Discrete Decision Tree Induction to Avoid Overfitting on Categorical Data

NITTAYA KERDPRASOP and KITTISAK KERDPRASOP
Data Engineering Research Unit,
School of Computer Engineering, Suranaree University of Technology,
111 University Avenue, Nakhon Ratchasima 30000
THAILAND
nittaya@sut.ac.th, kittisakThailand@gmail.com

Abstract: - A decision tree is a hierarchical structure commonly used to visualize steps in the decision making process. Decision tree induction is a data mining method to build decision tree from archival data with the intention to obtain a decision model to be used on future cases. The advantages of decision tree induction over other data mining techniques are its simple structure, ease of comprehension, and the ability to handle both numerical and categorical data. For numerical data with continuous values, the tree building algorithm simply compares the values to some constant. If the attribute has value smaller than or equal to the constant, then proceeds to the left branch; otherwise, takes the right branch. Tree branching process is much more complex on categorical data. The algorithm has to calculate the optimal branching decision based on the proportion of each individual value of categorical attribute to the target attribute. A categorical attribute with a lot of distinct values can lead to the overfitting problem. Overfitting occurs when a model is overly complex from the attempt to describe too many small samples which are the results categorical attributes with large quantities. A model that overfits the training data has poor predictive performance on unseen test data. We thus propose novel techniques based on data grouping and heuristic-based selection to deal with overfitting problem on categorical data. Our intuition is on the basis of appropriate selection of data samples to remove random error or noise before building the model. Heuristics play their role on pruning strategy during the model building phase. The implementation of our proposed method is based on the logic programming paradigm and some major functions are presented in the paper. We observe from the experimental results that our techniques work well on high dimensional categorical data in which attributes contain distinct values less than ten. For large quantities of categorical values, discretization technique is necessary.

Key-Words: - Overfitting problem, Categorical data, Data mining, Decision tree induction, Prolog language.

1 Introduction

Decision tree induction is a popular method for mining knowledge from data by means of decision tree building and then representing the end result as a classifier tree. Popularity of this method is due to the fact that mining result in a form of decision tree is interpretability, which is more concern among casual users than a sophisticated method but lacking of understandability such as support vector machine or neural network [6].

A decision tree is a hierarchical structure with each internal node containing a decision attribute, each node branch corresponding to a distinct attribute value of the decision node, and the class of decision appears at the leaf node [3]. The goal of building a decision tree is to partition data with mixing classes down the tree until each leaf node contains data instances with pure class.

When a decision tree is built, many branches may be overly expanded due to noise or random error in the training data set. Noisy data contain incorrect attribute values caused by many possible reasons, for instance, faulty data collected from instruments, human errors at data entry, errors in data transmission [1]. If noise occurs in the training data, it can lower the performance of the learning algorithm [14]. The serious effect of noise is that it can confuse the learning algorithm to produce too specific model because the algorithm tries to classify all records in the training set including noisy ones. This situation leads to the overfitting problem [4], [8], [13]. Even training data do not contain any noise, but they instead contain categorical data with excessive number of distinct values. The tree induction results also lead to the same problem because with large quantities of categorical values, the algorithm has to divide data
into a lot of small groups. The extreme example is a group of one data instance. That introduces a lot of noise into the model.

General solution to this problem is a tree pruning method to remove the least reliable branches, resulting in a simplified tree that can perform faster classification and more accurate prediction about the class of unknown data class labels [4], [8], [10].

Most decision tree learning algorithms are designed with the awareness of noisy data. The ID3 algorithm [9] uses the pre-pruning technique to avoid growing a decision tree too deep down to cover the noisy training data. Some algorithms adopt the technique of post-pruning to reduce the complexity of the learning results. Post-pruning techniques include the cost-complexity pruning, reduced error pruning, and pessimistic pruning [7], [11]. Other tree pruning methods also exist in the literature such as the method based on minimum descriptive length principle [12], and dynamic programming based mechanism [2].

A tree pruning operation, either pre-pruning or post-pruning, involves modifying a tree structure during the model building phase. Our proposed method is different from most existing mechanisms in that we deal with noisy data prior to the tree induction phase. Its loosely coupled framework is intended to save memory space during the tree building phase and to ease the future extension on dealing with streaming data. We present the framework and detail of our methodology in Section 2. The prototype of our implementation based on the logic programming paradigm is illustrated in Section 3. Efficiency of our implementation on categorical data is demonstrated in Section 4. Conclusion and discussion appear as the last section of this paper.

2 A Method for Building Decision Tree to Handle Categorical Data

Our proposed system has been named discrete-tree induction to enunciate our intention to design a decision tree induction method to handle categorical data containing numerous discrete values. The framework as shown in Figure 1 is composed of the discrete-tree component, which is the main decision tree induction part, and the testing component responsible for evaluating the accuracy of the decision tree model as well as reporting some statistics such as tree size and running time.

Categorical value handling of our discrete-tree induction method can be achieved through the selection of the representative data, instead of learning from each and every training data. These selected data are used further in the tree building phase. Training data are first clustered by clustering module to find the mean point of each data group. The data selection module then uses these mean points as a criterion to select the training data representatives. It is a set of data representatives that to be used as input of the tree induction phase.

Heuristics have to be applied as a threshold in the selection step and as a stopping criterion in the tree building phase. The algorithms of a main module as well as the clustering, data selection, and tree induction modules are presented in Figures 2-5, respectively.

