RESEARCH ARTICLE OPEN ACCESS

Optimized Pattern Mining Of Sample Data

Yamini Chowdary Nalamothu[*], P. V. Subba Reddy[**]
[*](Computer Science and Engineering, QIS College of Engineering and Technology, Andhra Pradesh, India.)
[**](Computer Science and Engineering, QIS College of Engineering and Technology, Andhra Pradesh, India.)

ABSTRACT
Data precise is often found in real-world applications due to reasons such as imprecise measurement, outdated sources, or sampling errors. Much research has been published in the area of managing in databases. In many cases, only partially aggregated data sets are available because of privacy concerns. Thus, each aggregated record can be represented by a probability distribution. In other privacy-preserving data mining applications, the data is perturbed in order to preserve the sensitivity of attribute values. In some cases, probability density functions of the records may be available. Some recent techniques construct privacy models, such that the output of the transformation approach is friendly to the use of data mining and management techniques. Here data is inherent in applications such as sensor monitoring systems, location-based services, and biological databases. There is an increasing desire to use this technology in new application domains. One such application domain that is likely to acquire considerable significance in the near future is database mining. The information in those organizations are classified using EMV, mean obtained after calculating the profit. Decision tree is constructed using the profit and EMV values which prunes the data to the desired extent. An increasing number of organizations are creating ultra large data bases (measured in gigabytes and even terabytes) of business data, such as consumer data, transaction histories, sales records, etc. Such data forms a potential gold mine of valuable business information.

Keywords —Averaging, Classification, Database mining, Decision trees, Knowledge discovery.

I. INTRODUCTION

Classification is a classical problem in machine learning and data mining [1]. The databases relate to various aspects of their business and are information mines that they would like to exploit to improve the quality of their decision making. The maturing of database technology and the successful use of commercial database products in business data processing, the market place is showing evidence of increasing desire to use database technology in new application domains. One such application domain that is likely to acquire considerable significance in the near future is database mining. Several organizations have created ultra large data bases, running into several gigabytes and more.The database systems of today offer little functionality to support such “mining" applications. At the same time, statistical and machine learning techniques usually perform poorly when applied to very large data sets. This situation is probably the main reason why massive amounts of data are still largely unexplored and are either stored primarily in an offline store or are on the verge of being thrown away.

Database mining is confluence of machine learning technique. The performance is emphasis of database technology. We argue that a number of database mining problems can be uniformly viewed as requiring discovery of rules embedded in massive data. Database technology has been used with great success in traditional business data processing. There is an increasing desire to use this technology in new application domains. One such application domain that is likely to acquire considerable significance in the near future is database mining. The database systems of today offer little functionality to support such “mining" applications. An increasing number of organizations are creating ultra large data bases of business data, such as consumer data, transaction histories, sales records, etc. Such data forms a potential gold mine of valuable business information.

In today’s, Classification is one of the most widely and efficiently used data mining techniques. Thus data-mining can be very effective for extracting knowledge from huge amount of data. Classification is separation or ordering of objects into classes. There are various classification techniques i.e. Decision tree, K-nearest neighbor, Naive Bayes classifier, neural network. Classification is a classical problem in machine learning and data mining. In decision tree classification, an attribute of a tuple is either categorical or numerical. Decision tree mainly makes use of classification. Given a set of training data tuples, each having a class label and being

represented by a feature vector, the task is to algorithmically build a model that predicts the class label of an unseen test tuple based on the tuples feature vector. We present in this paper our perspective of database mining as the consequence of machine learning techniques and the performance emphasis of database technology. We argue that a number of database mining problems can be uniformly viewed as requiring discovery of rules embedded in massive data. We describe a model and some basic operations for the process of rule discovery [2]. The emphasis in [5] is on having a declarative language that makes it easier to formulate and revise hypotheses. The emphasis in is on providing a large bandwidth between the machine and human so that user-interest is maintained between successive iterations.

II. RELATED WORK

A data tuple in a relational database could be associated with a probability that represents the confidence of its presence. "Probabilistic databases" have been applied to semi-structured data and XML [1]. When some attribute values are not available during data collection or due to data entry errors. Solutions include approximating missing values with the majority value or inferring the missing value (either by exact or probabilistic values) using a classifier on the attribute. In probabilistic decision trees [6], missing values in training data are handled by using fractional tuples. During testing, each missing value is replaced by multiple values with probabilities based on the training tuples, thus allowing probabilistic classification results. We have also adopted the idea of probabilistic classification results.

