CLUSTERING AND PREDICTIVE MODELING: AN ENSEMBLE APPROACH

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

_____
Philicity K. Williams

Certificate of Approval:

_____	_____
Juan Gilbert Associate Professor Computer Science and Software Engineering	Cheryl Seals Assistant Professor Computer Science and Software Engineering
_____	_____
Hari Narayanan Professor Computer Science and Software Engineering	George T. Flowers Interim Dean Graduate School

CLUSTERING AND PREDICTIVE MODELING: AN ENSEMBLE APPROACH

Philicity K. Williams

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama
August 9, 2008

CLUSTERING AND PREDICTIVE MODELING: AN ENSEMBLE APPROACH

Philicity K. Williams

_____
Signature of Author

_____
Date of Graduation

VITA

Philicity K. Williams, daughter of Allen and Ella Williams was born on January 18, 1977 in Tuskegee, AL. She graduated from Booker T. Washington High School with honors in 1995. She attended Southern University and A & M College in the fall of 1996 and received a Bachelor of Science in Computer Science in May 2000. She entered Auburn University in the fall of 2003 where she is currently working towards her Ph.D.

THESIS ABSTRACT

CLUSTERING AND PREDICTIVE MODELING: AN ENSEMBLE APPROACH

Philicity K. Williams

Master of Science, August 9, 2008
(B.S., Southern University and A & M College, May 2000)

47 Typed Pages

Directed by Juan E. Gilbert

Today's data storage and collection abilities have allowed the accumulation of enormous amounts of data. Data mining can be a useful tool in transforming these large amounts of raw data into useful information. Predictive modeling is a very popular area in data mining. The results of these type tasks can contain helpful information that can be used in decision making. Ensemble method techniques involve using the results of multiple models in combination. Research has shown that by applying an ensemble method approach to predictive modeling one can increase the model's accuracy. However, these techniques focus on classification data mining algorithms. This research investigates the notion of using a data clustering and predictive modeling approach to increase predictive model accuracy.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank God for he is my strength and through him I can do a11 things. I would like to thank Dr. Juan Gilbert for all his support, encouragement, and patience. The many times when I did not know how I was going to make it, or didn't think I could do it, he was there to guide me. Thank you to my family for their love, support, and encouragement. They have been very supportive an encouraging throughout my academic tenure here at Auburn. Last but not least I would like to say thank you to all of the members of the Human Centered Computing Lab. There support, encouragement, and just being there will forever be appreciated!!

Style manual or journal used Journal of SAMPE

Computer software used Microsoft Word 2003

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Advances in data collection and storage capabilities have enabled organizations, companies, institutions, etc to collect immense amounts of data. The challenge lies in trying to turn all of that raw data into useful information. One way to address this problem is with the use of data mining.

Data Mining can be defined as "sorting through data to identify patterns and establish relationships (TechTarget, 2006)". Data mining blends traditional data analysis methods with sophisticated algorithms for processing large amounts of data. These algorithms search large data repositories in order to find new and useful patterns that might have otherwise been unidentified. Data mining methods can and are being used in many domains to address a variety of tasks.

One of the more popular uses of data mining techniques is predictive modeling. Certain types (i.e. decision trees, neural networks, etc.) of data mining algorithms have the ability to predict future outcomes using the attributes in the data set. For example, let's assume that a long distance company wants to promote its new long distance plan. The company can apply predictive modeling algorithms to specifically target customers who are more likely to sign up for the plan based on their past behavior.

As one could imagine, the accuracy of predictive modeling algorithms is very important. Their results contain information that can be very valuable and have the ability

to help make decisions and drive change. An ensemble method approach involves using the results of individual classifiers in combination. Research says that applying an ensemble method approach will, more often than not, produce a predictive model with higher accuracy than with a single learning algorithm. However, these ensemble method techniques only deal with classification algorithms (i.e. decision trees, neural networks, etc). This research investigates the premise of using a clustering algorithm then a predictive modeling approach.

The rest of this thesis is organized as follows: Chapter 2 contains the literature review which gives an overview of data mining including a discussion of the various data mining tasks and techniques as well as the problem definition. Chapter 3 discusses the experiment procedure, materials/tools, and results. Chapter 4 presents conclusions and ideas for future work.

2. LITERATURE REVIEW

2.1 DATA MINING OVERVIEW

Data mining is the process of discovering useful information in large amounts of data. It is fundamentally built on the principles of algorithms used in statistics, artificial intelligence, pattern recognition, and machine learning. From statistics, data mining incorporates things such as sampling, estimation, and hypothesis testing. From artificial intelligence, pattern recognition, and machine learning, data mining incorporates the following: search algorithms, modeling techniques, and learning theories. Figure 1 conveys data mining's relationship to other areas (Tan, Steinbach, and Kumar, 2006).

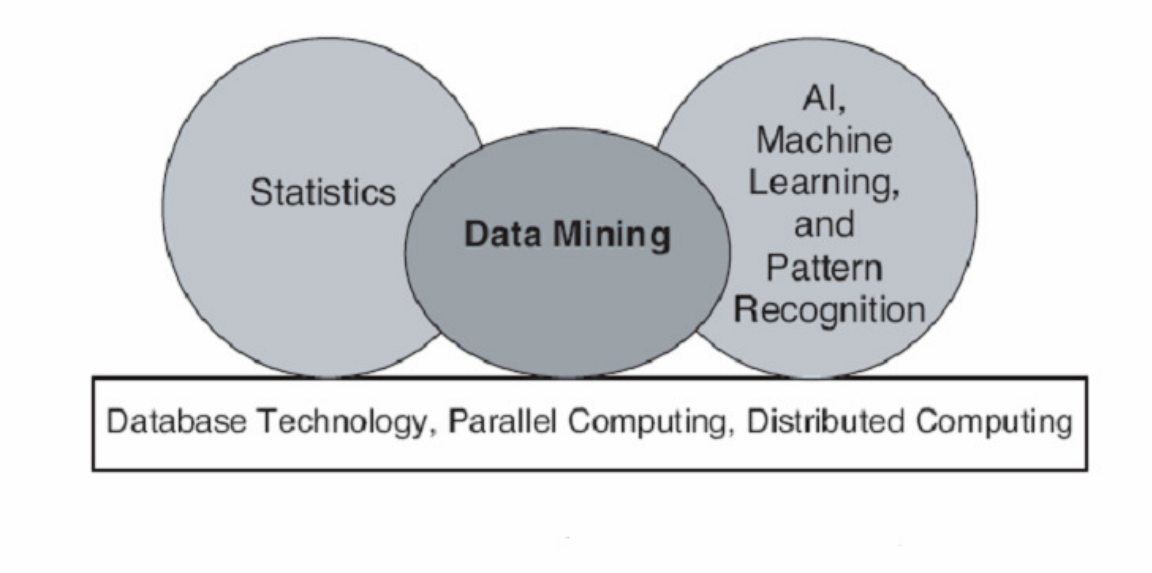


Figure 1 Data mining's relationship to other disciplines

Data mining is being used in many domains in a variety of ways. Due to this, a CRross Industry Standard for Data Mining Projects (CRISP-DM) was developed. This standard is tool, industry, and application neutral. The life cycle according to the standard has six phases: Business/Research Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. Figure 2 depicts CRISP-DM (CRISP-DM, 2000).