---

**Figure 1.** A decision tree induction framework.

**Input:** Data D with class label

**Output:** A tree model M

**Steps:**

1. Read D and extract class label to check distinctive values K
2. Cluster D to group data into K groups
3. In each group
   3.1 Get mean attribute values
   3.2. Compute similarity of each member compared to its mean
   3.3 Compute average similarity and variance
   3.4 Set threshold $T = 2 \times \text{Variance}$
   3.5 Select only data with similarity $> T$
4. Set stopping criteria $S$ for tree building as
   \[ S = K - \log \left( \frac{\text{number of removed data} + K}{|D|} \right) \]
5. Send selected data and criteria $S$ into tree-induction module
6. Return a tree model

**Figure 2.** Discrete-tree induction main algorithm.
3 The Implementation Based on Logic Programming

We implement the discrete-tree induction method based on the logic programming paradigm using SWI-Prolog (www.swi-prolog.org). Program and data set are in the same format, that is Horn clauses. Example of data set is shown in Figure 6.

```prolog
% attribute detail
attribute(size, [small, large]).
attribute(color, [red, blue]).
attribute(shape, [circle, triangle]).
attribute(class, [positive, negative]).
% data
instance(1, class=positive, [size=small, color=red, shape=circle]).
instant(2, class=positive, [size=large, color=red, shape=circle]).
instant(3, class=negative, [size=small, color=red, shape=triangle]).
instant(4, class=negative, [size=large, color=blue, shape=circle]).
```

Figure 6. Sample data set in a Horn clause format.

Discrete-tree induction program provides two schemes of tree building: 0 and 1. Scheme 0 corresponds to ordinary ID3 style [9] without additional noise handling mechanism. Scheme 1 is a tree induction with a heuristic-based mechanism to deal with noisy and categorical data. Prolog coding of both schemes are as follows.

```
%% Main module: dt
%% =========
dt :-
    write(" Training-data file name = > "),
    read(D), % get data file
    consult(D). % data is also a prolog program
    get_time(StartTime),
    % clear all nodes and node-ID counter in the DB
    retractall(node(_, _, _)),
    retractall(counter(_)),
    % make list Attr of all predictive attribute names
    findall(A, (attribute(A, _, A \= class), Attr),
    dtree(L,Attr), % call discrete tree induction module
    get_time(FinishTime),
    Time is FinishTime-StartTime.

%----------------------
% start traditional tree-induction with ID3 algorithm

dtree(0,Attr) :- 1,
    % make a list Ins = [1,2,...,n] of all instance ID
    findall(N, instance(N,_,_), Ins),
    % create decision tree, start with the root node
    % set MinInstance in leaf nodes = 1
    % then show model as decision tree once finish
    building phase
    induce_tree(root, Ins, Attr, 1),
    print_tree_model.
```

Figure 5. Tree building algorithm.
Program running on sample data set start with a command ‘dt’ as shown in Figure 7. Users have two choices of tree induction method: conventional decision tree induction (response with 0), and a discrete-tree induction with facilities to handle categorical data (response with 1).

4 Experimentation and Results
To test the accuracy of the proposed discrete-tree induction system, we use the standard UCI data repository [5] including the Wisconsin breast cancer, SPECT heart, DNA splice-junction, and audiology data sets. Each data set is composed of two separate subsets of training and test data. We then run the discrete-tree program and observe the results comparing to other learning algorithms, namely C4.5, Naive Bayes, k-Nearest Neighbor, and support vector machine. The comparison results are graphically shown in Figure 8.

It can be noticed from the results that the discrete-tree induction method shows considerably accurate prediction on SPECT heart data set. On Wisconsin breast cancer and DNA splice-junction data sets, our algorithm is as good as the other learning algorithms. But the discrete-tree induction performs poorly on audiology dataset. The poor performance may be due to the fact that such data set contains a single value on many attributes causing our data selection scheme making a poor set of samples.

For a SPECT heart data set, we provide tree model obtained from our algorithm to compare against the model obtained from the C4.5 algorithm. The two models are presented in Figure 9.
5 Conclusion

Categorical data can cause serious problems for many tree learning algorithms in terms of distorted results and the decrease in predicting performance of the learning results. In this paper, we propose a methodology to deal with categorical values in a decision tree induction algorithm. Our intuitive idea...
is to select only potential representatives, rather than applying the whole training data that some values are highly dispersed, to the tree induction algorithm.

Data selection process starts with clustering in order to obtain the mean point of each data group. For each data group, the heuristic \( T = 2 \times \text{Variance-of-cluster-similarity} \) will be used as a threshold to select only data around mean point within this \( T \) distance. Data that lie far away from the mean point are considered prone to noise and outliers; we thus remove them.

The removed data still play their role as one factor of a tree building stopping criterion, which can be formulated as \( S = K - \log\left(\frac{\text{number of removed data instances} + K}{|D|}\right) \), where \( K \) is the number of clusters, which has been set to be equal to the number of class labels, and \( D \) is the number of training data.

From experimental results, it turns out that our heuristic-based decision tree induction method produces a good predictive model on categorical data set. It also produces a compact tree model. With such promising results, we thus plan to improve our methodology to be incremental such that it can learn model from steaming data.

Acknowledgment

This work has been supported by grants from the National Research Council of Thailand (NRCT) and Suranaree University of Technology via the funding of Data Engineering Research Unit.

References:


