INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY STUDIES & SPPP's, Karmayogi Engineering College, Pandharpur Organize National Conference
Special Issue March 2016
Vol. 2, Special Issue 1, March, 2016 | ISSN (Online): 2454-8499 | Impact Factor: 1.3599(GIF), 0.679(IIFS)

Decision Tree Classification of Spatial Data Streams Using Peano Trees of classification

Swapna V. Tikore[1], Dhotre Virendrakumar A.[2],
[1]Department of Computer Science and Engineering, RIT, Paniv
[2]Department of Computer Science and Engineering, Fabtech Technical Campus. Sangola

Many organizations have large quantities of spatial data collected in various application areas, including remote sensing, geographical information systems (GIS), astronomy, computer cartography, environmental assessment and planning, etc. These data collections are growing rapidly and can therefore be considered as spatial data streams. For data stream classification, time is a major issue. However, these spatial data sets are too large to be classified effectively in a reasonable amount of time using existing methods. In this paper, we developed a new method for decision tree classification on spatial data streams using a data structure called Peano Count Tree (P-tree). The Peano Count Tree is a spatial data organization that provides a lossless compressed representation of a spatial data set and facilitates efficient classification and other data mining techniques. Using P-tree structure, fast calculation of measurements, such as information gain, can be achieved. We compare P-tree based decision tree induction classification and a classical decision tree induction method with respect to the speed at which the classifier can be built (and rebuilt when substantial amounts of new data arrive). Experimental results show that the P-tree method is significantly faster than existing classification methods, making it the preferred method for mining on spatial data streams.

Index Terms—Big Data; Data mining, Classification, Decision Tree Induction, Spatial Data, Data Streams, Twitter .

I. INTRODUCTION

In many areas, large quantities of data are generated and collected everyday, such as supermarket transactions, phone call records. These data arrive too fast to be analyzed or mined in time. Such kinds of data are called “data streams” [9, 10]. Classifying open-ended data streams brings challenges and opportunities since traditional techniques often cannot complete the work as quickly as the data is arriving in the stream [9, 10]. Spatial data collected from sensor platforms in space, from airplanes or other platforms are typically updated periodically. For example, AVHRR (Advanced Very High Resolution Radiometer) data is updated every hour or so (8 times each day during daylight hours). Such data sets can be very large (multiple gigabytes) and are often archived in deep storage before valuable information can be obtained from them. An objective of spatial data stream mining is to mine such data in near real time prior to deep storage archiving.

Classification is one of the important areas of data mining [6,7,8]. In classification task, a training set (or called learning set) is identified for the construction of a classifier. Each record in the learning set has several attributes, one of which, the goal or class label attribute, indicates the class to which each record belongs. The classifier, once built and tested, is used to predict the class label of new records that do not yet have a class label attribute value.

A test set is used to test the accuracy of the classifier. The classifier, once certified, is used to predict the class label of future unclassified data. Different models have been proposed for classification, such as decision trees, neural networks, Bayesian belief networks, fuzzy sets, and generic models. Among these models, decision trees are widely used for classification. We focus on decision tree induction in this paper. ID3 (and

its variants such as C4.5) [1, 2] and CART [4] are among the best known classifiers that use decision trees. Other decision tree classifiers include Interval Classifier [3] and SPRINT [3, 5] which concentrate on making it possible to mine databases that do not fit in main memory by only requiring sequential scans of the data. Classification has been applied in many fields, such as retail target marketing, customer retention, fraud detection and medical diagnosis [8]. Spatial data is a promising area for classification. In this paper, we propose a decision tree based model to perform classification on spatial data streams. We use the Peano Count Tree (P-tree) structure [11] to build the classifier.

P-trees [11] represent spatial data bit-by-bit in a recursive quadrant-by-quadrant arrangement. With the information in P-trees, we can rapidly build the decision tree. Each new component in a spatial data stream is converted to P-trees and then added to the training set as soon as possible. Typically, a window of data components from the stream is used to build (or rebuild) the classifier. There are many ways to define the window, depending on the data and application. In this paper, we focus on a fast classifier-building algorithm.

The rest of the paper is organized as follows. In section 2, we briefly review the spatial data formats and the P-tree structure. In Section 3, we detail our decision tree induction classifier using P-trees. We also walk through an example to illustrate our approach. Performance analysis is given in Section 4. Finally, there is a conclusion in Section 5.Authors of regular and conference-related papers should prepare their papers for review using Microsoft Word and this template or LaTeX and the files contained in IEEEtran.zip. If your paper is submitted in conjunction with a conference, please observe any page limits specified by the conference

II. PIANO TREE STRUCTURE

A spatial image can be viewed as a 2-dimensional array of pixels. 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 spatial 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 its P-tree.