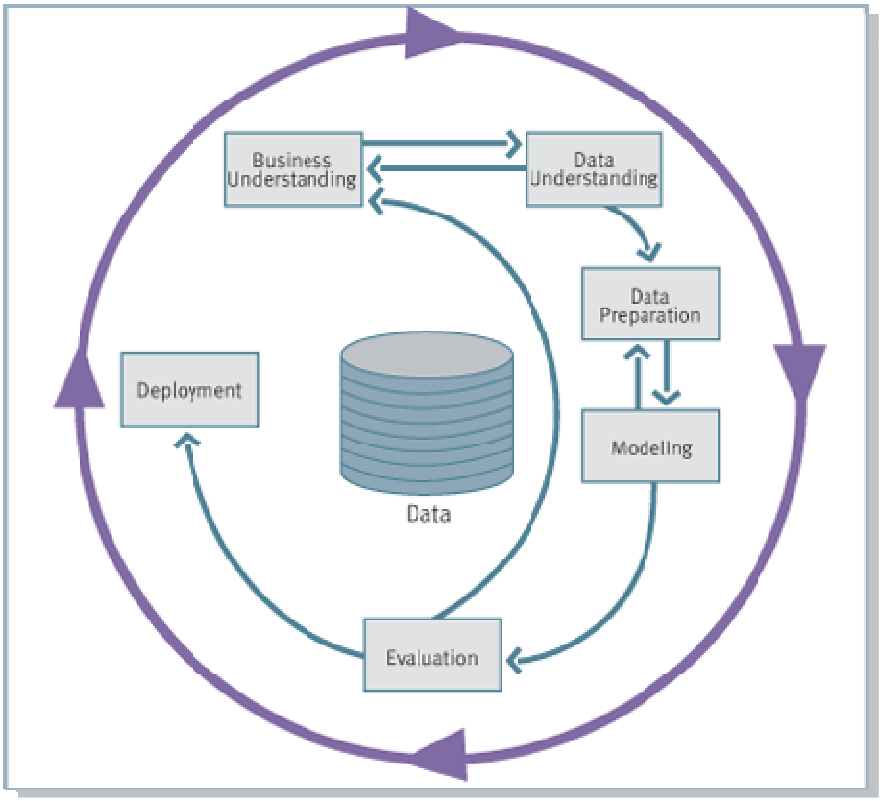
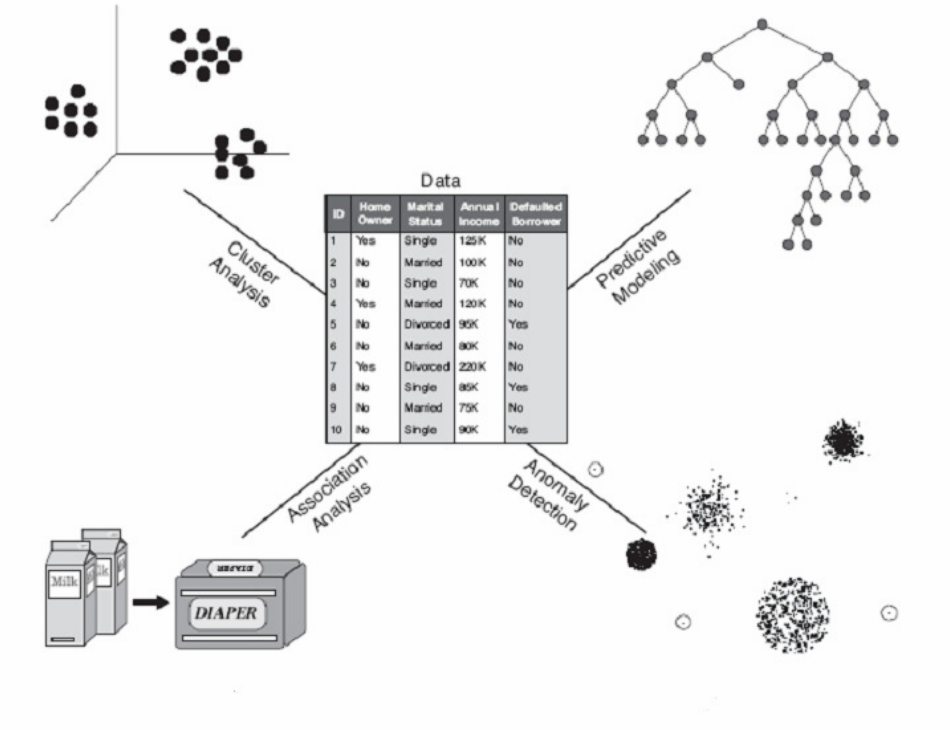


Figure 2 CRISP-DM Model

Most data mining tasks are usually either predictive or descriptive in nature. Predictive tasks are ones in which the goal is to predict the value of a particular attribute based on the values of other attributes. The attribute that is being predicted is frequently referred to as the target or dependent variable. The other attributes that are used for making the prediction are referred to as the explanatory or independent variables.

Descriptive tasks are ones in which the goal is to derive patterns (i.e. correlations, clusters, trends) that will provide a summary of the underlying relationships in the data. These tasks are usually exploratory and often require some post-processing to confirm as well as explain the results.

There are four tasks that are at the heart of data mining. They are Predictive Modeling, Cluster Analysis, Association Analysis, and Anomaly Detection (see Figure 3) (Tan, Steinbach, and Kumar, 2006).



ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Figure 3 Four of the core data mining tasks

2.1.1 PREDICTIVE MODELING

Predictive modeling is one of the most popular subfields in data mining. It is the process of using the patterns found in the data set to predict future outcomes. Algorithms

used for predictive modeling build a model for the dependent variable as a function of the independent variables (Thearling, 2008). There are two types of predictive modeling tasks: classification and regression. Classification tasks are used when the target or dependent variable is discrete. Regression tasks are used when the target variable is continuous (Tan, Steinbach, and Kumar, 2006).

Predictive modeling problems are composed of four main things: a dependent variable, independent variables, a learning/training data set, and a test data set (Lewis, 2000). The learning/training data set contains values for both the dependent and independent variables. This data is used to build the model. This model is then applied to the test set for evaluation. The test set is a subset of the training/learning data set. The performance of the model is based on the counts of the test records that are correctly or incorrectly predicted or classified.

There are several predictive modeling techniques. The next section provides an overview of some of them.

2.1.1.1 DECISION TREES

A decision tree is a predictive model in which the results are displayed as a tree type structure (Thearling, 2008). A decision tree consists of a collection of decision nodes, which are connected by branches, descending from the root node until coming to an end at the leaf nodes. Each branch of the tree is a classification question and the leaves are the partitions or segments of the dataset with the classification or decision (Larose, 2005).

The first step in the decision tree building process is to "grow" the tree. The goal is to create a tree that works as close to perfect as possible with the data provided. When growing the tree, the main task is to find the best question to ask at each branch or split point in the tree. This task is repeated until there is either only one record in the segment, each of the records in the segment are the same, or there is not any significant gain in making a split. When one of these conditions are met the tree stops growing (Thearling, 2008). Figure 4 conveys a simple decision tree example (DMS, 2008).

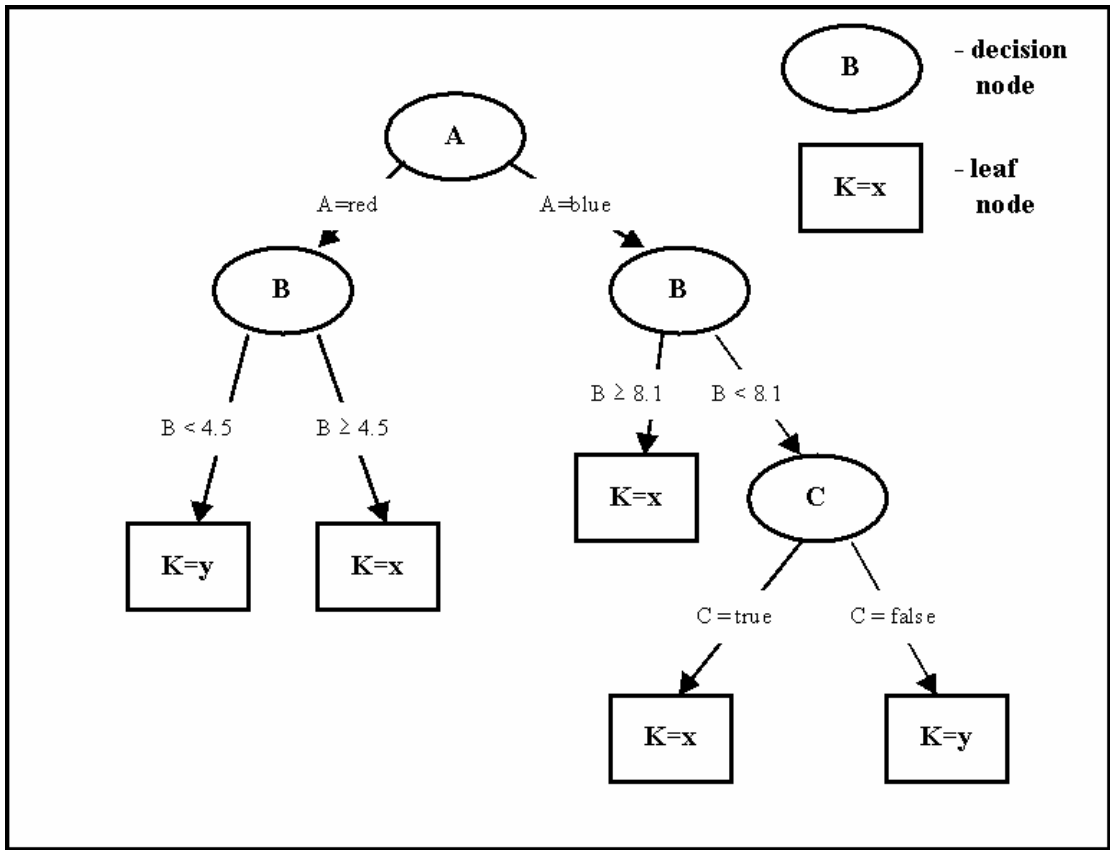


Figure 4 Simple Decision Tree

There are certain requirements that must be adhered to in order to apply a decision tree algorithm. A training data set must be provided. This training data should be varied in nature and provide the algorithm with a robust sampling of the types of records that

will need classification in the future. This is very important as decision trees learn by example. If the algorithm is provided bad data then the ability to correctly classify future data will greatly diminish. Also, the target variable must be discrete (Larose, 2005).

