Volume 2, Issue 4, April 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Decision Tree Induction Approach for Data Classification Using Peano Count Trees

B V Chowdary, Annapurna Gummadi, UNPG Raju, B Anuradha
Computer Science and Engineering Department
Vignan Institute of Technology & Science, Hyderabad, India

Ravindra Changala
Information Technology Department
Guru Nanak Engineering College, Hyderabad, India

Abstract: ***Many organizations have large quantities of data collected in various application areas. Classification of data is a major issue which leads less efficiency and scalability. In this paper, we developed a new method for decision tree for classification of data using a data structure called Peano Count Tree (P-tree) which enhances the efficiency and scalability. We apply Data Smoothing and Attribute Relevance techniques along with a classifier. Experimental results show that the P-tree method is significantly faster than existing classification methods, making it the preferred method for mining on data to be classified.***

Key Words: Decision Tree Induction, Data Mining, Classification, Data Smoothing, Attribute Relevance Data, Peano Count Trees.

1. INTRODUCTION

Data mining is an interdisciplinary concept which holds large volumes of data**.** Classification of data is another consideration of data mining. Data Mining is an automated extraction of hidden predictive information from databases and it allows users to analyze large databases to solve business decision problems with interesting outcome. Data classification, an important task of data mining, is the process of finding the common properties among a set of objects in a database and classifies them into different classes. Decision Trees are widely used in classification [5].A decision tree method chooses an attribute, which maximizes certain and fixes the time. Then values of the attribute are split into several branches recursively, until the termination is reached. The efficiency of the existing decision tree algorithms, such as ID3 [5], C4.5 [6] and CART [3], has been established for relatively small data sets [7]. These algorithms have the restriction that the training tuples should reside in the main memory, thus limits the scalability and efficiency. The induction of decision trees from very large training sets has been previously addressed by the SLIQ [9] and SPRINT [10] decision tree algorithms. However, the data stored in databases without generalization is usually at the primitive concept level which is including continuous values for numerical attributes. Classification model.

construction process performed on huge data, so most decision tree algorithms may result in very bushy or meaningless results. In the worst case, the model cannot be constructed if the size of the data set is too large for the algorithms to handle. Hence, we address this issue by using an approach, [4] consisting of three steps: 1) Data smoothing and Attribute Relevance techniques, Classification by Decision Tree Induction Using P-trees and Classifier. The integration of these steps leads to efficient, high quality and the elegant handling of continuous noisy data. An inherent weakness of C4.5 [6] is that the information gain attribute selection criterion has a tendency to favor multi valued attributes. By creating a branch for each decision attribute value, C4.5 encounters the over-branching problem caused by unnecessary partitioning of the data.

The final decision tree constructed is efficient enough to derive the classification rules effectively. This paper is organized as follows. Section 2 describes about the Peano Count Tree Structure, Section 3 presents the Classification process and Decision Tree Construction using the proposed approach. Section 4 illustrates the use of the proposed Decision Tree method compares the results of this method with those of other classification techniques. We conclude our study in Section 5 and discuss the possible extensions based on our current work.

2. PEANO COUNT TREE STRUCTURE

A data can be viewed as different dimensions. Associated with each pixel are various descriptive attributes, called "bands". For example, visible reflectance bands (Blue, Green and Red), infrared reflectance bands (e.g., NIR,

MIR1, MIR2 and TIR) and possibly some bands of data gathered from ground sensors (e.g., yield quantity, yield quality, and soil attributes such as moisture and nitrate levels, etc.). All the values have been scaled to values between 0 and 255 for simplicity. The pixel coordinates in raster order constitute the key attribute. One can view such data as table in relational form where each pixel is a tuple and each band is an attribute. There are several formats used for data, such as Band Sequential (BSQ), Band Interleaved by Line (BIL) and Band Interleaved by Pixel (BIP). In our previous works [11], we proposed a new format called bit Sequential Organization (bSQ). Since each intensity value ranges from 0 to 255, which can be represented as a byte, we try to split each bit in one band into a separate file, called a bSQ file. Each bSQ file can be reorganized into a quadrant-based tree (P-tree). The example in Figure 1 shows a bSQ file and it's P-tree. In this example, 55 is the count of 1's in the entire image (called root count), the numbers at the next level, 16, 8, 15 and 16, are the 1-bit counts for the four major quadrants. Since the first and last quadrant is made up of entirely 1-bits (called pure-1 quadrants), we do not need sub-trees for them. Similarly, quadrants made up of entirely 0-bits are called pure-0 quadrant. This pattern is continued recursively. Recursive raster ordering is called Peano or Z-ordering in the literature.