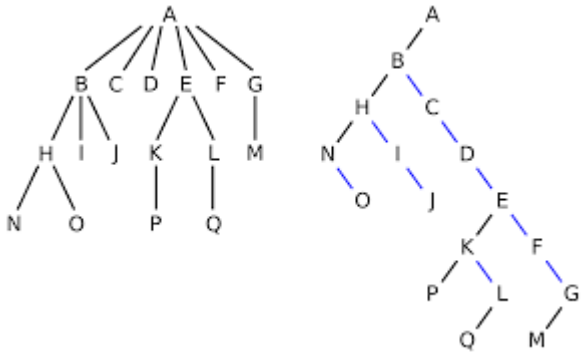

Fig: 8 by 8 image and its p-tree

This pattern is continued recursively. Recursive raster ordering is called Peano or Z-ordering in the literature. 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. Some of the useful features of P-trees can be found later in this paper or our earlier work [11].

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-bit, 2-bit,.., or 8-bit precision. For example, Pb,110 can be constructed from the basic P-trees as:

$$P_{b,110} = P_{b,1} \text{ AND } P_{b,2} \text{ AND } P_{b,3'}$$

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:

$$P(v1,v2,\ldots,vn) = P_{1,V1} \text{ AND } P_{2,V2} \text{ AND } \ldots \text{ AND } P_{n,Vn}$$

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 spatial data.

III. The Classifier

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 streams.

A. 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. The roll-up can be computed from the basic P-trees without necessitating the actual creation of the P-cube. This can be done by ANDing the P-trees of class label attribute with the P-trees of the potential decision attribute. Only an estimate of uniformity in the root counts is all that is needed. Better estimates can be discovered by ANDing down to a fixed depth of the P-trees. For instance, ANDing to depth=1 counts provides the rough set of distribution information, ANDing at depth=2 provides better distribution information and so forth. Again, the point is that P-trees facilitate simple real-time relevance analysis, which makes it feasible for data streams.

B. Data 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.
- 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 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]

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY STUDIES & SPPP's, Karmayogi Engineering College, Pandharpur Organize National Conference
Special Issue March 2016
Vol. 2, Special Issue 1, March, 2016 | ISSN (Online): 2454-8499 | Impact Factor: 1.3599(GIF), 0.679(IIFS)

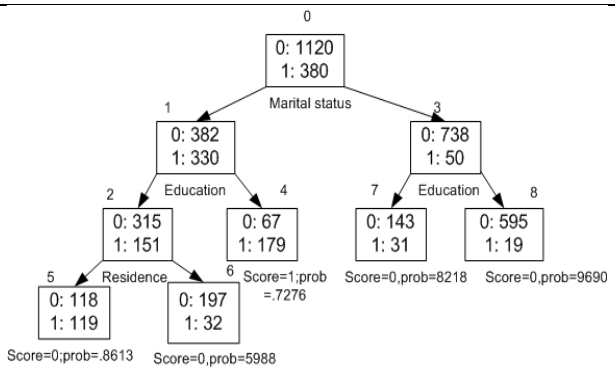

Fig: A decision tree example

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

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

where

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

Gain(A)=I(P)-E(A), where entropy

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

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

C. 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.

FIELD *CLASS* REMOTELY SENSED COORDS *LABEL* 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

Fig: Learning Dataset

This learning dataset (Figure 3) is converted to bSQ format. We display the bSQ bit-bands values in their spatial positions, rather than displaying them in 1-column files. The Band-1 bit-bands are:

B11	B12	B13	B14
0000	0011	1111	1111
0000	0011	1111	1111
0011	0001	1111	0001
0111	0011	1111	0011

Thus, the Band-1 basic P-trees are as follows (tree pointers are omitted).

P1,1	P1,2	P1,3	P1,4
5	7	16	11
0014	0403	4403	
0001	0111	0111	

We can use AND and COMPLEMENT operation to calculate all the value P-trees of Band-1 as below. (e.g., P1,0011 = P1,1' AND P1,2' AND P1,3 AND P1,4)

P1,0000	P1,0100	P1,1000	P1,1100
0	0	0	0

P1,0010 P1,0110 P1,1010 P1,1110
3 0 2 0
0030 0011
1110 0001 1000

P1,0001 P1,0101 P1,1001 P1,1101
0 0 0 0

P1,0011 P1,0111 P1,1011 P1,1111
4 4 0 3

Then we generate basic P-trees and value P-trees similarly to B2, B3 and B4.

