A Study on Fraud Detection Based on Data Mining Using Decision Tree

[1] A.N.Pathak, [2] Manu Sehgal and [3] Divya Christopher

[1] Dr.A.N.Pathak, Professor & Head Applied Science, NITTTR
Chandigarh -160019, India

[2] Manu Sehgal, Department of Information Technology, GGDSD College.
Chandigarh -160019, India.

[3] Divya Christopher Senoir Lecturer, Amity university
Noida, India

Abstract

Fraud is a million dollar business and it is increasing every year. The U.S. identity fraud incidence rate increased in 2008 returning to levels unseen since 2003. Almost 10 million Americans learned they were victims of identity (ID) fraud in 2008 which is up from 8.1 million victims in 2007. More consumers are becoming identity (ID) fraud victims reversing the previous trend in which identity (ID) fraud had been gradually decreasing. This reverse makes sense since overall criminal activity tends to increase where there is a recession.

Fraud involves one or more persons who intentionally act secretly to deprive another of something of value, for their own benefit. Fraud is as old as humanity itself and can take an unlimited variety of different forms. However, in recent years, the development of new technologies has also provided further ways in which criminals may commit fraud (Bolton and Hand 2002). In addition to that, business reengineering, reorganization or downsizing may weaken or eliminate control, while new information systems may present additional opportunities to commit fraud.

Keywords: *Data mining, decision tree, gini impurity, information gain, leaf, binary decision diagram*

1. Introduction

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases

2. Decision tree learning, used in Data mining and machine learning, uses a decision tree as a predictive Model which maps observations about an item to conclusions about the item's target value. More descriptive names for such tree models are **classification trees** or **regression trees**. In these tree structures, leaves represent classifications and branches represent conjuctions of features that lead to those classifications.

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decsion making. In Datamining, a decision tree describes data but not decisions; rather the resulting classification tree can be an input for decsion making. This paper deals with decision trees in data mining.

Decision tree learning is a common method used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. Each interior node corresponds to one of the input variables; there

are edges to children for each of the possible values of that input variable. Each **leaf**

represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions.

In dataminig, trees can be described also as the combination of mathematical and computational techniques to aid the description, categorisation and generalisation of a given set of data.

Data comes in records of the form:

$$(\mathbf{x}, Y) = (x_1, x_2, x_3, ..., x_k, Y)$$

The dependent variable, Y, is the target variable that we are trying to understand, classify or generalise. The vector **x** is composed of the input variables, x_1, x_2, x_3 etc., that are used for that task

3. Formulae

The algorithms that are used for constructing decision trees usually work top-down by choosing a variable at each step that is the next best variable to use in splitting the set of items. "Best" is defined by how well the variable splits the set into subsets that have the same value of the target variable. Different algorithms use different formulae for measuring "best". This section presents a few of the most common formulae. These formulae are applied to each candidate subset, and the resulting values are combined (e.g., averaged) to provide a measure of the quality of the split.

3.1 Gini impurity

Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labelled if it were randomly labelled according to the distribution of labels in the subset. Used by the CART algorithm Gini impurity can be computed by summing the probability of each item being chosen times the probability of a mistake in categorizing that item. It reaches its minimum (zero) when all cases in the node fall into a single target category.

To compute Gini impurity for a set of items, suppose y takes on values in {1, 2, ..., m}, and let f_i = the fraction of items labelled with value i in the set.

$$I_G(f) = \sum_{i=1}^{m} f_i(1 - f_i) = \sum_{i=1}^{m} (f_i - f_i^2) = \sum_{i=1}^{m} f_i - \sum_{i=1}^{m} f_i^2 = 1 - \sum_{i=1}^{m} f_i^2$$

3.2 Information gain

Used by the ID3,C4.5 and C5.0 tree generation algorithms.Information gain is based on the concept of entropy used in Information theory.

$$I_E(f) = -\sum_{i=1}^{m} f_i \log_2 f_i$$

3.3 Binary decision Diagram

A Boolean function can be represented as a rooted, directed, acyclic graph, which consists of decision nodes and two terminal nodes called 0-terminal and 1-terminal. Each decision node is labeled by a Boolean variable and has two child nodes called low child and high child. The edge from a node to a low (high) child represents an assignment of the variable to 0 (1). Such a **BDD** is called 'ordered' if different variables appear in the same order on all paths from the root. A BDD is said to be 'reduced' if the following two rules have been applied to its graph:

- Merge any isomorphic subgraphs.
- Eliminate any node whose two children are isomorphic.

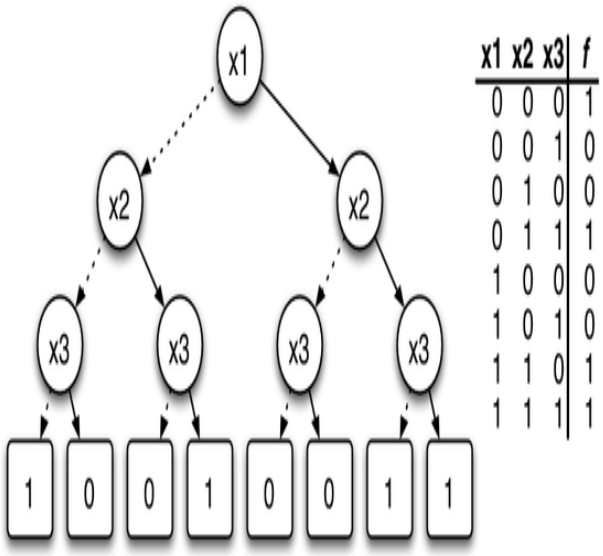
In popular usage, the term **BDD** almost always refers to **Reduced Ordered Binary Decision Diagram** (**ROBDD** in the literature, used when the ordering and reduction aspects need to be emphasized). The advantage of an ROBDD is that it is canonical (unique) for a particular function and variable order.[1] This property makes it useful in functional equivalence checking and other operations like functional technology mapping.