A simple method of "filling in" the missing values could be adopted to handle the missing values, taking advantage of the capability of handling arbitrary pdf's in our approach. The advantage of our approach is that the tuple splitting is based on probability values, giving a natural interpretation to the splitting as well as the result of classification. Building a decision tree on tuples with numerical, point valued data is computationally demanding [7]. A numerical attribute usually has a possibly infinite domain of real or integral numbers, inducing a large search space for the best "split point". Given a set of n training tuples with a numerical attribute, there are as many as n -1 binary split points or ways to partition the set of tuples into two non-empty groups. Finding the best split point is thus computationally expensive. To improve efficiency, many techniques have been proposed to reduce the number of candidate split points [8], [7], [9]. These techniques utilize the convex property of well-known evaluation functions like Information Gain [2] and Gini Index [10]. For the evaluation function TSE (Training Set Error), which is convex but not strictly convex, one only needs to consider the "alternation points" as candidate split points.[11] An alternation point is a point at which the ranking of the classes (according to frequency) changes. In this paper, we consider only strictly convex evaluation functions. Compared to those works, ours can be considered an extension of their optimization techniques for handling data.

We present three classes of database mining problems involving classification, associations, and sequences. We present a unifying framework and show how these three classes of problems can be uniformly viewed as requiring discovery of rules. We introduce operations that may form the computational kernel for the process of rule discovery. We show how the database mining problems under consideration can be solved by combining these operations. To make the discussion concrete, we consider the classification problem in detail in Section 5, and present a concrete algorithm for classification problems obtained by combining these operations. We show that the classifier so obtained is not only client but has classification accuracy comparable to the well-known classifier [12].

III. PATTERN MINING

"Pattern mining" is a data mining method that involves finding existing pattern in data. The original motivation for searching association rules came from the desire to analyze supermarket transaction data, that is, to examine customer behavior in terms of the purchased products. For example, an association rules "beer ⇒ potato chips (80%)" states that four out of five customers that bought beer also bought potato chips.

In the context of pattern mining as a tool to identify terrorist activity, the National Research Council provides the following definition: "Pattern-based data mining looks for patterns (including anomalous data patterns) that might be associated with terrorist activity — these patterns might be regarded as small signals in a large ocean of noise". Pattern Mining includes new areas such a Music Information Retrieval (MIR) where patterns seen both in the temporal and non-temporal domains are imported to classical knowledge discovery search methods.

ALGORITHMS

A data tuple in a relational database could be associated with a probability that represents the confidence of its presence. "Probabilistic databases" have been applied to semi-structured data and XML.

In this section, we discuss Expected Monetary Value (EMV), profit, mean and probability of getting contract. In database the user

gave the price, cost of tender and component cost and company name. The process of calculating profit is calculated in the process of algorithm based.

Profit = Price-(cost of tender + Component Cost);

The profit values calculation is important to know the profit of user/manipulator. The data will be stored in the database. The database records maintain the MS access dbs file. The mean and EMV value is calculated by formula based. The mean value formula is,

Mean= [Probability * Profit]/ 100;

And EMV value is calculated on the process of formulas based

t = [Probability * Profit] / 100;

u = [100 – Probability]* [tcost] / 100;

EMV = t – u;

These all values calculated on the below algorithm based. The tender will be allotted to the company whose EMV value is high. The algorithm given follows:

To calculate the profit, mean, EMV.

1. Read/ Take price.
2. Read component cost, tender cost (tcost).
3. Calculate Profit.
   Profit= Price- [tcost + component cost];
4. Calculate Probability.
   Probability= [1/ Total no. of Tenders] * 100;
5. Calculate Mean.
   Mean= [Probability * Profit]/ 100;
6. Calculate EMV.
   t = [Probability * Profit] / 100;
   u = [100 – Probability]* [tcost] / 100;
   EMV = t – u;

Here the values will be calculated on above algorithm based. Where tcost is the cost of tender, and the tcost can be given by a user. The cost of tender price, tcost will be given by user and component cost & probability is retrieved from the database. Calculation of profit, mean and EMV values is selecting which one own the tender, EMV value is necessary. Here the profit and mean will be calculated by algorithm based and the EMV value calculated on the profit and mean based. After calculation of tender cost user may represent a tree classifier, on the other hand could work as a filter, if it displays good retrieval performance. This involves, in general, building a classifier that best approximates a given "black box" function and has desirable retrieval characteristics.

