Chapter 2 Related Work

2.1 VPRS Model for Data Mining

The VPRS theory has been developed for data mining in databases and experimental data sets. This theory provides a powerful foundation to reveal and discover important structures in data and to classify complex objects.

An attributes-oriented VPRS technique reduces the computational complexity of learning processes and eliminates the unimportant or irrelevant attributes so that the data mining in databases or in experimental data sets can be efficiently learned. The VPRS model was introduced by Ziarko (1993) to provide a systematic framework for studying imprecise and insufficient knowledge. Using VPRS model has been shown to be very effective for revealing relationships within imprecise data, discovery dependencies among objects and attributes, evaluating the classificatory importance of attributes, removing data redundancies (and thus reducing the size of information systems), and generating decision rules.

The idea of VPRS model is based on equivalence relations which partition a data set into equivalence classes, and consists of the approximation of a set by a pair of sets, called β-lower and β-upper approximations. The β-lower approximation of a set of objects contains all objects that based on the knowledge of a given sets of attributes, can be classified as certainly belonging to the concept. The β-upper approximation of a set contains all objects that cannot be classified categorically as not belonging to the concept (Cios, et al. 1998).

2.1.1 Discretization of Real Value Attributes

Deriving classification rules is an important task in data mining. As such, discretization is an effective technique in dealing with continuous attributes for rule generating. Many classification algorithms require that the training data contain only discrete attributes, and some would work better on discretized or binarized data (Li et al. 2002; Kerber, 1992). However, for these algorithms, discretizing continuous attributes is a first step for deriving classification rules. The Variable Precision Rough Sets (VPRS) model is one example. The VPRS model is a powerful mathematical tool for data analysis and knowledge discovery from imprecise and ambiguous data. Although the theory of VPRS has been successfully applied to diverse areas, such as corporate failure prediction, identification of low-paying workplaces, and WEB searching (Beynon et al. 2001; Beynon, 2002; Ziarko et al. 2002), it cannot conduct continuous data without discretization. Thus, this requires studies on appropriate discretization methods.

There are three different axes by which discertization methods can be classified: local versus global, supervised versus unsupervised, and static versus dynamic (Dougherty et al. 1995). Local methods, such as ID3 (interactive dichotomizer 3, Quinlan 1983), produce partitions that are applied to localized regions of the instance space. By contrast, the global discertization method uses the entire instance space to discretize. Several discretization methods, such as equal width interval and equal frequency interval methods, do not utilize instance class labels in the discretization process. These methods are called unsupervised methods. Conversely, discretization methods that utilize the class labels are referred to as supervised methods. Many discretization methods require some parameter, *m,* indicating the maximum number of intervals to produce in discretizing an attribute. Static methods, such as entropy-based partitioning, perform one discretization pass of the data for each

attribute and determine the value of m for each attribute independent of the other attributes. Dynamic methods conduct a search through the space of possible m values for all attributes simultaneously, thereby capturing interdependencies in attribute discretization.

A number of methods based on entropy measure established the strong group of works in the discretization domain. This concept uses class entropy as a criterion to evaluate a list of best cuts, which together with the attribute domain induce the desired intervals (Nguyen, 1998).

Holte (1993) proposed a one-level decision tree algorithm, called 1RD (One Rule Discretizer) that attempts to greedily divide the attribute range into a number of intervals, using constraint that each interval must include at least the user-specified minimum number of values. It starts with initial partition into intervals, each containing the minimum number of values, and then moves the initial partition boundary (cut), by adding the attribute values, so that each interval contains a strong majority of objects from one decision class.

Nguyen et al. (1997) proposed an approach deals with discretization problem, which based on rough set and Boolean reasoning and the computational complexity is $O(n^3k)$, where n is the number of objects and k is the number of attributes. The main results states the problem of optimal discretization of real value attributes is polynomially reducible to the problem of minimal reduct finding, so it is NP-hard.

Nguyen (1997) proposed a general genetic strategy-based algorithm of searching for optimal set of separating hyperplanes by genetic algorithm. In case of consistent decision table (which misclassification rate equals to 0) the algorithm will be continued until the hyperplants cut the space R^k into regions containing objects from one decision class only.