Decision trees are a part of data mining technology that has existed in some form for decades. Originally, these techniques were created for statisticians to automate the process of determining which attributes were useful or in correlation with the problem they were attempting to solve or understand. They are also particularly adept at handling raw data with little or no pre-processing (Thearling, 2008).

Perhaps the most appealing aspect of decision trees is how easy they are to interpret. This is especially evident when deriving decision rules. Rules can be derived by simply traversing the tree from the root to any leaf node and come in the form of *if antecedent, then consequent*. The antecedent contains the attribute values from the branches taken by a particular path throughout the tree. The consequent contains the classification value for the dependent variable given by that particular leaf node (Larose, 2005). Using the simple decision tree in figure 4 an example of a decision rule would be if A = red and B < 4.5 then K = y. There are several algorithms that are used to produce decision trees. These include: ID3, C4.5, Classification and Regression Trees (CART), and Chi-Square Automatic Interaction Detector (CHAID). The next sections describe each.

2.1.1.1.1 ID3

Developed in the 1970's, ID3 was one of the first decision tree algorithms. Initially, the algorithm was used for things like learning good game play strategies for

chess end games. In chess, the end game is the portion of the game where there are only a few pieces left on the board (Wikipedia, 2008). However, ID3 has been used for a variety of tasks in various domains and has been modified and improved many times.

ID3 works by choosing the predictors and their corresponding split values based on the gain in information that the split(s) will make available. Gain is the difference in the amount of information that will be needed to properly make an accurate prediction before and after a split(s) is made (Thearling, 2008).

2.1.1.1.2 C4.5

C4.5 is an enhancement of the ID3 algorithm. It improved ID3 algorithm by allowing predictors with missing or continuous values to be used, introducing tree pruning, and enabling rule derivation. The trees produced by this algorithm are also not restricted to binary splits. These trees are more variable in shape (Thearling, 2008 & Larose, 2005).

2.1.1.1.3 CLASSIFICATION AND REGRESSION TREES

Classification and Regression Trees (CART) was developed in 1984 by Leo Breiman, Jerome Friedman, Richard Olshen and Charles Stone (Brieman, Friedman, & Olshen, 1984). Many of the C4.5 techniques appear in CART. The trees that are produced by the CART algorithm are strictly binary. This means that each node in the tree can only have two branches. CART analysis is a form of recursive partitioning. The algorithm recursively partitions the data in the training set into subsets of records with similar values for the target or dependent variable. To grow the tree, CART performs an exhaustive search of available variables to pick the best or most optimal split variable.

The fully grown tree that is initially produced will yield the lowest error rate when compared against the training data. However, this model may be too complex and usually results in overfitting. An overfit model is one that too closely follows all of the traits of the training data and is not general enough to represent the overall population (Thearling, 2008, Larose, 2005 & Lewis, 2000).

CART uses a cross validation approach and pruning to combat overfitting. Cross validation, also know as leave-one-out training, is a method for confirming a procedure for building a model that is computationally intensive (Lewis, 2000). In cross-validation, the training data is divided randomly into N segments, stratified by the variable of interest. One of the segments is set aside for use as an independent test data set. The other (N-1) segments are combined for use as the training data set. The entire model building process is repeated N times, using a different segment of data as the test data each time. N different models are produced, each of which that can be tested against the independent test data. Based on the results of the cross-validation the initial complex tree is pruned to produce the most optimal general tree. The most complex tree rarely performs best on the test data. By using cross validation, the tree that is most likely to perform well on new data is produced (Lewis, 2000).

2.1.1.1.4 CHI-SQUARE AUTOMATIC INTERACTION DETECTOR

Chi-Square Automatic Interaction Detector (CHAID) was published in 1980 by Gordon V. Kass [10]. CHAID is very similar to CART but differs in the way splits are determined. The algorithm uses the chi square test to choose the best split. The chi square test is used in contingency tables to determine which categorical predictor is the

furthest from independence with the prediction values. All predictors must either be categorical or forced into a categorical form, because of CHAID's reliance on contingency tables, to shape its test of significance for each predictor (Thearling, 2008).

2.1.1.2 NAIVE BAYESIAN CLASSIFIERS

Bayesian classifiers are based on the Bayes theorem. This theorem is a simple mathematical formula that is used for calculating conditional probabilities. The naïve Bayes classifier learns the conditional probability for each attribute given a particular class label. For example, if A and B are two random events then P(A|B) is the conditional probability of A occurring given B. Classification is performed by using the Bayes theorem to calculate the probability of a class given a particular instance. The class with the highest posterior probability is then used as the prediction (Friedman & Goldszmidt, 1996; Langley & Sage, 1994; Friedman, Geiger, & Goldszmidt, 1997). Using the previous example, P(A|B) is the posterior probability.

2.1.1.3 NEURAL NETWORKS

An example of an authentic neural network would be the human brain. The brain can recognize patterns, make predictions, and learn. Artificial neural networks seek to emulate these capabilities. Neural networks in data mining are essentially computer programs applying pattern recognition and machine learning algorithms to build predictive models (Thearling, 2008).

There are two main structures in a neural network: nodes and links. Nodes are artificial neurons and links are the connections between them. To make a prediction, the neural network accepts values for the independent variables or predictors at the input

nodes. The values of these nodes are then multiplied by values stored in the links. These values are added together at the output node, after which some threshold function is used and the resulting number is the prediction. Figure 5 (Thearling, 2008) is an example of a simple neural network. In this example Age and Income are the input nodes and Default is the output node if a person will default on a bank loan.

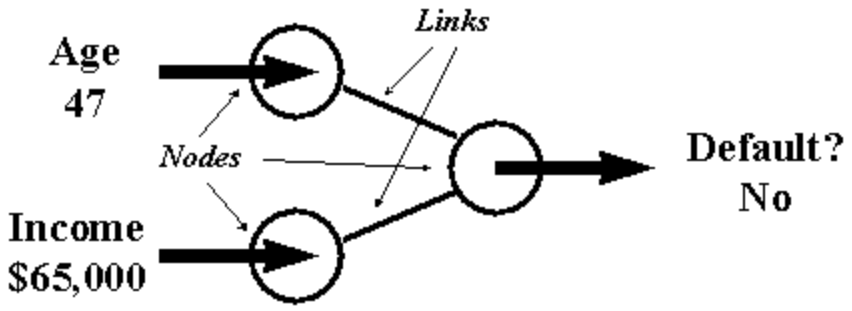


Figure 5 Example of Simple Neural Network

Most neural networks are not as simple as depicted above. There is usually hidden layer of nodes between the input and output nodes. They are deemed "hidden" because their contents are not made known to the end user. It is also possible to have more than one hidden layer, thus making the network very complex.

2.1.2 CLUSTER ANALYSIS

Clustering can be defined as "a division of data into groups of similar objects (Berkhin, 2002)". Instances within these groups or clusters are more similar to each other than instances belonging to other clusters. Clustering and classification are different from one another in that there is no target or dependent variable. Clustering does not attempt to classify or predict the value of the target variable in clustering. These

algorithms attempt to segment the whole data set into homogenous clusters. The more similarity within a cluster and the bigger the difference between clusters the better the clustering. For the best possible performance, clustering algorithms require that the data be normalized so that any one attribute or variable will not control the analysis.

There are two main types of clustering algorithms: hierarchical and partitional. The next section describes each.

2.1.2.1 HIERARCHICAL CLUSTERING

Hierarchical clustering creates a tree of clusters known as a dendrogram. With this type of clustering, the smallest clusters in the tree join together to create the next level of clusters. At this level, the clusters then join together to create the next level of clusters. The top or root of this tree is the cluster that contains all the records. There are two types of hierarchical clustering: agglomerative and divisive (Berkhin, 2002 & Thearling, 2008).

Agglomerative clustering algorithms begin with having as many clusters as there are records. Each cluster will contain one record. Then the clusters that are closest to one another, based on some distance, are joined together to create the next largest cluster. This process is continued until the hierarchy is built with a single cluster, which contains all records, at the top. Figures 6 and 7 describe the agglomerative clustering process (Wikipedia, 2008). Figure 6 contains a set of records that will be clustered. Figure 7 shows the cluster hierarchy after the agglomerative approach has been applied.

