart.org. *Heart Disease and Stroke Statistics – 2004 Update.* Compiled by American Heart Association and American Stroke Association.

[Bäck and Hoffmeister, 1991] Bäck, T. and Hoffmeister, F.: *Extended Selection Mechanisms in Genetic Algorithms.* Proceedings of the Fourth International Conference on Genetic Algorithms, San Mateo, California, USA: Morgan Kaufmann Publishers, 1991.

[Block et al., 2000] Block, D. E., Kinney, A. M., Sundberg, L., Peterson, J. M., Kelly, G. L. and Bridge To Health Colloborative (2000). *Bridge to Health Survey 2000: Northeastern Minnesota and Northwestern Wisconsin regional health status survey.* (Available from Community Health Department, St. Mary's/Duluth Clinic Health System, 407 East Third Street, Duluth, MN 55805)

[Blickle and Thiele, 1995] Blickle, T. and Thiele, L.: *A Comparison of Selection Schemes used in Genetic Algorithms* (2. Edition). TIK Report No. 11, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zürich, Switzerland, 1995.

[Breiman et al., 1984] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, P. J. *Classification and regression trees.* Belmont CA: Wadsworth International Group, 1984.

[Chen et. al., 1996] Chen, M. S., Han, J. and Yu, P. S. *"Data Mining: An overview from a database perspective,"* IEEE Trans. Knowl. Data Eng., vol. 8, pp. 866--883, Dec. 1996.

[Cohoon et. al., 1987] Cohoon, J. P., Hegde, S. U., Martin, W. N., and Richards, D. *Punctuated Equilibria: A Parallel Genetic Algorithm.* Proceedings of the second International Conference on Genetic Algorithms (pp. 148-154), 1987

[Darwin, 1859] Darwin, C.: *On the origin of species by means of natural selection.* London: Murray, 1859.

[Davidson et al., 2001] Davidson, S., Judd, F., Jolley. D., Hocking, B., Thompson, S. and Hyland, B. *Cardiovascular risk factors for people with mental illness.* Aust N Z J Psychiatry. Apr 2001;35(2):196-202.

[DeJong and Spears, 1990] DeJong, K.A. and Spears, W.M. *"An Analysis of the Interacting Roles of Population Size and Crossover in Genetic Algorithms,"* Proc. First Workshop Parallel Problem Solving from Nature, Springer-Verlag, Berlin,1990.pp. 38-47

[Demsar et al., 2001] Demsar, J., Zupan, B., Aoki, N., Wall, M. J., Granchi, T.H., and Beck J. R., *Feature Mining and predictive model construction from severe trauma patient's data* doi:10.1016/S1386-5056(01)00170 Elsevier Science Ireland Ltd., 2001

[Ford et al., 1998] Ford, D.E., Mead, L.A., Chang, P.P., Cooper-Patrick, L., Wang, N.Y. and Klag, M.J. *Depression is a risk factor for coronary artery disease in men: the precursors study.* Arch Intern Med. Jul 13 1998;158(13):1422-1426.

[Freund and Schapire, 1996] Freund, Y. and Schapire, R. *Experiments with a new boosting algorithm.* In Proceedings of the Thirteenth International Conference on Machine Learning, pages 148-156, Bari, Italy, 1996.

[Friedman et al., 1997] Friedman, N., Geiger, D. and Goldszmidt, M. : *Bayesian Network Classifier*. Volume 29, Issue 2-3, Nov-Dec 1997 Special issue on learning with probabilistic representations page: 131-163, Kluwer Academic Publishers , 1997

[Galindo and Tamayo, 1997] Galindo, J., and Tamayo, P., *Credit Risk Assessment using Statistical and Machine Learning Methods as an Ingredient for Risk Modeling of Financial Intermediaries* Handle: RePEc:sce:scecf7:31 Series: Computing in Economics and Finance, 1997

[Gamberger et al., 2003] Gamberger, D., Lavrac, N., and Krtacic, G., *Active subgroup mining : A case study in coronary heart disease risk group detection.* Artificial Intelligence in medicine Volume 28 Issue 1, 27-57, May, 2003

[Geisler and Manikas, 2002] Geisler, T. and Manikas, T. W. *Autonomous Robot Navigation System Using a Novel Value Encoded Genetic Algorithm.* In proceedings of 45th IEEE Int. Midwest Symposium on Circuits and Systems, 2002

[Goldberg, 1989] Goldberg, D. E.: *Genetic Algorithms in Search, Optimization, and Machine Learning.* New York, Addison-Wesley Publishing Company, 1989

[Goldberg and Deb, 1991] Goldberg, D. E. and Deb, K.: *A Comparative Analysis of Selection Schemes Used in Genetic Algorithms.* In Foundations of Genetic Algorithms. San Mateo, California, USA: Morgan Kaufmann Publishers, 1991.