Start with A = B2. Because the node currently dealing is the root node, P is the full P-tree. So pi can be 3/16, 1/4, 1/4, 1/8, 3/16, thus we can calculate

I(P) = 3/16*log2(3/16) + 4/16*log2(4/16) + 4/16*log2(4/16) + 2/16*log2(2/16) + 3/16*log2(3/16)) = 2.281

To calculate E(B2), first P^PA,VA[i] should be all the value P-trees of B2. Then I(P^PA,VA[i]) can be calculated by ANDing all the B2 value P-trees and B1 value P-trees. Finally we get E(B2)=0.656 and Gain(B2)=1.625.

Likewise, the Gains of B3 and B4 are computed: Gain(B3) = 1.084 , Gain(B4) = 0.568. Thus, B2 is selected as the first level decision attribute.

Branches are created for each value of B2 and samples are partitioned accordingly.

B2=0010 → Sample_Set_1

B2=0011 → Sample_Set_2

B2=0111 → Sample_Set_3

B2=1010 → Sample_Set_4

B2=1011 → Sample_Set_5

Advancing the algorithm recursively to each sub-sample set, it is unnecessary to rescan the learning set to form these sub-sample sets, since the P-trees for those samples have been computed.

The algorithm will terminate with the decision tree:

B2=0010 → B1=0111

B2=0011 → B3=0100 → B1=0111

B3=1000 → B1=0011

B2=0111 → B1=0011

B2=1010 → B1=1111

B2=1011 → B1=0010

IV. Performance Analysis

Prediction accuracy is usually used as a basis of comparison for different classification methods. However, for data mining on streams, speed is a significant issue. In this paper, we use the ID3 algorithm with the P-tree data structure to improve the speed. 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. If A is a candidate for the current decision attribute with kA basic P-trees, we only need to AND the P-trees of the class label defining the sub-sample set with each of the kA basic P-trees. If the P-tree of the current sample set is P2, 0100 ^ P3, 0001, and the current attribute is B1 (with, say, 2 bit values), then P2, 0100 ^ P3, 0001 ^ P1, 00, P2, 0100 ^ P3, 0001 ^ P1, 01, P2, 0100 ^ P3, 0001 ^ P1, 10 and P2, 0100 ^ P3, 0001 ^ P1,11 identifies the partition of the current sample

set. In our algorithm, only P-tree ANDings are required.

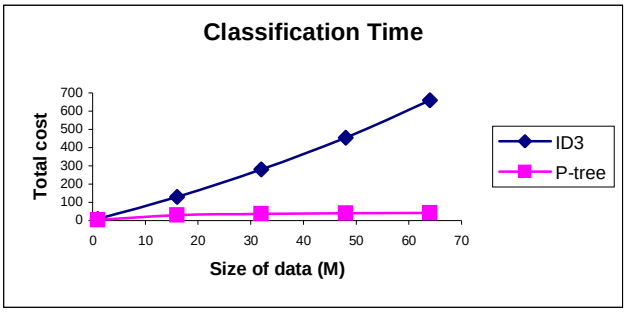

Fig: Classification cost with respect to the dataset size

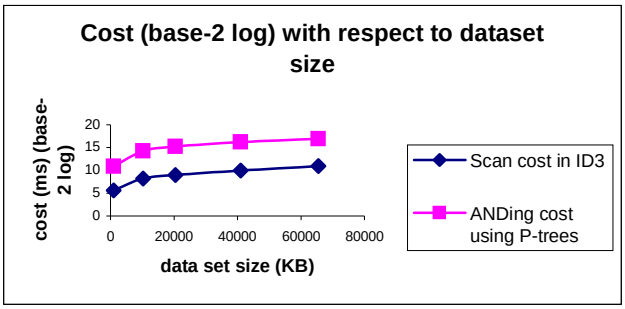

Fig: Cost Comparison between scan and ANDing

V. CONCLUSION

In this paper, we proposed a new approach to decision tree induction that is especially useful for the classification of spatial data streams. 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 classifier development time significantly. This makes classification of open-ended streaming datasets feasible in near real time.

ACKNOWLEDGMENT

We would like to express our thanks to Amalendu Roy of Motorola, William Jockheck of Northern State University and Stephen Krebsbach of Dakota State University for their help and suggestions.

REFERENCES

[1] G. Eason, 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.