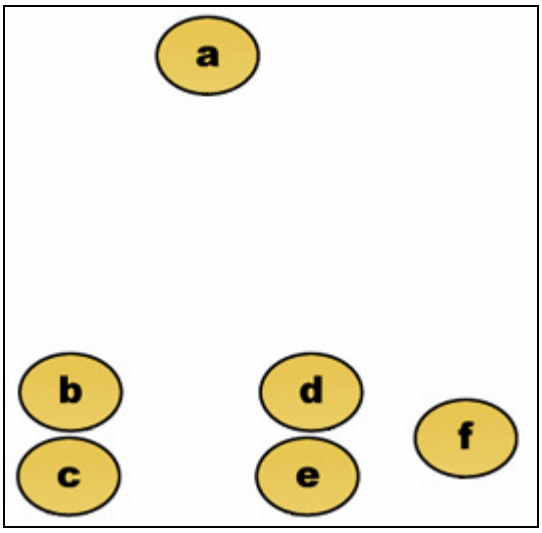


Figure 6 Records to be clustered

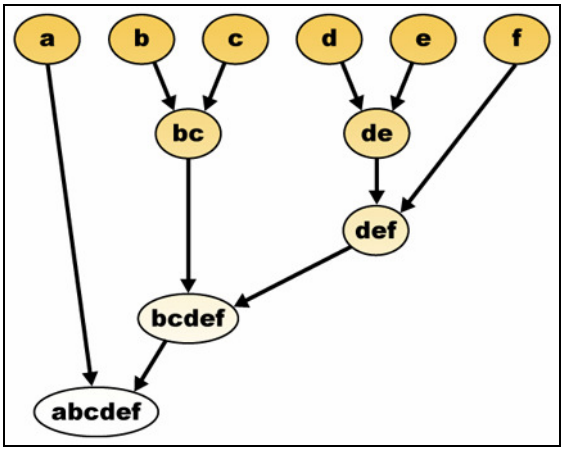


Figure 7 Cluster Hierarchy

Divisive clustering algorithms apply an opposite approach. These types of algorithms start will all of the records in one cluster and then recursively splits the most suitable cluster. This process continues until some stopping criteria is met.

How appropriate a cluster or clusters for merging or splitting depends on how similar or dissimilar items in each cluster are. The distance between individual records has to be generalized to the distance between clusters in order for splitting or merging to occur. This proximity measure is called a linkage metric. The major linkage metrics

include: Single Linkage, Complete Linkage, and Average Linkage (see Figure 8) (Tan, Steinbach, and Kumar, 2006 & Berkhin, 2002).

Single linkage (nearest neighbor) is based on the minimum distance between any record in one cluster and any record in another cluster. Cluster similarity is based on the similarity of the most similar members from each cluster.

Complete linkage (farthest neighbor) is based on the maximum distance of any record in one cluster and any record in another cluster. Cluster similarity is based on the similarity of the most dissimilar members from each cluster.

Average linkage was designed to decrease the dependence of the cluster linkage criteria on extreme values. The criteria here is the average distance of all the records in one cluster from all of the records in another cluster.

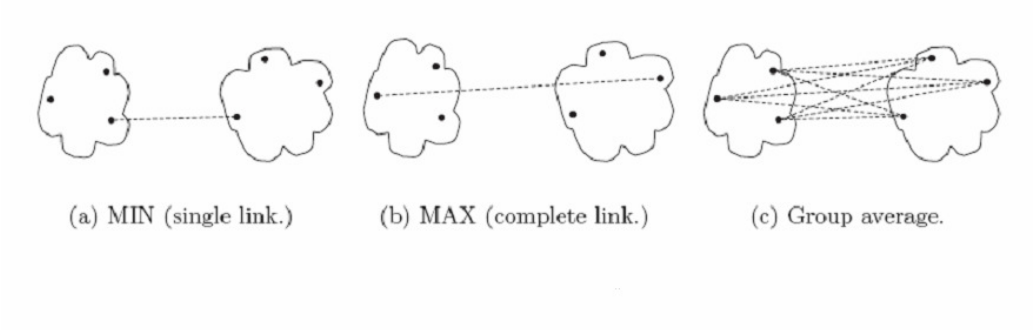


Figure 8 Linkage Examples

2.1.2.2 PARTITIONAL CLUSTERING

Partitional clustering is dividing the data set in such a way that each record belongs to one and only one cluster. These algorithms don't produce dendrogram as in hierarchical clustering. A single partition of the data is produced. Partitional clustering algorithms begin with a randomly picked or user defined number of clusters. The

algorithms then optimize each cluster based on some validity measure. There are several partitioning clustering approaches. The next section discusses two of these: K-Means and Expectation Maximization (EM) (Berkhin, 2002).

2.1.2.2.1 K-MEANS

K-means is one of the oldest and most widely used clustering techniques. The name K-means comes from how each of the *K* clusters is represented by the mean of the points within that cluster. This point is called the centroid. The basic K-means algorithm is very simple and straight forward. Figure 9 describes this algorithm (Tan, Steinbach, and Kumar, 2006).

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

Figure 9 Basic K-means algorithm

The first step is to pick *K* initial centroids. *K* is the number of clusters wanted and is a user specified parameter. Next each data instance or point is assigned to the centroid it is closest to. This collection of points is a cluster. The centroid of each of these clusters is updated based on the points assigned to the cluster. These steps are repeated until there is no change in the centroids.

Instances are assigned to a centroid based on a proximity measure that quantifies closeness for a particular data set. There are several types of proximity measures such as Euclidean distance and Cosine similarity (Berkhin, 2002).

2.1.2.2.2 EXPECTATION MAXIMIZATION (EM)

The EM technique is very similar to the k-means approach. They are both iterative in nature and start with a random guess. The difference lies in the idea of hard versus soft cluster membership. In k-means a hard membership approach is adopted. This means that an instance belongs to one and only one cluster. In EM a soft membership approach is used, which means that membership of an instance can be spread amongst many clusters.

The EM algorithm is a two step process: Expectation and Maximization. The expectation portion is the first step and involves calculating cluster probabilities (the expected class values). The maximization step involves calculating the distribution parameters. This step is the maximization of the possibilities of the distribution given the data. These steps are repeated until a "log-likelihood convergence is achieved." (Berkhin, 2002 ,Witten & Frank 2005).

2.1.3 ASSOCIATION ANALYSIS

Association analysis is useful for finding interesting relationships that are hidden in large data sets. The goal of this type of analysis is to uncover rules (or associations) for quantifying the relationship between two or more attributes (Larose, 2005). These relationships are displayed in the form of an association rule. An association rule is an implication expression of the form: $X \rightarrow Y : \sup_X^Y , conf_X^Y$, where X and Y are disjoint itemsets (i.e., $X \cap Y = \emptyset$), $\sup_X^Y$ is the support, and $conf_X^Y$ is the confidence.

The strength or goodness a rule is measured by its support and confidence. Support defines how many times a rule is pertinent to a particular data set. Confidence defines how often item in Y appear in instances that contain X. For example, a store may find that out of 100 customers shopping on a Tuesday night, 20 bought diapers, and of the 20 who bought diapers, 5 bought beer. The association rule for this example would be if buy diapers then buy beer. This rule would have a support of 5/100 = 5% and a confidence of 5/20 = 25% (diapers → beer: 0.05, 0.25) (Tan, Steinbach, and Kumar, 2006).

Mining association rules from large data repositories involves a two step process: frequent itemset generation and rule generation. First, all of the frequent itemsets must be found. Frequent itemsets are those that satisfy some minimum support threshold. Then, from the frequent item sets found in the first step, association rules are generated that satisfy the minimum support and confidence conditions. In association analysis an itemset is a collection of zero or more items.

Frequent itemset generation is generally more expensive than rule generation. A dataset that has *k* items could produce up to 2^k -1 itemsets. Because *k* can be quite large, the search space of itemsets that needs to be investigated is exponentially large. To enumerate the list of all possible itemsets a lattice structure can be used (Figure 10). The brute force method of finding frequent itemsets involves determining the support count for each candidate itemset in the lattice structure. There are ways of reducing the computational complexity of frequent itemset generation. One of these is to reduce the number of candidate itemsets using the Apriori principle. The Apriori principle states that if an itemset is frequent, then all of its subsets must also be frequent (Figure 11). On the other hand, if an itemset is infrequent then all of its supersets must be infrequent too. For

example in Figure 12, because {a,b} is an infrequent itemset the entire subgraph containing the supersets {a,b} can be pruned. This method of reducing the exponential search space based on the support measure is known as support-based pruning. Apriori was the first rule mining algorithm which pioneered the use of support based pruning (Tan, Steinbach, and Kumar, 2006). This algorithm uses support based pruning to control the exponential growth of candidate itemsets. Apriori uses a bottom up technique in which the frequent item sets are extended one at a time. This is known as candidate itemset generation. This process continues until no further successful extensions are found (Wikipedia, 2008).