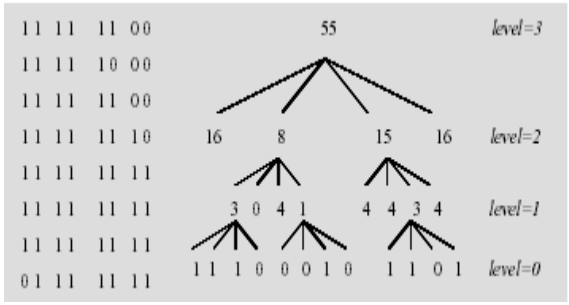

Figure 1. 8 by 8 image and its p-tree

The process terminates at the leaf level (level-0) where each quadrant is a 1- row-1-column quadrant. If we were to expand all sub-trees, including those pure quadrants, then the leaf sequence is just the Peano space-filling curve for the original raster image. For each band (assuming 8-bit data values), we get 8 basic P-trees, one for each bit positions. For band, Bi, we will label the basic P-trees, Pi, 1, Pi, 2, …, Pi,8, thus, Pi,j is a lossless representation of the jth bits of the values from the ith band. However, Pij provides more information and are structured to facilitate data mining processes. The basic P-trees defined above can be combined using simple logical operations (AND, OR and COMPLEMENT) to produce P-trees for the original values (at any level of precision, 1-bit precision, 2-bit precision, etc.). We let Pb,v denote the Peano Count Tree for band, b, and value, v, where v can be expressed in 1-bi t, 2-bit,.., or 8-bit precision. For example, Pb,110 can be constructed from the basic P-trees as: Pb,110 = Pb,1 AND Pb,2 AND Pb,3' where ' indicates the bit-complement (which is simply the count complement in each quadrant). This is called the value P-tree. The AND operation is simply the pixel wise AND of the bits. The data in the relational format can also be represented as P-trees. For any combination of values, (v1,v2,…,vn), where vi is from band-i, the quadrant-wise count of occurrences of this tuple of values is given by:

P(v1,v2,…,vn) = P1,V1 AND P2,V2 AND … AND Pn,Vn

This is called a tuple P-tree. Finally, we note that the basic P-trees can be generated quickly and it is only a one-time cost. The logical operations are also very fast [12]. So this structure can be viewed as a "data mining ready" and lossless format for storing data.

3. THE CLASSIFICATION PROCESS

Classification is a data mining technique that typically involves three phases, a learning phase, a testing phase and an application phase. A learning model or classifier is built during the learning phase. It may be in the form of classification rules, a decision tree, or a mathematical formula. Since the class label of each training sample is provided, this approach is known as supervised Learning. In unsupervised learning (clustering), the class labels are not known in advance. In the testing phase test data are used to assess the accuracy of classifier. If the classifier passes the test phase, it is used for the classification of new, unclassified data tuples. This is the application phase. The classifier predicts the class label for these new data samples. In this paper, we consider the classification of spatial data in which the resulting classifier is a decision tree (decision tree induction). Our contributions include. A set of classification-ready data structures called Peano Count trees, which are compact, rich in information and facilitate classification; A data structure for organizing the inputs to decision tree induction, the Peano count cube; A fast decision tree induction algorithm, which employs these structures. We point out the classifier is precisely the classifier built by the ID3 decision tree induction algorithm [4]. The point of the work is to reduce the time it takes to build and rebuild the classifier as new data continue to arrive. This is very important for performing classification on data.

3.1 Data Smoothing and Attribute Relevance

In the overall classification effort, as in most data mining approaches, there is a data preparation stage in which the data are prepared for classification. Data preparation can involve data cleaning (noise reduction by applying smoothing techniques and missing value management techniques). The P-tree data structure facilitates a proximity-based data smoothing method, which can reduce the data classification time considerably. The smoothing method is called bottom-up purity shifting. By replacing 3 counts with 4 and 1 counts with 0 at level-1 (and making resultant changes on up the tree), the data is smoothed and the P-tree is compressed. A more drastic smoothing can be effected. The user can determine which set of counts to replace with pure-1 and which set of counts to replace with pure-0. The most important thing to note is that this smoothing can be done almost instantaneously once P-trees are constructed. With this method it is feasible to actually smooth data from the data stream before mining. Another important pre-classification step is relevance analysis (selecting only a subset of the feature attributes, so as to improve algorithm efficiency). This step can involve removal of irrelevant attributes or redundant attributes. We can build a cube, called Peano Cube (P-cube) in which each dimension is a band and each band has several values depending on the bit precision. For example, for an image with three bands using 1-bit precision, the cell (0,0,1) gives the count of P1' AND P2 ' AND P3 . We can determine relevance by rolling-up the P-cube to the class label attribute and each other potential decision attribute in turn. If any of these roll-ups produce counts that are uniformly distributed, then that attribute is not going to be effective in classifying the class label attribute.