[Grefenstette, 1986] Grefenstette, J.J. "Optimization of Control Parameters for Genetic Algorithms," IEEE Trans. Systems, Man, and Cybernetics, Vol. SMC-16, No. 1, Jan./Feb. 1986, pp. 122-128.

[Harp et al., 1990] Harp, S. A., Samad, T. and Guha, A., *Designing application specific neural works using the genetic algorithm* in Advances in Neural Information Processing Systems 2 (Youretzky, ed.), (SanMateo, CA), pp. 447-454, Morgan Kaufmann Publishers, 1990

[Hauser and Purdy, 2003] Hauser, J. W. and Purdy, C. N., *Approximating Nonlinear Functions with Genetic Algorithms,* Embedded Systems Programming 16 (1), February 2003

[Holland, 1975] Holland, J. H., *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* MIT Press, 1975.

[Holland et al., 1986] Holland, J. H., Holyoak, K. J., Nisbett, R. E., Thagard, P. R. *Induction: process of inference, learning and discovery.* Computational models of cognition and perception. MIT Press, Cambridge, 1986.

[Inza et al., 2001] Inza, I., Merina, M., Larranaga, P., Quiroga, J., Sierra, B., Girala, M., *Feature Subset Selection by genetic algorithms and estimation of distribution algorithms : A case study in survival of cirrhotic patients treated with TIPS.* Artificial Intelligence in Medicine 23/2 187-205, 2001.

[Kankaria, 2004] Kankaria, R. V. *A tool for constructing and Visualizing tree augmented Bayesian networks for survey data.* Master's Thesis, University of Minnesota Duluth, 2004.

[Kira and Rendell, 1992] Kira, K. and Rendell, L.A. *The feature selection problem: traditional methods and a new algorithm,* in proceedings of AAAI 92, San Jose, CA, 1992

[Kokol et al., 1994] Kokol, P., Mernik, M., Završnik, J., Kancler, K. and Malèiæ, I. *Decision trees based on automatic learning and their use in cardiology.* Journal of medical systems. - ISSN 0148-5598, 18 (1994), 4 ; str. 201-206

[Kononenko, I. *Estimating features: analysis and extensions of relief.* In : Bergadano, F. and De Raedt, L. Editors, Proceedings of the European Conference on Machine Learning (ECML-94), Springer, Berlin (1994). pp 171-182

[Koza, 1992] Koza, J. *Genetic Programming: On the programming of computers by means of natural selection.* Cambridge, MA: MIT Press.

[Mannila, 1996] Mannila, H. *'Data mining: machine learning, statistics, and databases'*, Eight International Conference on Scientific and Statistical Database Management, Stockholm. http://citeseer.ist.psu.edu/article/mannila96data.html, 1996.

[Mathias and Whitley, 1992] Mathias, K. and Whitley, D. *Genetic operators, the fitness landscape and the traveling salesman problem* in parallel problem solving from nature - proceedings of 2nd workshop PPSN 2 (Manner, R., Manderick, R., eds.) pp 219-228 Elseview Publishers, 1992.

[Matsui et al., 1999] Matsui, K., Suganami, Y., and Kosugi, Y., *Feature Selection by Genetic Algorithm for MRI Segmentation.* Systems and Computers in Japan. Volume 30, Issue 7, p 69-78, 1999

[Mesman, 1995] Mesman, B., *Genetic Algorithms for Scheduling Purposes,* Master Thesis, Eindhoven University of Technology, 1995.

[Michalewicz, 1992] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolutionary Programs*, Springer-Verlag, 1992.

[Mitchell, 1997[Mitchell, T. M., *Machine Learning.* WCB/McGraw Hill, 1997

[NIMH, 2004] *Mental Health Statistics.* Statistics provided by National institute of Mental Health. Available at http://www.nimh.nih.gov/publicat/numbers.cfm.