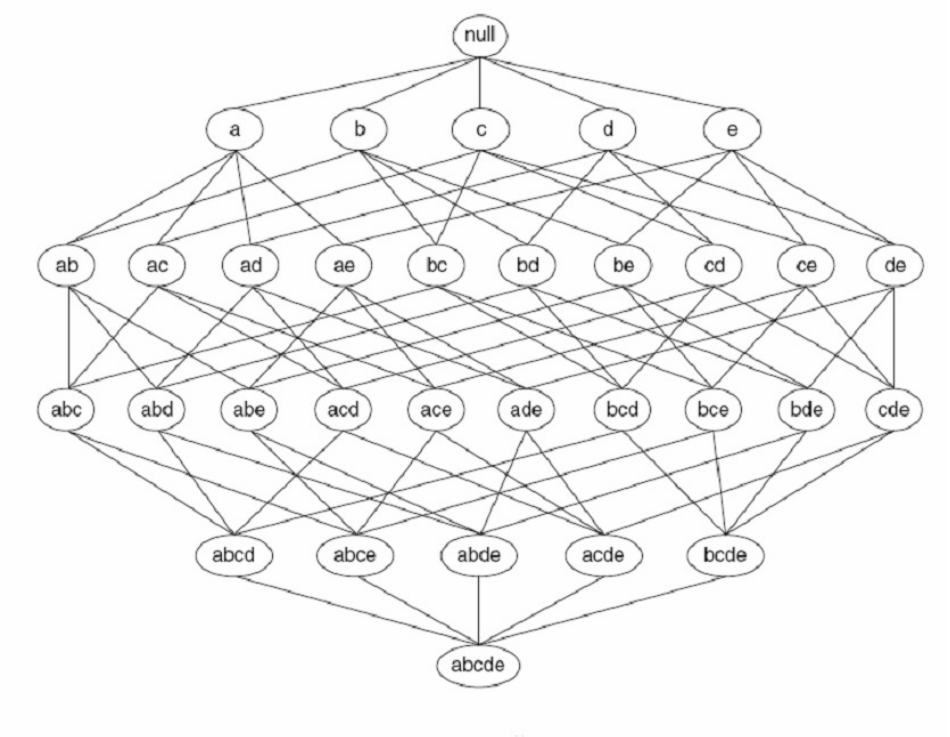


Figure 10 Lattice Structure

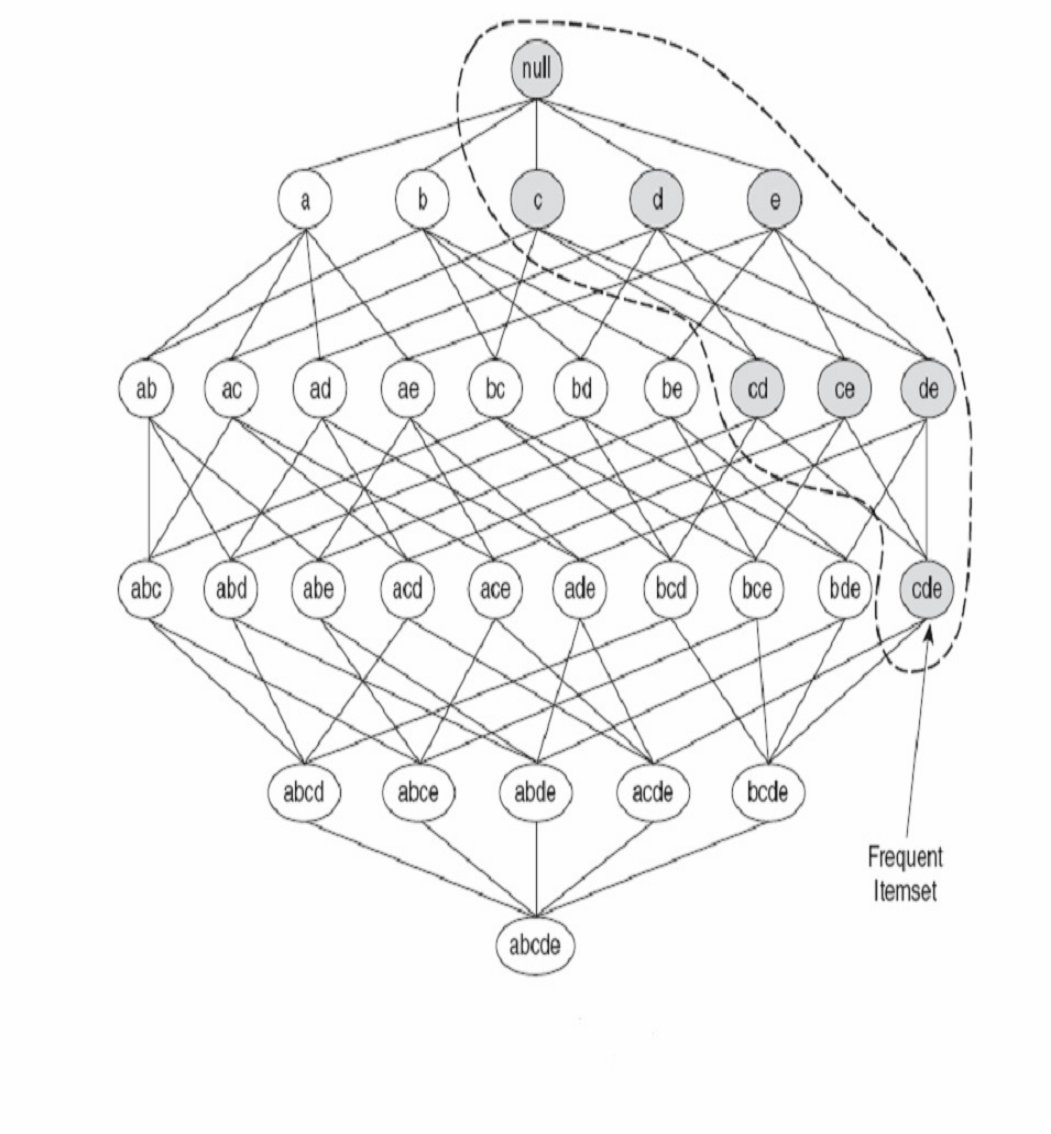


Figure 11 Apriori Principle

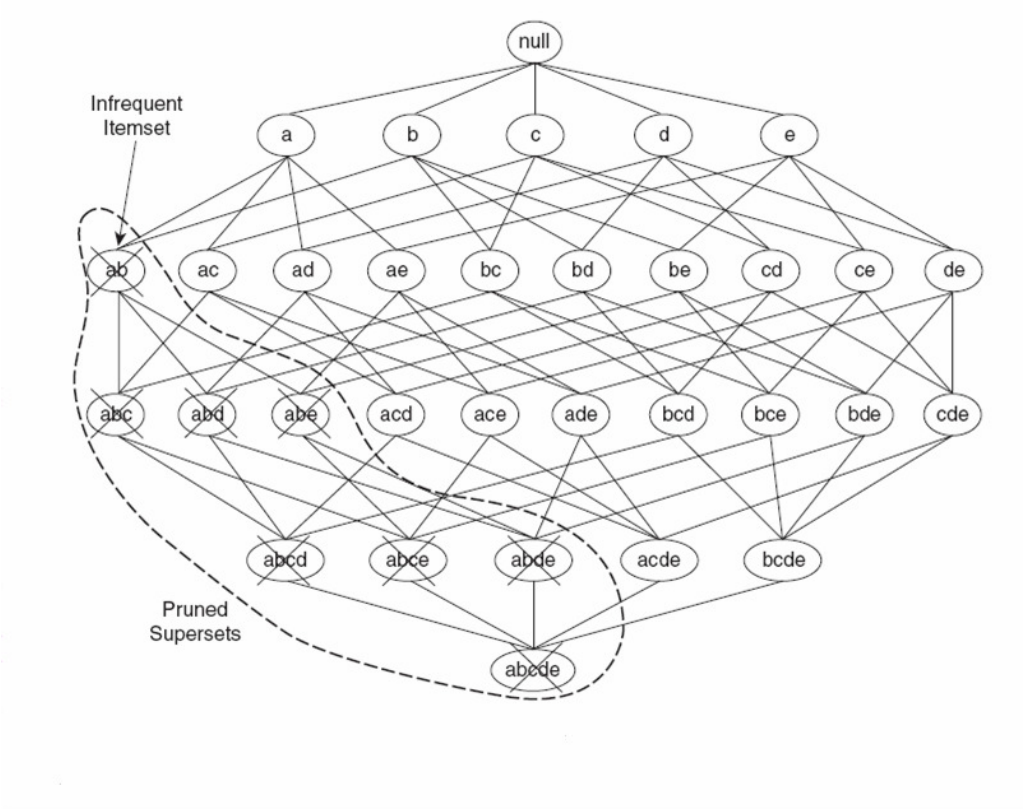


Figure 12 Support based pruning

2.1.4 ANOMALY DETECTION

The main objective in anomaly detection is to locate instances that are different from most of the other instances. These abnormal objects are known as outliers and have attribute values that deviate considerably from the norm or expected values. Some areas where this type of analysis is very important are fraud detection, intrusion detection, public health and medicine. Some common causes of anomalies are things such as data from different types, natural variation and data measurement or collection errors (Tan, Steinbach, and Kumar, 2006).