Nguyen (1998) considered the relationship between reduct problem in the rough

set and the problem of real value attribute discretizaition, which searching for a minimal set of cuts on attribute domains that preserves discernibility of objects respect to any chosen attributes subset of cardinality *t* (*t* denotes a parameter given by user). Such a discretization procedure assures that one can keep all reducts consisting of at least *t* attributes.

The ChiMerge algorithm introduced by Kerber (1992) is a supervised global discertization method. The user has to provide several parameters such as the significance level α, and the maximal intervals and minimal intervals during the application of this algorithm. ChiMerge requires α to be specified. Nevertheless, too big or too small a α will over-discretize or under-discretize an attribute. Liu et al. (1997) proposed a Chi2 algorithm that uses a ChiMerge algorithm as a basis, whereby the Chi2 algorithm improves the ChiMerge algorithm in that the value of α is calculated based on the training data itself.

The modified Chi2 algorithm introduced by Shen et al. (2001) can be sectioned into two phases: The first phase of the modified Chi2 algorithm can be regarded as a generalization version of ChiMerge algorithm. Instead of specifying a χ^2 threshold, the modified Chi2 algorithm provides a wrapping that automatically increments the χ^2 threshold (decreasing the significant level α). A consistency check is used as a stopping criterion to make sure that the modified Chi2 algorithm automatically determines a proper χ^2 threshold while still keeping the fidelity of the original data.

The second phase is a finer process of the first phase, beginning with the significant level α_0 determined in the first phase, where each attribute *i* is associated with a sigLvl[*i*] and takes turns for merging. A consistency check is conducted after each attribute's merging. If the inconsistency rate does not exceed the pre-defined inconsistency rate (δ), then sigLvl[*i*] is decreased for attribute *i*'s next round of merging. Otherwise, the attribue *i* will not be involved in further merging.

This process is repeated until no attribute's value can be merged.

In the modified Chi2 algorithm, inconsistency checking (InConCheck (data)$<\delta$) of the original Chi2 algorithm is replaced by the quality of approximation L_c after each step of discretization ($L_{c-discretized} \leq L_{c-original}$). This inconsistency rate is utilized as the termination criterion. The quality of approximation coined from the Rough Sets Theory is defined as follows:

$$L_c = \frac{\sum card(\underline{B}X_i)}{card(U)}, \tag{2.1}$$

where U is the set of all objects of the data set;

X can be any subset of U;

$\underline{B}X$ is the lower approximation of X in B ($B \subseteq A$); A is the set of attributes.

The card denotes set cardinality.

The merge criterion of original Chi2 algorithm does not consider the degrees of freedom, it only used the fixed degrees of freedom (the classes' number minus one). The original Chi2 algorithm merges the pair of adjacent intervals with the lowest x^2 value being the critical value. The merge criterion of modified Chi2 considers the degrees of freedom of each two adjacent intervals. When two adjacent intervals have a maximal difference in the calculated χ^2 value, the threshold should be merged first.

The rough sets approach is inspired by the notion of inadequacy of the available information to perform a complete classification of objects. That is, to perform a complete classification requires that the collected data must be fully correct or certain. However, in real-world decision making, the objects of classes often overlap, suggesting that predictor information may be incomplete.

2.1.2 β-reducts Selection

In VPRS, the precision parameter, β, can be regarded as a classification error or the proportion of correct classifications. Because the VPRS model has no formal historical background of having empirical evidence to support any particular method of β-reduct selection (Beynon, 2002), VPRS-related research studies do not focus in detail on the choice of the β value. Ziarko (1993) proposed the β value to be specified by the decision maker. Beynon (2000) proposed two methods of selecting a β-reduct without such a known β value. Beynon (2001) proposed the allowable β value range to be an interval, where the quality of classification may be known prior to determining the β value range.

The extended VPRS was introduced by Katzberg et al. (1996), which allowed asymmetric bounds l and u to be used. The VPRS models the restriction $l < 0.5$ and $u = 1 - l$ must hold. Beynon (2002) introduced the (l,u)-quality graph, which elucidates the associated level of quality of classification, based on the selected l and u values. The results in this paper, within a criteria for the effective choice of l and u values is still required.