[O'Connor et al., 2000] O'Connor, C. M., Gurbel, P.A. and Serebruany, V.L. Depression and is chemic heart disease. Am Heart J. Oct 2000;140(4 Suppl):63-69.

[Oliveira et al., 2001] Oliveira, L. S., Benahmed, N., Sabourin, R., Bortolozzi, F., and Suen, C. Y., *Feature Subset Selection using genetic algorithms for handwritten digit recognition.* SIBGRAPI 2001:362.

[Pomerleau, 1995] Pomerleau, D. *"Neural Network Vision for Robot Driving,"* The Handbook of Brain Theory and Neural Networks, M. Arbib, ed., 1995

[Pratt et al., 1996] Pratt, L.A., Ford, D.E., Crum, R.M., Armenian, H.K., Gallo, J.J. and Eaten, W.W. *Depression, psychotropic medication, and risk of myocardial infarction - Prospective data from the Baltimore ECA follow-up.* Circulation. DEC 15 1996;94(12):3123-3129.

[Quinlan, 1986] Quinlan, J. R. *Induction of decision trees.* Machine Learning, 1(1), 81-106, 1986.

[Quinlan, 1987] Quinlan, J. R., *Rule induction with statistical data – a comparison with multiple regression.* Journal of the Operational Research Society, 38, 347-352, 1987

[Quinlan, 1993] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

[Quinlan,1996] Quinlan. J. R., *Bagging, boosting, and C4.5.* In Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96), pp. 725-730. AAAI/MIT Press, 1996.

Runyon, 1977] Runyon, K. E. *Consumer Behavior and the Practice of Marketing*. Columbus, Ohio: Charles E. Merrill Publishing Company., 1977

[Salzberg et. al, 1995] Salzberg. S., Chandar, R., Ford, H., Murthy, S. K., and White, R. *Decision trees for automated identification of cosmic-rays hits in Hubble Space Technology images.* Publication of Astronomic Society Pacific 107:279-288, 1995

[Schnecke and Vornberger, 1996] Schnecke, V. and Vornberger, O., *A Genetic Algorithm for VLSI Physical Design Automation Process/* Second International Conference on Adaptive Computing in Engineering and Control, ACEDC'96, 26-28 March 1996, Plymouth, U.K., 53-58

[Tanese, 1989] Tanese, R. *Distributed Genetic Algorithm. Proceedings* of the 3rd International Conference on Genetic Algorithms (pp. 434-439) , 1989

[VanWave, 2004] VanWave, T., *Secondary Analysis of Bridge to Health Survey 2000 Dataset using statistical methods*. Unpublished work. Department of Family Medicine, University of Minnesota Duluth, 2004.

[Ware et al., 1996] Ware J. E., Jr., Kosinski and M., Keller S.D., *A 12 Item Short Form Health Survey: Construction of scales and preliminary tests of reliability and validity.* Med Care 1996; 34:220 233.

[Whitley, 1989] Whitley, D.: *The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best.* Proceedings of the Third International Conference on Genetic Algorithms, San Mateo, California, USA: Morgan Kaufmann Publishers, 1989.

Appendix A: Predictors of Cardiovascular Disease Risk Factors

Table A1: Top predictors identifying that an individual in the regional population has diagnosed high blood cholesterol (Q5_12 = 1) and the relationship between top predictors and Q5_12

Feature	***Rule***	***Description***	***Tag***
Q5_9	Q5_9=1	Has diagnosed heart trouble	Has support
Q49REC2	Q49REC2<1.5	Checked cholesterol in last 2 years	Has support
AGE	AGE>50.5	Age of the individual > 50.5 years	Has support
PCS12	PCS12<49.62	Composite physical health score < 49.62	Has support
Q5_11	Q5_11<1.5	Has diagnosed high blood pressure	Has support
Q46	Q46<=2	Felt downhearted and blue most of the time	Interesting
EXCERCIS	EXCERCIS<=2	Does moderate and vigorous activities	Has support
DBDPANX2	DBDPANX2=31	Felt downhearted and blue without depression or anxiety	Interesting
BMI	BMI>=27.082	Body mass index > 27.082	Has support
Q48	Q48<2.5	Checked blood pressure within past 2 years	Has support
OVERWGT	OVERWGT=1	Is overweight according to BMI value	Has support
ALCWDNODA	ALCWDNODA <95.5	Accomplished less, careless work, down-hearted and blue w/o depression or anxiety	Interesting
Q5_13	Q5_13<1.5	Diagnosed joint problems	Interesting
MCS12	MCS12<59.76	Composite mental health score	Irrelevant
Q55	Q55<1.5	Had a hysterectomy	Interesting
Q27	Q27>1.5	Has prescriptions for medicine	Has support
HHSIZE	HHSIZE<=2	Household size	Irrelevant
Q47	Q47=1	Had a flu shot	Irrelevant