There are three fundamental approaches to anomaly detection: supervised, unsupervised, and semi-supervised. The difference in these techniques is the degree to which known outcomes or classifications (class labels) are available for at least some portion of the data (Tan, Steinbach, and Kumar, 2006).

The supervised anomaly detection approach requires a training set that contains both normal and abnormal instances. Unsupervised anomaly detection techniques seek to assign a score to each instance that reflects how abnormal that instance is. With semi-supervised anomaly detection techniques the goal is to find an anomaly score (label) for a set or group of instances using information from labeled normal instances.

2.2 *ENSEMBLE METHODS*

As one could imagine the accuracy of predictive modeling algorithms is quite important. Ensemble methods seek to improve the accuracy of classifiers (such as decision trees) by combining the predictions of multiple classifiers (see Figure 13). These methods construct a set of base classifiers using the training data and makes classifications for new instances by voting on the prediction each base classifier make (Tan, Steinbach, and Kumar, 2006). Research has shown that ensembles have a tendency to perform better than single classifiers in the ensemble (Opitz & Macline 1999;Quinlan, 2006, Freund & Schapire, 1999; Berk, 2004). Bagging, boosting, and stacking are well known ensemble techniques.

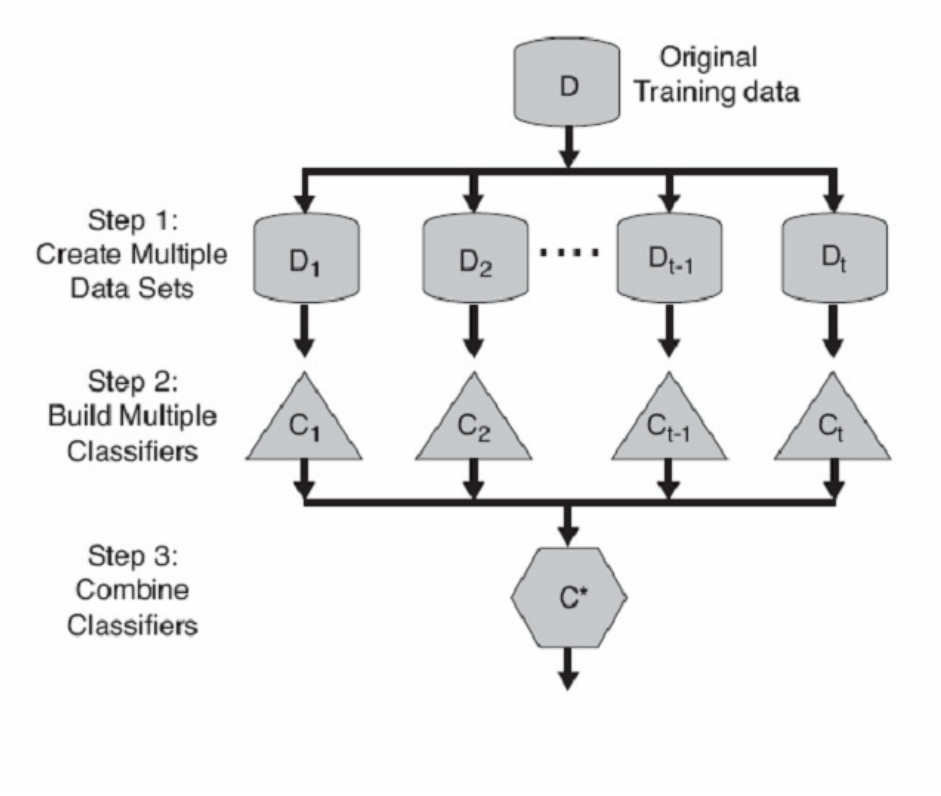


Figure 13 View of Ensemble Method

2.2.1 BAGGING

Bagging or bootstrap aggregation trains each base classifier with a random redistribution of the training (Opitz & Macline, 1999). Each of the training sets for the base classifiers are generated by randomly selecting, with replacement, N examples, where N is the size of the original training set. Since sampling is done with replacement, some of the instances may possibly appear many times within the same training set. Likewise, some instances may be omitted from the training set (see Figure 14).

As previously stated, bagging involves combining multiple models (for example decision trees). These models can be combined by having each model vote on each new instance. The class that receives the most votes is deemed to be the correct one. When

the target variable is numeric the average is used. The predictions or classifications made by voting become more dependable as more votes are considered. Researchers have determined that bagging is effective on the learning algorithms of which small changes in the training data can produce big changes in predictions (Tan, Steinbach, and Kumar, 2006; Witten & Frank, 2005; Opitz & Macline, 1999).

2.2.2 BOOSTING

This ensemble method technique gets its name from its capacity to take a "weak learning algorithm" and 'boosting" it into a "strong" learning algorithm (Freund & Schapire, 1999). A weak learning algorithm is one that performs slightly better than random guessing. Like bagging, boosting uses voting to combine the output of multiple models of the same type (for example decision trees). Boosting, however, is iterative, unlike bagging in which each base classifier is built separately; each new model is influenced by the performance of the models built before it. New models are pushed to become experts for instances mishandled by earlier models. Boosting weights a model's contribution by its performance instead of giving equal weight to all as in bagging (Witten & Frank, 2005). For example in Figure 14 (Opitz & Macline, 1999), assume instance 1 is an outlier and hard to classify correctly. This instance appears more in later training sets because boosting will focus (increase its weight) more on correctly classifying it.

A sample of a single classifier on an imaginary set of data.	
	(Original) Training Set
Training-set-1:	1, 2, 3, 4, 5, 6, 7, 8

A sample of Bagging on the same data.	
	(Resampled) Training Set
Training-set-1:	2, 7, 8, 3, 7, 6, 3, 1
Training-set-2:	7, 8, 5, 6, 4, 2, 7, 1
Training-set-3:	3, 6, 2, 7, 5, 6, 2, 2
Training-set-4:	4, 5, 1, 4, 6, 4, 3, 8

A sample of Boosting on the same data.	
	(Resampled) Training Set
Training-set-1:	2, 7, 8, 3, 7, 6, 3, 1
Training-set-2:	1, 4, 5, 4, 1, 5, 6, 4
Training-set-3:	7, 1, 5, 8, 1, 8, 1, 4
Training-set-4:	1, 1, 6, 1, 1, 3, 1, 5

Figure 14 Bagging and Boosting Example

2.2.3 STACKING

Stacking or Stacked Generalization differs from bagging and boosting in that it involves combining classifiers of different types (e.g., decision trees and neural networks). It is less commonly used because it can be difficult to analyze and there is no standard or accepted best practice. Stacking eliminates voting by introducing the idea of a metalearner. Stacking attempts to "learn" which base classifiers are the most reliable by using the metalearner to determine how best to combine the output of the base classifiers (Witten & Frank, 2005).

The inputs to the metalearner (level 1 model) are the predictions of the base classifiers (level 0 model). Each level 1 instance will have as many attributes as there are level 0 learners. When classifying, the new instance is given to each of the level 0 learners and those results are given to the level 1 learner. The level 1 learner then "learns" the best way to combine the predictions to make the final prediction (Witten & Frank, 2005).

2.3 *DATA MINING TOOLS*

There are many data mining tools available. This section gives a brief description of some of them.

2.3.1 MICROSOFT SQL SERVER

Microsoft SQL Server is a "comprehensive, integrated data management and analysis software that enables organizations to reliably manage mission-critical information and confidently run today's increasingly complex business applications. (Microsoft, 2007) " SQL Server 2005 is the platform leader in a variety of areas including business intelligence and database management systems.

2.3.2 RAPID MINER

"Rapid Miner (formerly YALE) is the world-leading open-source system for knowledge discovery and data mining (Rapid-I, 2008)". This tool is not only available as a stand-alone application for data analysis but also as a data mining engine that can be integrated into your own products.

2.3.3 WEKA

Weka (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software that is written in Java. The software was developed at the University of Waikato and is free under the GNU General Public License (Wikipedia, 2008).

2.3.4 ORACLE DATA MINING

Oracle Data Mining (ODM) is an option of Oracle Database 10*g* Enterprise Edition. This tool provides the ability to "produce actionable predictive information and build integrated business intelligence applications (Oracle, 2007)".