3.2 Classification by Decision Tree Induction Using P-trees

A Decision Tree is a flowchart-like structure in which each node denotes a test on an attribute. Each branch represents an outcome of the test and the leaf nodes represent classes or class distributions. Unknown samples can be classified by testing attributes against the tree. The path traced from root to leaf holds the class prediction for that sample. The basic algorithm for inducing a decision tree from the learning or training sample set is as follows [2, 7]: . Initially the decision tree is a single node representing the entire training set. . If all samples are in the same class, this node becomes a leaf and is labeled with that class label. . Otherwise, an entropy-based measure, "information gain", is used as a heuristic for selecting the attribute which best separates the samples into individual classes (the "decision" attribute). . A branch is created for each value of the test attribute and samples are partitioned accordingly.

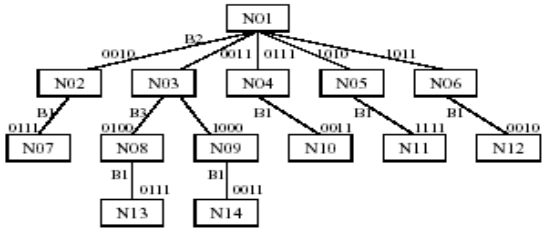

Figure2.A Decision Tree Example

The algorithm advances recursively to form the decision tree for the sub-sample set at each partition. Once an attribute has been used, it is not considered in descendent nodes. . The algorithm stops when all samples for a given node belong to the same class or when there are no remaining attributes (or some other stopping condition). The attribute selected at each decision tree level is the one with the highest information gain. The information gain of an attribute is computed by using the following

We can calculate node N's information I(P) through Where

$$I(P) = -\sum_{i=1}^{n} p_i * \log_2 p_i$$

$$p_i = \mathrm{RC}(\mathrm{P}\text{^}\mathrm{P}_{\mathrm{B}[0],\, \mathrm{v}[i]})/\mathrm{RC}(\mathrm{P}).$$

Here V0[1], ... , V0[n] are possible B[0] values if classified by B[0] at node N. If N is the root node, then P

Figure 3. Learning Dataset

FIELD COORDS		CLASS LABEL	REMOTELY SENSED REFLECTANCES		
X	Y	B1	B2	B3	B4
0	0	0011	0111	1000	1011
0	1	0011	0011	1000	1111
0	2	0111	0011	0100	1011
0	3	0111	0010	0101	1011
1	0	0011	0111	1000	1011
1	1	0011	0011	1000	1011
1	2	0111	0011	0100	1011
1	3	0111	0010	0101	1011
2	0	0010	1011	1000	1111
2	1	0010	1011	1000	1111
2	2	1010	1010	0100	1011
2	3	1111	1010	0100	1011
3	0	0010	1011	1000	1111
3	1	1010	1011	1000	1111
3	2	1111	1010	0100	1011
3	3	1111	1010	0100	1011

algorithm. Assume B[0] is the class attribute; the others are non-class attributes. We store the decision path for each node. For example, in the decision tree below (Figure 2), the decision path for node N09 is "Band2, value 0011, Band3, value 1000". We use RC to denote the root count of a P-tree, given node N's decision path B[1], V[1], B[2], V[2], … , B[t], V[t], let P-tree P=PB[1],v[1]^PB[2],v[2]^…^PB[t],v[t]

pi = RC(P^PB[0], V0[i])/RC(P).

is the full P-tree (root count is the total number of transactions). Now if we want to evaluate the information gain of attribute A at node N, we can use the formula:

$$\text{Gain}(A)=I(P)-E(A), \text{ where entropy}$$

$$E(A)=\sum_{i=1}^{n} I(P \wedge P_{A,V[i]}) * RC(P \wedge P_{A,V[i]}) / RC(P)$$

Here VA[1], ... ,VA[n] are possible A values if classified by attribute A at node N.