We discuss two approaches for handling the data. The first approach, called "Averaging", our second approach, called "Distribution-based", transforms a dataset to a point-valued one by replacing each data tuple with its mean value. A decision tree can then be built by applying a traditional tree construction. Also, a slight change of the split point modifies that probability, potentially altering the tree structure. We present details of the tree-construction algorithms under the two approaches in the following subsections.

Averaging and Distribution based approaches

A straight-forward way to deal with the information is to replace each pdf with its expected value, thus effectively converting the data tuples to point-valued tuples. This reduces the problem back to that for point-valued data, and hence traditional decision tree algorithms such as mean can be reused. We call this approach averaging. Averaging and Distribution based approaches builds a tree top-down.

IV. EXPERIMENTAL RESULTS & ANALYSIS

To explore the potential of achieving higher classification accuracy by considering data, we have implemented averaging and distribution based approaches and applied them to 7 real data sets (see Table I). For the purpose of our experiments, classifiers are built on the numerical attributes and their "class label" attributes. Some data sets are already divided into "training" and "testing" tuples.

A. Results Obtained For Averaging

A straight-forward way to deal with the information, this reduces the problem back to that for point-valued data, and hence traditional decision tree algorithms we call this approach AVG (for Averaging). To exploit the full information carried by the database, considers all the sample points that constitute each data set. The challenge here is that a training tuple can now "pass" a test at a tree node probabilistically when it's properly contains data the split point of the test.

In traditional decision-tree classification, a feature (an attribute) of a tuple is either categorical or numerical. For the latter, a precise and definite point value is usually assumed. In many applications, however, data uncertainty is common. The value of a feature/attribute is thus best captured not by a single point value, but by a range of values giving rise to a probability distribution. A simple way to handle data is to abstract probability distributions by summary statistics such as means and variances. We call this approach averaging.

Table1 is used to generate a decision tree; example table has emv, profit and mean values.

The above Fig1 decision tree is developed from the emv value based. Another approach is to consider the complete information carried by the probability distributions to build a decision tree. We call this approach Distribution-based. To the constructing decision tree classifiers on data with numerical attributes. An Averaging algorithm for building decision trees from data base using the Distribution-based approach; The Distribution-based approach could lead to a higher classification accuracy compared with the Averaging approach.

TABLE 1
Example Table

CName	ABC	BCD	CD	DE	EF	FG	GH
Price	103000	120000	100500	105310	99990	102030	125500
Cost of Tender	50000	50000	50000	50000	50000	50000	50000
Probability	15	15	15	15	15	15	15
Component cost	45000	45000	45000	45000	45000	45000	45000
EMV	41300	38750	41675	40954	41752	41446	37925
Profit	8000	25000	5500	10310	4990	7030	30500
Mean	15450	18000	15075	15796	14998	15304	18825

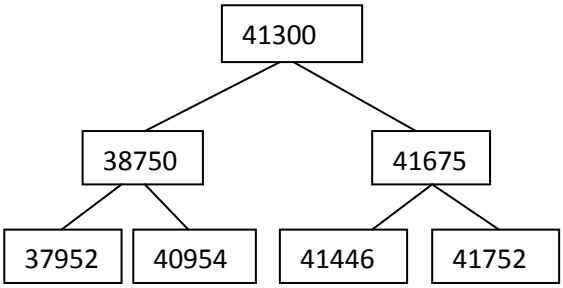

figure 1: Decision tree built form EMV tuples in example table.

B. Distribution-Based Approach

The same decision tree building frame work as described above for handling point data. Let us reexamine the example tuples in Table 1 to see how the distribution-based algorithm can improve classification accuracy. The key to building a good decision tree is a good choice of an attribute and a split point for each node, by taking into account the probability distribution.

The number of choices of a split point given an attribute is not limited to m − 1 point values, but the union of the domains of all pdfs $f_{i,jn}$ ∀ i = 1, . . . ,m, m is number of customers. Fig 2 will shows the process of nodes will move from the root R. The nodes value will getting from data base, the left l and right r inserted values will be compared with root value, and traversed in R_{ll}, R_{lr}, R_{rl}, R_{rr}. The accuracy is 100%. This example thus illustrates that by considering probability distributions rather than just expected values, we can potentially build a more accurate decision tree.