2.4 *PROBLEM DEFINITION*

As stated previously the accuracy of predictive modeling algorithms is very important. The results of these models contain information that can be very valuable and useful in aiding decision making and driving change. Ensemble methods (e.g., bagging, boosting and stacking) have been proven to produce a model that will outperform single models. However, these methods only deal with combining classification algorithms. This research investigates the idea of using clustering and predictive modeling as an ensemble. The hypothesis is that by clustering the data set then applying a predictive model to each cluster and combining the results of each cluster's predictive model we will produce a model that will have a higher accuracy than the predictive model alone.

3. EXPERIMENT

A preliminary experiment was conducted to determine if using a clustering then predictive modeling approach could possibly produce a model or set of rules that has a higher prediction accuracy rate than a single classifier.

3.1 *DATA*

The data for this experiment was taken from the UCI data repository.[22] The breast cancer data set was used. This set contained 286 records with two classification possibilities: recurrence (85) and no recurrence (201). Eighty percent of the data was used for the training set and the remaining twenty percent was used for the test data set. Proportionate samples were used for the training and test sets. The training set contained 161 (201 * 80%) no recurrence and 68 recurrence records. Similarly, the test set contained 40 no recurrence and 17 recurrence records.

3.2 *MATERIALS/TOOLS*

Microsoft SQL Server 2005 Business Intelligence, SQL Server Management Studio and Microsoft Excel were used for all data storage and manipulation for this experiment.

3.3 *PROCEDURE*

Both the training and test data sets were imported into MS SQL Server database tables. The training set was then used to build the decision tree predictive model. The

default decision tree algorithm used in MS SQL Server is CART. All system defaults were used to build the decision tree model. This model was then evaluated against the test data set using MS SQL server. The Microsoft Clustering portion of SQL server was then run with the training data. The default clustering algorithm used in this tool is EM and the number of clusters created was 10. The data within each of these clusters was saved into Excel files and imported into SQL server database tables. The Microsoft Decision tree algorithm was then run against each of these tables creating 10 different decision tree models. The minimum coverage for this portion of the experiment was changed to 1 in order for the trees to properly split. All other system defaults remained the same. These models were then combined and evaluated manually against the test data set.

3.4 *RESULTS*

Figure 15 depicts the decision tree predictive model derived from the training data. Table 1 shows this model's prediction accuracy when validated using the test data. In this chart the columns of the classification matrices correspond to actual values and the rows correspond to predicted values. The predictive model alone had an overall prediction accuracy rate of approx 65%. 37 of the 57 test instances were correctly classified. 36 of the 40 no-recurrence-events instances were correctly classified yielding and accuracy of 90%. Only 1 of the 17 recurrence-events instances was correctly classified yielding and accuracy of approximately 0.06%.

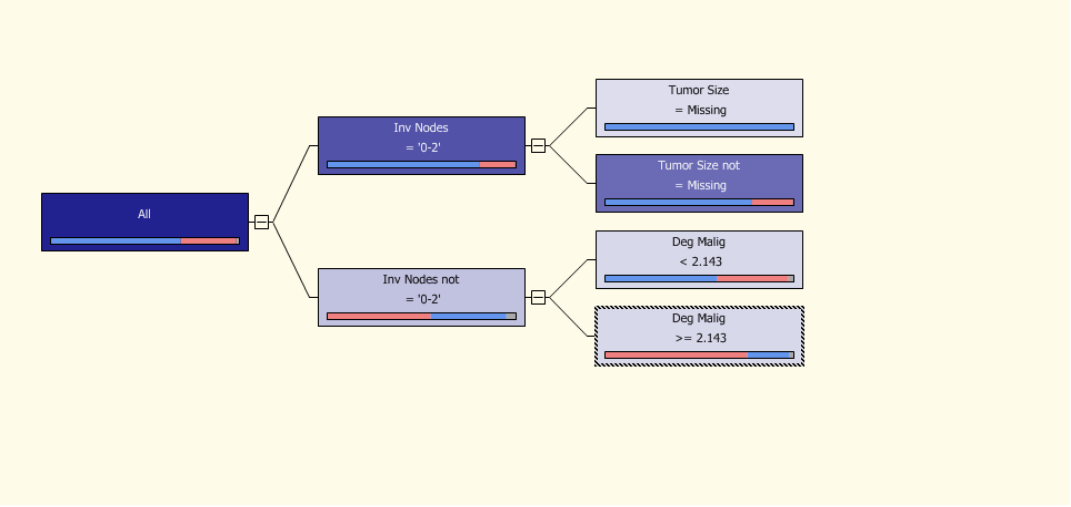


Figure 15 Decision Tree for Training Data

Predicted	No-recurrence-events (Actual)	Recurrence-events (Actual)
No-recurrence-events	36	16
Recurrence-events	4	1

Table 1 Mining Model Accuracy Chart

Figure 16 depicts the cluster map after the EM clustering algorithm was applied to the training data. Because there is currently no tool that does this type of analysis, validation against the test data set had to be done manually. Table 2 contains the combined set of rules generated by each cluster's decision tree. Table 3 contains the model's accuracy chart for the clustering and predictive model ensemble method.

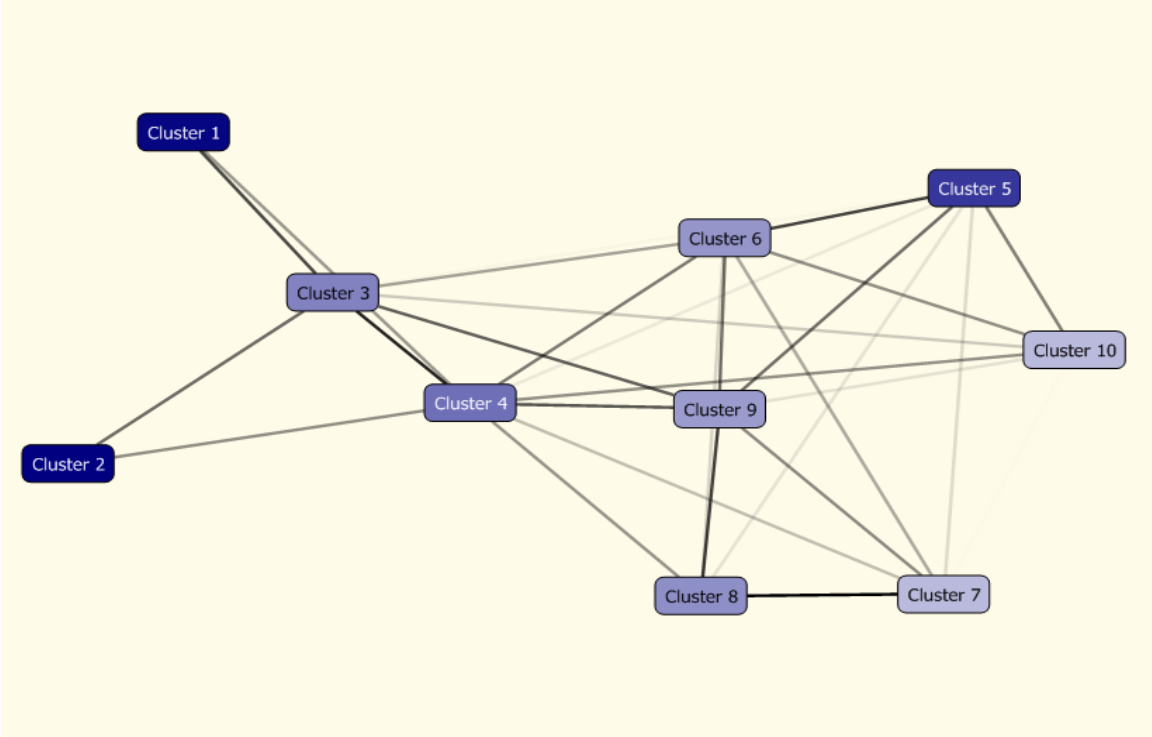


Figure 16 Cluster Map for Training Data

Rule Number	Cluster	Rule	Classification
1	1	Breast = left	No-recurrence-events
2	1	Breast = right	No-recurrence-events
3	2	Menopause = premeno	No-recurrence-events
4	2	Menopause != premeno	Recurrence-events
5	3	NodeCaps = no	No-recurrence-events
6	3	NodeCaps != no	No-recurrence-events
0	4	No tree generated	No-recurrence-events
7	5	InvNodes = 0-2	No-recurrence-events
8	5	InvNodes != 0-2	No-recurrence-events
9	6	TumorSize = 30-34	Recurrence-events
10	6	TumorSize != 30-34 and BreastQuad = leftUp	No-recurrence-events
11	6	TumorSize != 30-34 and BreastQuad != leftUp	Recurrence-events
0	7	No tree generated	No-recurrence-events
12	8	DegMalig = 3	Recurrence-events
13	8	DegMalig != 3	Recurrence-events
0	9	No tree generated	No-recurrence-events
0	10	No tree generated	Recurrence-events

Table 2 Combined rules from the decision trees built for each cluster

Predicted	No-recurrence-events (Actual)	Recurrence-events (Actual)
No-recurrence-events	40	0
Recurrence-events	0	17

Table 3 Clustering Predictive Modeling Mining Model Accuracy Chart

As shown in the table above, the clustering predictive modeling ensemble method generated a set of rules that could correctly classify every instance in the test data set (100% accuracy). Table 4 shows the ID for each test instance with its known classification and the manual classification and the rule which correctly classified it. These rules are highlighted in Table 2.