Table A1 lists the top predictors for identifying that an individual in the regional population has diagnosed high cholesterol (Q5_12=1). The features are listed in descending order of their importance assessed on the basis of change in the classification accuracy after dropping the feature from the subset of data . The predictors are tagged by a medical expert into three categories: (1) predictors make sense and have support in the medical field for that predictor being meaningful (2) predictors make sense and are interesting (something which the medical field has not seen before and (3) predictors does not make any sense from medical perspective.

Table A2: Top predictors identifying not overweight (BMICUTS=0) in a regional population

Feature	***Rule***	***Description***	***Tag***
Q5_12	Q5_12>1.5	Told by doctor: high cholesterol	Interesting
Q31BREC	Q31BREC>=2	More than 3 days of moderate activity per week	Has support
PCS12	PCS12<42.125	Composite physical health score < 42.125	Interesting
Q5_9	Q5_9>1.5	Told by doctor: Heart trouble	Interesting
Q32BREC	Q32BREC>=1	More than 1-2 days of vigorous activity per week	Has support
Q15NEW2	Q15NEW2=5	Diagnosed depression	Interesting
Q3	Q3<=3	Has good general health	Has support
Q49	Q49>=2	Blood cholesterol not checked in last 2 years	Interesting
Q38	Q38>=2	Not accomplished less due to physical health	Has support
DEPRANX2	DEPRANX2>5	No diagnosed depression and anxiety	Interesting
Q35	Q35>1.5	No limitation of moderate activity due to health	Has Support
Q48REC	Q48REC<1.5	Blood pressure not checked in last 2 years	Interesting
Q40	Q40>1.5	Not accomplished less due to mental health	Interesting
MCS12	MCS12>37.625	Composite mental health score > 37.625	Interesting
Q7A	Q7A>=3	More than 1 serving of fruit/vegs per day	Has support

Table A2 lists the top predictors for identifying that an individual in the regional population is not overweight (BMICUTS=0). The feature BMICUTS can take three distinct values: 0(not overweight), 1 (overweight) and 2 (obese). Similarly the predictive model can be used to find the rules for predicting BMICUTS=1 and BMICUTS=2. The features are listed in descending order of their importance assessed on the basis of change in the classification accuracy after dropping the feature from the subset of data.

Table A3: Top predictors for EXCERREC=1: exercise less than 3 times a week

Feature	***Rule***	***Description***	***Tag***
AGE	AGE>=52	Age of the individual >= 52	Has support
Q15NEW2	Q15NEW2=1	Diagnosed depression	Has support
Q3	Q3>=3	Has poor general health	Has support
Q5_9	Q5_9<1.5	Told by doctor: Heart trouble	Has support
OVERWGT2	OVERWGT2=1	Overweight according to BMI value	Has support
ALCWDBNODA	ALCWDBNODA=93	Accomplished less, careless work, downhearted and blue without anxiety or depression	Interesting
BMI	BMI>26.72	Body mass index > 26.72	Has support
PCS12	PCS12<44.176	Composite physical health score < 44.176	Has support
ALDPANX2	ALDPANX2=31	Accomplished less without depression or anxiety	Interesting
Q5_12	Q5_12<1.5	Told by doctor: high cholesterol	Has support
Q27	Q27>1.5	Had prescriptions written for medicines	Has support
DEPRANX2	DEPRANX2=2	Diagnosed depression and anxiety	Interesting
Q45	Q45>2.5	Has lot of energy some of the time	Interesting
BMICUTS	BMICUTS>=1	Overweight or obese	Has support
Q4REC	Q4REC>3.5	Has poor general health	Has support
Q49REC	Q49REC>=3	Blood cholesterol not checked within last 5 yeas	Interesting
Q47	Q47<1.5	Had a flu shot	Irrelevant

Table A3 lists the top predictors for identifying that an individual in the regional population exercises for less than three days a week (EXCERREC=1). The feature EXCERREC can take two values: 1(exercise less than three times per week) and 2 (exercise more than three times per week). Similarly the predictive model can be used to find the rules for predicting EXCERREC=2. The features are listed in descending order of their importance assessed on the basis of change in the classification accuracy after dropping the feature from the subset of data.