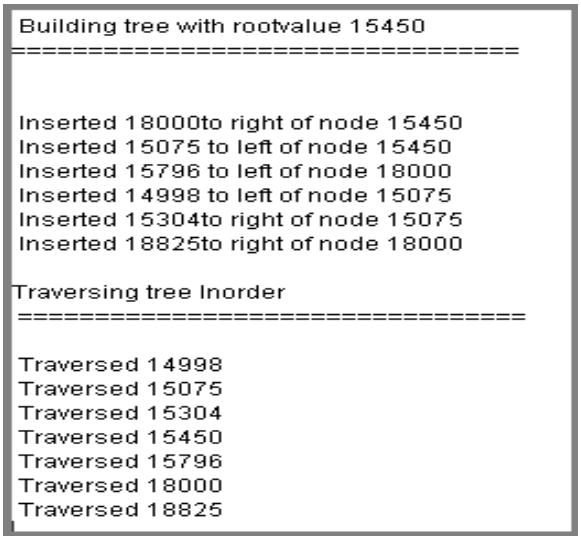

figure 2: From example table Tree nodes will be in decision tree format.

C. Experiments On Efficiency

The algorithms described above have been implemented in Java using JDK 1.6 and a series of experiments were performed on a PC with an Intel Core 2 Duo 2.66GHz CPU and 2GB of main memory, running Linux kernel 2.6.22 i686. An experiment on the accuracy of our novel distribution-based algorithm has been presented already in algorithm section.

The data sets used are the same as those used in Section algorithm, data taken from raw database. The results are show in column chart, column charts are used to compare values across categories. In these charts x-axis displays company names and y-axis displays price. Each dataset in every company will be displayed in different colors. Only Gaussian distribution is used for the experiments below.

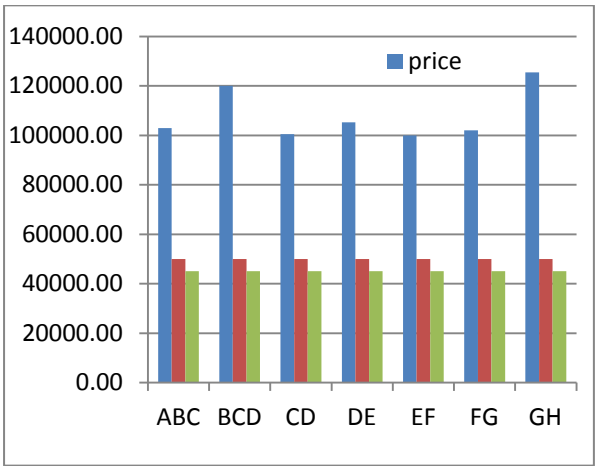

figure 3: Data Given in Database (above example table).

We first examine the execution time of the algorithms, which is charted in Fig 3, 4. In these figure's, 6 bars are drawn for each company. The vertical axis, which is in log scale, represents the execution price. We have given also the execution price of the AVG algorithm (see Section IV). Note that AVG builds different decision trees from those constructed by the Distribution based algorithms, and that AVG generally builds less accurate classifiers. The execution time of AVG shown in the Fig3 is for reference only. From the Fig4, we observe the following general (ascending) order of efficiency: price, cost of tender, component cost, emv (Expected Monetary Value), profit, mean.

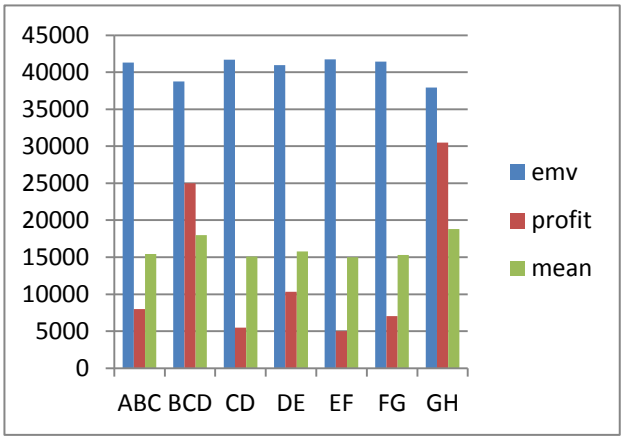

figure 4: These values calculated from example table Values.

This agrees with the successive enhancements of these techniques discussed in Section IV. Minor fluctuations are expected as the effectiveness depends on the actual distribution of the data. The AVG algorithm, which does not exploit the uncertainty information, takes the least time to finish, but cannot achieve as high an accuracy compared to the distribution-based algorithms (see Section IV). Among the distribution-based algorithms, emv is the most efficient.