ID	Classification	Ensemble Method Classification	Rule Number
1	Recurrence-events	Recurrence-events	12
2	Recurrence-events	Recurrence-events	12
3	Recurrence-events	Recurrence-events	12
4	Recurrence-events	Recurrence-events	12
5	Recurrence-events	Recurrence-events	12
6	Recurrence-events	Recurrence-events	12
7	Recurrence-events	Recurrence-events	12
8	Recurrence-events	Recurrence-events	12
9	Recurrence-events	Recurrence-events	12
10	Recurrence-events	Recurrence-events	12
11	Recurrence-events	Recurrence-events	12
12	Recurrence-events	Recurrence-events	12
13	Recurrence-events	Recurrence-events	12
14	Recurrence-events	Recurrence-events	12
15	Recurrence-events	Recurrence-events	12
16	Recurrence-events	Recurrence-events	12
17	Recurrence-events	Recurrence-events	12
18	No-recurrence-events	No-recurrence-events	1
19	No-recurrence-events	No-recurrence-events	2
20	No-recurrence-events	No-recurrence-events	1
21	No-recurrence-events	No-recurrence-events	1

22	No-recurrence-events	No-recurrence-events	1
23	No-recurrence-events	No-recurrence-events	2
24	No-recurrence-events	No-recurrence-events	2
25	No-recurrence-events	No-recurrence-events	1
26	No-recurrence-events	No-recurrence-events	2
27	No-recurrence-events	No-recurrence-events	2
28	No-recurrence-events	No-recurrence-events	2
29	No-recurrence-events	No-recurrence-events	1
30	No-recurrence-events	No-recurrence-events	1
31	No-recurrence-events	No-recurrence-events	2
32	No-recurrence-events	No-recurrence-events	1
33	No-recurrence-events	No-recurrence-events	1
34	No-recurrence-events	No-recurrence-events	1
35	No-recurrence-events	No-recurrence-events	1
36	No-recurrence-events	No-recurrence-events	2
37	No-recurrence-events	No-recurrence-events	2
38	No-recurrence-events	No-recurrence-events	2
39	No-recurrence-events	No-recurrence-events	1
40	No-recurrence-events	No-recurrence-events	1
41	No-recurrence-events	No-recurrence-events	1
42	No-recurrence-events	No-recurrence-events	2
43	No-recurrence-events	No-recurrence-events	1
44	No-recurrence-events	No-recurrence-events	1
45	No-recurrence-events	No-recurrence-events	2
46	No-recurrence-events	No-recurrence-events	2
47	No-recurrence-events	No-recurrence-events	2
48	No-recurrence-events	No-recurrence-events	2
49	No-recurrence-events	No-recurrence-events	1
50	No-recurrence-events	No-recurrence-events	1
51	No-recurrence-events	No-recurrence-events	1
52	No-recurrence-events	No-recurrence-events	1
53	No-recurrence-events	No-recurrence-events	1
54	No-recurrence-events	No-recurrence-events	1
55	No-recurrence-events	No-recurrence-events	1
56	No-recurrence-events	No-recurrence-events	1
57	No-recurrence-events	No-recurrence-events	1

Table 4 Test instances and classifications

4. CONCLUSIONS

4.1 CONCLUSIONS

Data mining is a powerful and useful tool that can be used to extract information from large data repositories. These techniques are used in a variety of domains to perform an array of different tasks. Predictive modeling is one such task. Accuracy in predictive modeling is very important. Researchers have determined that applying an ensemble approach to predictive modeling will produce higher accuracy models. However, these techniques focus on classification algorithms. The goal of this experiment was to investigate the idea of using a clustering and predictive modeling ensemble approach. The results suggest that the clustering and predictive modeling approach can possibly produce a model that has a higher accuracy versus a predictive model alone.

4.2 *FUTURE WORK*

This research focused on the idea of using a clustering and predictive modeling approach. The dataset used in this experiment was very small. We would like to do more experiments with data sets varying in size and domain. Also, there is currently no tool available to do this type of analysis. All of the analysis for the clustering and predictive modeling portion of this experiment had to be done manually.

REFERENCES

1. TechTarget. (2008). *What is data mining?* Retrieved January 29, 2008, from http://searchsqlserver.techtarget.com/sDefinition/0,,sid87_gci211901,00.html

2. Tan, P, Steinbach, M, & Kumar, V (2006). *Introduction to Data Mining*. Boston, MA: Addison Wesley.

3. Witten, I. & Frank E. (2005). *Data Mining. Practical Machine Learning Tools and Techniques.* San Francisco, CA: Morgan Kaufmann.

4. Thearling, K. (2008). Overview *of Data Mining Techniques*. Retrieved February 13, 2008, from http://www.thearling.com/text/dmtechniques/dmtechniques.htm

5. Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). *Classification and Regression Trees.* Belmont, CA: Wadsworth International Group.

6. Larose, D. (2005). Discovering Knowledge in Data: An Introduction to Data Mining. Hoboken, NJ: John Wiley & Sons.

7. Lewis, R. (2000). *Introduction to Classification and Regression Tree (CART) Analysis*. San Francisco, CA: Annual Meeting of the Society for Academic Emergency Medicine.

8. Wikipedia. (2007). *CHAID*. Retrieved February 15, 2008, from http://en.wikipedia.org/wiki/CHAID

9. CRISP-DM. (2000). *Cross Industry Standard Process for Data Mining*. Retrieved February 16, 2008, from http://www.crisp-dm.org/Process/index.htm

10. Friedman, N., Goldszmidt, M. (1996) *Building classifiers using Bayesian networks*. National Conference on Artificial Intelligence, Menlo Park, CA, AAAI Press 1277—1284.

11. Langley, P. & Sage. S. (1994). *Induction of Selective Bayesian Classifiers.* 10th Conference on Uncertainty in Artificial Intelligence, Seattle, WA, Morgan Kaufman 399 – 406.

12. Friedman, N., Geiger, D. & Goldszmidt, M.(1997). *Bayesian Network Classifiers*. Machine Learning, 29, 131 – 163

13. Berkhin, P. (2002). *Survey of Clustering Data Mining Techniques*. Technical Report, Accrue Software, San Jose, CA.

14. Wikipedia. (2008). *Cluster Analysis.* Retrieved March 6, 2008, from http://en.wikipedia.org/wiki/Data_clustering

15. Thacker, P. (2005). *Cluster Analysis.* Retrieved March 6, 2008, from http://csurs1.csr.uky.edu/~pthacker/cluster.html.

16. Opitz, D., & Macline, R. (1999). *Popular ensemble methods: An empirical study*. Artificial Intelligence Research, 11, 169 – 198.

17. Quinlan, J.R. (2006). *Bagging, Boosting, and C4.5*. Retrieved January 29, 2008, from http://www.rulequest.com/Personal/q.aaai96.ps

18. Freund, Y. and Schapire, R. E. (1999). *A short introduction to boosting*. Japanese Society for Artificial Intelligence, 14(5), 771 – 780.

19. Berk, R. (2004). *An Introduction to Ensemble Methods for Data Analysis*. California Center for Population Research. On-Line Working Paper Series.

20. Blake, C. F & Merz, C. (1998). UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science.

21. Microsoft Corporation. (2008). *Microsoft SQL Server 2005.* Retrieved March 15, 2008, from http://www.microsoft.com/sql/prodinfo/overview/default.mspx

22. Wikipedia. (2008). *Weka (machine learning).* Retrieved March 15, 2008, from http://en.wikipedia.org/wiki/Weka_(machine_learning).

23. Oracle. (2007). *Oracle Data Mining*. Retrieved March 15, 2008, from http://www.oracle.com/technology/products/bi/odm/index.html

24. DMS. (2008). *Decision Trees.* Retrieved March 16, 2008, from http://dms.irb.hr/tutorial/tut_dtrees.php

25. Wikipedia. (2008). *Apriori Algorithm*. Retrieved March 16, 2008, from http://en.wikipedia.org/wiki/Apriori_algorithm

26. InformIT. (2008). Expectation Maximization Theory. Retrieved March 17, 2008, from http://www.informit.com/articles/article.aspx?p=363730

27. Rapid-i. (2008). *Rapid Miner.* Retrieved March 16, 2008, from

http://rapid-i.com/

28. Wikipedia. (2008). Chess endgame. Retrieved April 3, 2008, from http://en.wikipedia.org/wiki/Endgame