3.3 Example

In this example the data is a remotely sensed image (e.g., satellite image or aerial photo) of an agricultural field and the soil moisture levels for the field, measured at the same time. We use the whole data set for mining so as to get as better accuracy as we can. This data are divided into learning and test data sets. The goal is to classify the data using soil moisture as the class label attribute and then to use the resulting classifier to predict the soil moisture levels for future time (e.g., to determine capacity to buffer flooding or to schedule crop planting). Branches are created for each value of the selected attribute and subsets are partitioned accordingly. The following training set contains 4 bands of 4-bit data values (expressed in decimal and binary). B1 stands for soil-moisture. B2, B3, and B4 stand for the channel 3, 4, and 5 of AVHRR, respectively.

4. RESULT ANALYSIS

Prediction accuracy is usually used as a basis of comparison for different classification methods. However, for data mining on data, efficiency and scalability is a significant issue. In this paper, we use the ID3 algorithm with the P-tree data structure to improve them.

The important performance issue in this paper is computation speed relative to ID3. In our method, we only build and store basic P-trees. All the AND operations are performed on the fly and only the corresponding root counts are needed. Our experimental results show that larger data size leads to more significant speed improvement (in Figure 4) by using P-trees. There are several reasons. First, let's look at the cost to calculate information gain each time. In ID3, to test if all the samples are in the same class, one scan on the entire sample set is needed. While using P-trees, we only need to calculate the root counts of the AND of relevant P-trees. These AND operations can be performed very fast. Figure 5 gives the experimental results by comparing the cost of scanning the entire dataset (for different sizes) and all the P-tree ANDings. Second, Using P-trees, the creation of sub-sample sets is not necessary.

Algorithm / Parameters	C ART	ID3& C4.5	SLIQ& SPRINT	PROPOSED APPROACH
Measure	Gini Diversity index	Entropy Info-Gain	Gini Index	Info Gain& Uncertainty Coefficient
Procedure	Constructs Binary Decision Tree	Top Down Decision Tree Construction	Decision Tree Construction in a Breadth first manner	Decision Tree with Concepts of node Merging and Peano Count trees
Pruning	Post pruning based on cost-complexity measure	Pre-pruning using a single pass algorithm	Post pruning based on MDL principle	Dynamic pruning based on thresholds

5. CONCLUSION

In this paper, we propose a new approach to decision tree induction that is especially useful for the classification of data. We use the Peano Count tree (P-tree) structure to represent the information needed for classification in an efficient and ready-to-use form. The rich and efficient P-tree storage structure and fast P-tree algebra facilitate the development of a fast decision tree induction classifier. The P-tree based decision tree induction classifier is shown to improve efficiency and scalability significantly.

7. REFERENCES

[1] J. R. Quinlan and R. L. Riverst, "Inferring decision trees using the minimum description length principle", Information and Computation, 80, 227-248, 1989.
[2] Quinlan, J. R., "C4.5: Programs for Machine Learning", Morgan Kaufmann, 1993.
[3] R. Agrawal, S. Ghosh, T. Imielinski, B. Iyer, and A. Swami. "An interval classifier for database mining applications", VLDB 1992.
[4] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classfication and Regression Trees", Wadsworth, Belmont, 1984.
[5] J. Shafer, R. Agrawal, and M. Mehta, "SPRINT: A scalable parallel classifier for data mining", VLDB 96.
[6] S. M. Weiss and C. A. Kulikowski, "Computer Systems that Learn: Classification and Prediction Methods from Satatistics, Neural Nets, Machine Learning, and Expert Systems", Morgan Kaufman,

1991.[7] Jiawei Han, Micheline Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufmann, 2001.
[8] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, "Machine Learning, Neural and Statistical Classification", Ellis Horwood, 1994.
[9] Domingos, P. and Hulten, G., "Mining high-speed data streams", Proceedings of ACM SIGKDD 2000.
[10]Domingos, P., & Hulten, G., "Catching Up with the Data:Research Issues in Mining Data Streams", DMKD 2001.
[11]William Perrizo, Qin Ding, Qiang Ding, Amlendu Roy, "Deriving High Confidence Rules from Spatial Data using Peano Count Trees", Springer-Verlag, LNCS 2118, July 2001.
[12]William Perrizo, "Peano Count Tree Technology", Technical Report NDSU-CSOR-TR-01-1, 2001.