Table A4: Top predictors identifying Q69A=1 (Current smoker) in a regional population

Feature	*Rule*	*Description*	*Tag*
CHRONIC	CHRONIC=1	Chronic drinker. 60+ drinks in the last month	Has support
AGE	AGE<=43.5	Age of the individual less than 43.5	Has support
BINGE	BINGE=1	Binge drinker. 5+ drinks on one occasion	Has support
Q44	Q44>=2	Felt calm or peaceful some of the time	Has support
PCS12	PCS12<59.375	Composite physical health score < 59.375	Interesting
Q16NEW2	Q16NEW2=2	Diagnosed anxiety	Has support
Q3	Q3>2.5	Fair general health	Has support
DEPRANX2	DEPRANX2=2	Diagnosed depression and anxiety	Interesting
Q40	Q40<2	Accomplished less due to physical health	Has support
EXCERREC	EXCERREC=1	Les than 3 times of moderate exercise per week	Has support
ALDPANX2	ALDPANX2=31	Accomplished less without depression or anxiety	Interesting
MCS12	MCS12<52	Composite mental health score < 52	Interesting
Q5_16	Q5_16=1	Diagnosed anxiety	Interesting
Q32BREC	Q32BREC<=2	Vigorous activities less than 3 times per week	Has support
Q38	Q38<1.5	Accomplished less due to mental health	Has support
CWDPANX2	CWDPANX2=31	Careless work	Interesting
DOLLARS	DOLLARS < 28700	Income in dollars less than 28700 per year	Interesting
ALCWNODA	ALCWNODA=62	Accomplished less and careless work without depression or anxiety	Interesting

Table A4 lists the top predictors for identifying that an individual in the regional population is a current smoker (Q69A=1). The feature Q69A can take four values: 1(yes), 2(no), 7(don't know/not sure) and 9 (refused). Similarly the predictive model can be used to find the rules for predicting Q69A=2. The features are listed in descending order of their importance assessed on the basis of change in the classification accuracy after dropping the feature from the subset of data.

Table A5: Top predictors identifying CHRONIC=1 (individual is a chronic drinker)

Feature	***Rule***	***Description***	***Tag***
Q40	Q40<1.5	Accomplished less due to mental health	Has support
AGE	AGE<=32	Age of the individual is less than 32 years	Has support
Q69A	Q69A=1	Currently a smoker	Has support
SEX	SEX=1	Male	Has support
Q44	Q44>3.5	Felt calm or peaceful little of the time	Interesting
Q38	Q38=1	Accomplished less due to physical health	Has support
Q48	Q48<=2	Blood pressure check within past 5 years	Interesting
MCS12	MCS12<=59	Composite mental health score<=59	Interesting
MARITAL	MARITAL>1.5	Marital status: separate, married, widowed	Has support
ALDPANX2	ALDPANX2=31	Accomplished without depression or anxiety	Has support
PCS12	PCS12<44.75	Composite physical health score < 44.75	Has support
Q42	Q42<=3	Interference of work with pain most of the time	Interesting

Table A5 lists the top predictors for identifying that an individual in the regional population is a chronic drinker (CHRONIC=1). The feature CHRONIC can take two values: 0(not a chronic drinker, consumed less than 60 drinks per month) and 1(chronic drinker, consumed more than 60 drinks per month). Similarly the predictive model can be used to find the rules for predicting CHRONIC=0. The features are listed in descending order of their importance assessed on the basis of change in the classification accuracy after dropping the feature from the subset of data.

APPENDIX B: Predictors of mental health variables

Table B1: Top predictors identifying Diagnosed Depression (Q15NEW2=1) in a regional population.