A path from the root node to the 1-terminal represents a (possibly partial) variable assignment for which the represented Boolean function is true. As the path descends to a low child (high child) from a node, then that node's variable is assigned to 0 (1).

3.4 Example

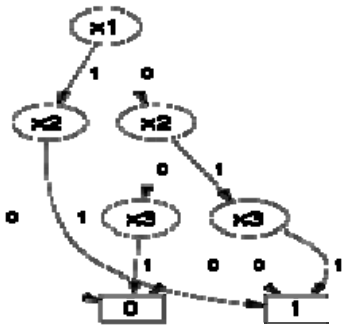
The left figure below shows a binary decision tree (the

reduction rules are not applied), and a truth table, each representing the function f (x1, x2, x3). In the tree on the left, the value of the function can be determined for a given variable assignment by following a path down the graph to a terminal. In the figures below, dotted lines represent edges to a low child, while solid lines represent edges to a high child. Therefore, to find (x1=0, x2=1, x3=1), begin at x1, traverse down the dotted line to x2 (since x1 has an assignment to 0), then down two solid lines (since x2 and x3 each have an assignment to one). This leads to the terminal 1, which is the value of f (x1=0, x2=1, x3=1).



x1	x2	x3	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

The binary decision *tree* of the left figure can be transformed into a binary decision *diagram* by maximally reducing it according to the two reduction rules. The resulting **BDD** is shown in the right figure.



4. Conclusion

This study has given a brief view of different methods on fraud detection based on decision trees.we have discussed three methods namely gini impurity,information gain and binary decision diagram which is explained with a small example.

References

1. Breiman, Leo; Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. ISBN 978-0412048418

2. Kass, G. V. (1980). "An exploratory technique for investigating large quantities of categorical data". *Applied Statistics* **29** (2): 119–127.

doi:10.2307/2986296.
http://jstor.org/stable/2986296.

3. Friedman, J. H. (1999). *Stochastic gradient boosting.* Stanford University.

4. Hastie, T., Tibshirani, R., Friedman, J. H. (2001). *The elements of statistical learning : Data mining, inference, and prediction.* New York: Springer Verlag.

5. Rokach, L.; Maimon, O. (2005). "Top-down induction of decision trees classifiers-a survey". *IEEE Transactions on Systems, Man, and Cybernetics, Part C* **35**: 476–487.

6. Hyafil, Laurent; Rivest, RL (1976). "Constructing Optimal Binary Decision Trees is NP-complete". *Information Processing Letters* **5** (1): 15–17. doi:10.1016/0020-0190(76)90095-8.

7. Murthy S. (1998). Automatic construction of decision trees from data: A multidisciplinary survey. *Data Mining and Knowledge Discovery*

8. Papagelis A., Kalles D.(2001). Breeding Decision Trees Using Evolutionary Techniques, Proceedings of the Eighteenth International Conference on Machine Learning, p.393-400, June 28-July 01, 2001

9. doi:10.1007/978-1-84628-766-4

10. Graph-Based Algorithms for Boolean Function Manipulation, Randal E. Bryant, 1986

11. C. Y. Lee. "Representation of Switching Circuits by Binary-Decision Programs". Bell Systems Technical Journal, 38:985–999, 1959.

12. Sheldon B. Akers. Binary Decision Diagrams, IEEE Transactions on Computers, C-27(6):509–516, June 1978.

13. Raymond T. Boute, "The Binary Decision Machine as a programmable controller". EUROMICRO Newsletter, Vol. 1(2):16–22, January 1976.

14. Randal E. Bryant. "Graph-Based Algorithms for Boolean Function Manipulation". IEEE Transactions on Computers, C-35(8):677–691, 1986.

15. R. E. Bryant, "Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams", ACM Computing Surveys, Vol. 24, No. 3 (September, 1992), pp. 293–318.

Dr A.N.Pathak , Professor and head Applied Scinece Department NITTTR,Chandigarh, member of professional Body : Fellow of Institution of Engineers India. Life member of Indian Society of Biotechnology. Gold medalist Institution of Engineers India.
Educational Qualification**:**M.Sc,B.Tech,M.Tech, Phd (chem-Engineering IIT Delhi) FIE Post Doctorate stuttgart University(Germany)
Gold medalist at Birbal Savitree Shahani Foundation area of Specilaization :Applied Chemistry,Biotechnoligy,Nanotechnology,IPR, Chemical engineering,Fluid Mechanics, Applied Scineces

Manu Sehgal Asssitant professor of Computer Science in Information Technology Department of GGDSD College,Chandigarh.She has done her bachelors in Computer Application and Masters in Information Technology with distinction from India. Aera of specialzation : Database Management.

Divya Christopher Senoir Lecturer in department of Biotechnology Management.She did her Bachlors in commerce and Masters in Management in India. Aera of specialzation: operations