2.2 Neural Networks

Neural networks have been successfully applied to diverse areas such as process control, automatic inspection and information retrieval (Smith, et al. 2000; Su, et al. 2000; Simpson, 1996). Neural networks possess the unique capability of learning arbitrary non-linear mappings between noisy sets of input and output patterns. A neural network approach can usually be constructed without requiring any information concerning the functional form of the relationship between the predictors and the response (Stern, 1996). It learns and extracts the process behavior from the

past operating information. Once trained, a neural network can be evaluated very quickly, and the knowledge pertaining to the relationships between the input and output is stored in the network weights. However, neural networks are difficult to interpret owing to lack of a mathematical model to express the training result.

The extraction of symbolic rules from trained neural networks can alleviate the knowledge acquisition problem and refine the initial domain knowledge. Using extracted rules, neural network users can understand what the neural networks have learned and how the neural network makes predictions.

2.2.1 Input Nodes Selection

Removing the unnecessary input nodes from the network allows easy rule extraction from the trained network and also produces higher classification accuracy. Until now, few researchers proposed related algorithms in the literature. The following equation is used for calculating the importance of an attribute (Tsukimoto, 2000):

$$X_i = \frac{\sum_{i=1}^{n} P_i^2}{n_i} \tag{2.1}$$

where X_i is ith input node value;

n_i is links number of ith input node;

P_i is weigh connected to the pth input node.

When the above value is large, the input node is important. When it is small the input node is unimportant. Judgment criteria: if $X_i <$ threshold then ith input node will be neglected.

Su et al. (2002) proposed an effective input nodes selection algorithm that was

based on the sum of the absolute multiplication values of the weights between the layers. The essence of algorithm is to compare the multiplication weights between the input-hidden layers and hidden-output layers. Only the multiplication with large absolute values are used. When the input nodes are reduced, the remaining input nodes can be used to retrain a new neural network model.

2.2.2 Pruning a Network

Pruning a network is the process of removing unnecessary links (weigh). In the literature, links with small weights can not affect the effective extraction rules (Towell, 1999). Gupta et al. (1999) sorted each input-hidden layer weights using absolute decrease, only retaining the two largest weights. However, for each hidden node in the network, most literature search for different combinations of input links whose weighted sum exceeds the bias or threshold of the current node. These links are retained and other links are deleted.

In the above two pruning methods, the decision on what connections are unnecessary considers only the links between the input-hidden layer weights, which may cause a problem. Su, et al. (2002) proposed an algorithm to solve this problem. This algorithm removes redundant connections based on the magnitudes of their weights. Connections with sufficiently small weights can be removed without affecting the classification accuracy of the network. The algorithm is illustrated as follows:

Step 1: Set up the classification accuracy rate for the network after pruning is performed.

Step 2: Let W and V represent connections with the input-hidden and hidden-output

layers, if $\sum_p |W_{ml} * V_{mp}| \leq mean$ (the outlier should be deleted), then remove W_{ml} from the network.

Step 3: For each V_{mp} in the network, if $\sum_p |V_{mp}| \leq mean$ (the outlier should be deleted), then remove V_{mp} from the network.

Step 4: If no weights satisfy the conditions in equations $\sum_p |W_{ml} * V_{mp}| \leq mean$ or $\sum_p |V_{mp}| \leq mean$, then for each W_{ml} and V_{mp} in the network, computes $W_{ml} = |W_{ml} * V_{mp}|$ and $V_{mp} = \sum_p |V_{mp}|$. Remove the weights with the smallest W_{ml} and V_{mp}.

Step 5: If the classification rate of the network falls below the setup level, then stop. Otherwise, go to step2.

2.2.3 Rules Extraction

Extracted rules can explain the classification procedure of the neural network and shed light on the relative importance of the input attributes as well as their relationship in determining a case's class affiliation. In addition to the explanation capability, extracted rules may also have the merit of predicting new cases more accurately than the neural network.

There are several rule extraction algorithms for the trained neural network. Some literature extraction processing rules use the RX algorithm (Setiono, 1997). When the number of links between a hidden node and its corresponding input nodes are below 5 in a pruned network, one can directly extract the rules. Otherwise, one must train a sub-network with this hidden node and the input nodes connected to this hidden node.

In general, the sub-network is trained and pruned to achieve a 100% accuracy rate. The extracted rules will then have the same accuracy rate as the network on the training data.

2.3 Decision Trees

A particularly efficient method for producing classifiers from data is to generate a decision tree. The decision tree representation is the most widely used logic method. There is a large number of decision tree induction algorithms described primarily in the machine learning and applied statistics literature. They are supervised learning methods that construct decision trees from a set of input-output samples.

A typical decision trees learning system adopts a top-down strategy that searches for a solution in a part of the search space. It guarantees that a simple, but not necessarily the simplest tree will be found. A decision tree consists of nodes that where attributes are tested. The outgoing branches of a node correspond to all the possible outcomes of the test at the node (Kanttradzic, 2003).

Currently developed software for decision trees are based on the ID3 algorithm. See5 (Windows 95/98/NT/2000) and its Unix counterpart C5.0 are sophisticated data mining tools for discovering patterns that delineate categories, assembling them into classifiers and using them to make predictions.

2.3.1 Construing Decision Trees

Decision trees that use univariate splits have a simple representational form, making it relatively easy for the user to understand the inferred model; at the same time, they represent a restriction on the expressiveness of the model. A well-known tree growing algorithm for generating decision tree based on univariate splits is Quinlan's ID3

(interactive dichotomizer 3) algorithm with an extended version called C4.5. Greedy search methods, which involve growing and pruning decision tree structures, are typically employed in these algorithms to explore the exponential space of possible models.

ID3 (interactive dichotomizer 3, Quinlan 1983) uses a tree representation for concepts. ID3 begins by choosing a random subset of the training instances. This subset is called the *window*. This procedure builds a decision tree that correctly classifies all instances in the window. The tree is then tested on the training instances outside of the window. If all of the instances are classified correctly, then the procedure halts. Otherwise, it adds some of the instances incorrectly classified into the window and repeats the process. This iterative strategy is empirically more efficient than considering all of the instances at once. In building a decision tree, ID3 selects the feature that minimizes the entropy function and thus best discriminates among the training instances. In practice, a statistical criterion can be applied to stop the tree from growing as long as most of the instances are classified correctly (Fu, 1994).

In building a decision tree, ID3 utilizes entropy criteria for splitting nodes. Given a node j, the splitting criteria used is $Etropy(j) = -\sum_i p_i \log p_i$, where p_i is the probability of class i within node j.

Attribute selection in ID3 and C4.5 algorithms are based on minimizing an information entropy measure applied to the examples at a node.

2.3.2 Pruning Decision Tree

Discarding one or more sub-trees and replacing them with leaves simplify a decision tree, and that is the main task in decision trees pruning. In replacing the

sub-tree with a leaf, the algorithm expects to lower the predicted error rate and increase the quality of a classification model. The basic idea of decision tree pruning is to remove parts of the tree (sub-trees) that do not contribute to the classification accuracy of unseen testing samples, producing a less complex and more comprehensible tree. There are two ways in which the recursive partitioning method can be modified (Kantardzic, 2003):

1. Deciding not to divide a set of samples any further under some conditions: The stopping criterion is usually based on some statistical tests, such as the x^2 test: If there are no significant differences in classification accuracy before and after division, then represent a current node as a leaf. The decision is made in advance, before splitting, and therefore this approach is called pre-pruning.
2. Removing retrospectively some of the tree structure using selected accuracy criteria: The decision in this process of post-pruning is made after the tree has been built.

C4.5 follows the post-pruning approach, but it uses a specific technique to estimate the predicted error rate. This method is called pessimistic pruning. For every node in a tree, the estimation of the upper confidence limit U_{cf} is computed using the statistical tables for binomial distribution. Parameter U_{cf} is a function of $|T_i|$ and E for a given node. C4.5 uses the default confidence level of 25%, and compares $U_{25\%}(|T_i|/E)$ for a given node T_i with a weighted confidence of its leaves. Weights are the total number of cases for every leaf. If the predicted error for a root node in a sub-tree is less than weighted sum of $U_{25\%}$ for the leaves, then a sub-tree will be replaced with its root node, which becomes a new leaf in a pruned tree.

2.3.3 Generating Decision Rules

Even thought the pruned trees are more compact than the originals, they can still be very complex. Large decision trees are difficult to understand because each node has a specific context established by the outcomes of tests at antecedent nodes. To make a decision tree model more readable, a path to each leaf can be transformed into an IF-THEN production rule.

The IF part consists of all tests on a path, the THEN part is a final classification. Rules in this form are called decision rules, and a collection of decision rules for all leaf nodes would classify samples exactly as the tree does. As a consequence of their tree origin, the IF parts of the rules would be mutually exclusive and exhaustive, so the order of the rules would not matter (Kantardzic, 2003).