Feature	***Rule***	***Description***	***Tag***
Q5_9	Q5_9<1.5	Has diagnosed heart trouble	Has support
Q38	Q38<1.5	Accomplished less due to physical health	Has support
Q45	Q45>=4.5	Has a lot of energy for little or none of the time	Has support
Q27	Q27>1.5	Has prescriptions for medicines	Has support
Q3	Q3>=3.5	Has fair or poor general health	Has support
AGE	AGE<72.5	Age of the person is less than 72.5	Has support
BMI	BMI>=32.23	Body mass index > 32.23	Interesting
Q31B	Q31B<4.5	Does moderate activity for upto 4 days per week	Interesting
Q4REC	Q4REc>=3.5	General health compared to others is poor	Has support
PCS12	PCS12>=57.3	Composite physical health score >= 57.3	Interesting

Table B1 lists the top predictors for identifying that an individual in the regional population has diagnosed depression (Q5_15NEW2=1). The feature Q5_15NEW2 can take two values: 1(has diagnosed depression) and 10 (does not have diagnosed depression). Similarly the predictive model can be used to find the rules for predicting Q5_15NEW2=10. The features are listed in descending order of their importance assessed on the basis of change in the classification accuracy after dropping the feature from the subset of data.

Table B2: Top predictors identifying Diagnosed Anxiety (Q16NEW2=2) in a regional population.

Feature	***Rule***	***Description***	***Tag***
MCS12	MCS12<39.41	Composite mental health score < 39.41	Has support
PCS12	PCS12<42.17	Composite physical health score < 42.17	Interesting
Q40	Q40<1.5	Accomplishes less due to mental health	Has support
Q44	Q44>4	Felt calm or peaceful only some of the time	Has support
SMOKECIG	SMOKECIG>1.5	Is a current smoker	Has support
Q5_8	Q5_8<1.5	Told by doctor: Headaches	Interesting
Q27	Q27>2.5	Had prescriptions for medicines but never got them filled	Interesting
Q45	Q45>=4	Had lot of energy for some of the time	Has support
Q5_11	Q5_11<1.5	Told by doctor: High blood pressure	Interesting
Q43	Q43<3.5	Interferes with social activity most of the time	Interesting
Q42	Q42<3.5	Interference of pain with work most of the time	Interesting
Q5_12	Q5_12<1.5	Told by doctor: High cholesterol	Has support

Table B2 lists the top predictors for identifying that an individual in the regional population has diagnosed anxiety (Q5_16NEW2=2). The feature Q5_16NEW2 can take two values: 2(has diagnosed anxiety) and 20 (does not have diagnosed anxiety). Similarly the predictive model can be used to find the rules for predicting Q5_16NEW2=20. The features are listed in descending order of their importance assessed on the basis of change in the classification accuracy after dropping the feature from the subset of data.

Table B3: Top predictors identifying Diagnosed Depression and Diagnosed Anxiety (DEPRANX2=2) in a regional population.

Feature	***Rule***	***Description***	***Tag***
Q40	Q40<1.5	Accomplished less due to mental health	Has support
MCS12	MCS12<35.71	Composite mental health score < 35.71	Has support
Q3	Q3>=3.5	Has poor general health	Interesting
Q5_12	Q5_12	Told by doctor: high cholesterol	Interesting
CHRONIC	CHRONIC=1	Is a chronic drinker, 60+ drinks last month	Has support
GENHLTH	GENHLTH>3.5	Has poor general health	Has support
PCS12	PCS12<47.216	Composite physical health score < 47.216	Interesting
Q43	Q43<3.5	Interfere with social activities most of the time	Interesting
Q4	Q4>=4.5	Has poor general health compared to others	Has support
BMI	BMI>28.35	Body mass index > 28.35	Interesting
Q45	Q45>=4.5	Has a lot of energy some of the time	Has support
Q38	Q38<1.5	Accomplished less due to physical health	Interesting

Table B3 lists the top predictors for identifying that an individual in the regional population has diagnosed depression and diagnosed anxiety (DEPRANX2=1). The feature DEPRANX2 can take two values: 2(has diagnosed depression and diagnosed anxiety) and 5 (does not have diagnosed depression and diagnosed anxiety) Similarly the predictive model can be used to find the rules for predicting DEPRANX2=5. The features are listed in descending order of their importance assessed on the basis of change in the classification accuracy after dropping the feature from the subset of data.

Table B4: Top predictors identifying Accomplished Less without depression or anxiety (ALDPANX2=31) in a regional population.

Feature	***Rule***	***Description***	***Tag***
MCS12	MCS12<41.297	Composite mental health score < 41.297	Has support
Q39	Q39<1.5	Limited in kind of work	Has support
PCS12	PCS12<=59.6	Composite physical health score <= 59.6	Interesting
Q43	Q43<5.5	Interfere with social activities most of the time	Interesting
Q45	Q45>=3.5	Has lot of energy little or none of the time	Has support
CWDPANX2	CWDPANX2 <32.5	Careless work without depression or anxiety	Has support
Q48	Q48<1.5	Checked blood pressure with past 2 years	Interesting
Q42	Q42>=1.5	Moderate interference of pain with work	Has support
Q38	Q38<1.5	Accomplished less due to physical health	Has support
Q34	Q34<2.5	Limited moderate activities: a lot	Interesting
DBDPANX2	DBDPANX2 = 31	Downhearted an blue without depression or anxiety	Has support
Q4	Q4>=3.5	Poor general health compared to others	Has support
Q25	Q25<1.5	Needed but did not get medical care	Interesting
Q55	Q55<1.5	Had a hysterectomy	Interesting
CHRONIC	CHRONIC=1	Is a chronic drinker, 60+ drinks last month	Has support
DOLLARS	DOLLARS<= 17500	Income in dollars per year	Interesting

Table B4 lists the top predictors for identifying that an individual in the regional population accomplishes less without depression or anxiety (ALDPANX2=31). The feature ALDPANX2 can take two values: 31(yes) and 33(no). Similarly the predictive model can be used to find the rules for predicting ALDPANX2=33. The features are listed in descending order of their importance assessed on the basis of change in the classification accuracy after dropping the feature from the subset of data.

Table B5: Top predictors identifying Careless Work without Depression and Anxiety (CWDPANX2=31) in a regional population.

Feature	***Rule***	***Description***	***Tag***
Q38	Q38=1	Accomplished less due to physical health	Has support
Q40	Q40<1.5	Accomplished less due to mental health	Has support
Q39	Q39<1.5	Is limited in kind of work	Has support
MCS12	MCS12<=43.5	Composite mental health score<=43.5	Has support
Q45	Q45>=3.5	Has lot of energy little or none of the time	Has support
OVERWGT2	OVERWGT2 >0.5	Is overweight according to the BMI value	Interesting
Q42	Q42>3	Moderate to extreme interference of pain with work	Interesting
EXCERREC	EXCERREC <1.5	preciser less than 3 times per week	Interesting
Q3	Q3>3.5	Has poor general health	Interesting
Q27	Q27<2	Did not have prescriptions for medicine for some of the time	Interesting
AGE	AGE<54	Age of the individual < 54	Interesting
Q5_8	Q5_8<1.5	Told by doctor: headaches	Interesting

Table B5 lists the top predictors for identifying that an individual in the regional population did careless work without depression or anxiety(CWDPANX2). The feature CWDPANX2 can take two values: 31(yes) and 34(no). Similarly the predictive model can be used to find the rules for predicting CWDPANX2=34. The features are listed in descending order of their importance assessed on the basis of change in the classification accuracy after dropping the feature from the subset of data.

Table B6: Top predictors identifying Downhearted and Blue without depression and anxiety(DBDPANX2=31) in a regional population.

Feature	***Rule***	***Description***	***Tag***
Q40	Q40<1.5	Accomplished less due to mental health	Has support
Q44	Q44>=3	Felt calm or peaceful some of the time	Interesting
Q3	Q3>3.5	Has poor general health	Has support
Q69A	Q69A<1.5	Smokes currently	Interesting
Q38	Q38<1.5	Accomplished less due to physical health	Has support
Q39	Q39<1.5	Is limited in kind of work	Has support
Q35	Q35=1	Moderate activities limited due to health	Interesting
OVERWGT2	OVERWGT2 =1	Is overweight according to BMI value	Has support
Q4	Q4>2.5	Has fair or poor general health compared to others	Has support
Q25	Q25<1.5	Needed but did not get medical care	Interesting
Q49	Q49<=2	Has checked cholesterol in past 2 years	Interesting
Q49REC2	Q49REC2=1	Has checked cholesterol in past 2 years	Interesting

Table B6 lists the top predictors for identifying that an individual in the regional population is downhearted and blue without depression or anxiety (DBDPANX2=31). The feature DBDPANX2 can take two values: 31(yes) and 33 (no). Similarly the predictive model can be used to find the rules for predicting DBDPANX2=35. The features are listed in descending order of their importance assessed on the basis of change in the classification accuracy after dropping the feature from the subset of data.