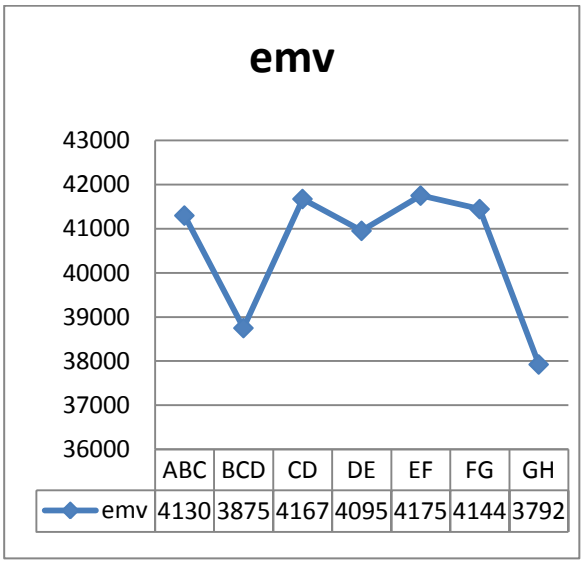

figure 5: Each company emv price displays on example table based.

The above Fig 5 shows emv values from the example table is represented in line chart. Line charts are useful and it compares pairs of values of each company. The effects of the number of values in the sample data table performance. The results are shown in Fig 6. The y-axis is in linear scale. For every data set, the execution time rises basically linearly with price. This is expected because with more sample points, the computations involved in the entropy calculation of each interval increases proportionately.

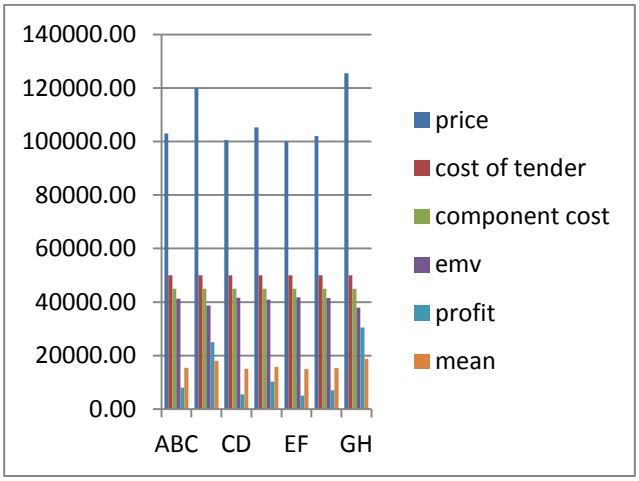

figure 6: From Example Table All Data Sets.

We have experimented with many real data sets with a wide range of parameters including the number of tuples, the number of attributes, and different application domains. Although values inevitably takes more time than the classical approach AVG in building decision trees, it can

potentially build more accurate trees because it takes the information into account.

V. CONCLUSION

The data is handled through "Averaging" by using means and variances. We have devised series of techniques to improve the tree construction efficiency. Some of these techniques are generalizations of analogous techniques for handling point-valued data. We have extended the model of decision-tree classification to accommodate data tuples having numerical attributes with uncertainty described by arbitrary values and data tree-construction algorithm.

We have found empirically that when suitable values are used, exploiting data uncertainty leads to decision trees with remarkably higher accuracies. Therefore we advocate that data be collected and stored with the information intact.

Performance is an issue, though, because of the increased amount of information to be processed, as well as the more complicated entropy computations involved. Therefore, we have devised a series of techniques to improve tree construction efficiency. Our algorithms have been experimentally verified to be highly effective. Their execution times are of an order of magnitude comparable to classical algorithms. Some of these pruning techniques are generalizations of analogous techniques for handling point-valued data. Although our novel techniques are primarily designed to handle data, they are also useful for building decision trees using classical algorithms when there are tremendous amounts of data tuples.

REFERENCES

[1] Smith Tsangy Ben Kaoy Kevin Y. Yipz Wai-Shing Hoy Sau Dan Lee, "Decision Trees for Uncertain Data". *Vol. 23, No. 1, pp.64-78, 2011*.

[2] R. Agrawal, T. Imielinski, and A. N. Swami, "Database mining: A performance perspective", *IEEE Trans. Knowl. Data Eng., 1993*.

[3] S. D. Lee, B. Kao, and R. Cheng, "Reducing UK-means to K-means", in 1st Workshop on Data Mining of Uncertain Data, *in ICDM, 2007*.

[4] Rakesh Agrawal, Sakti Ghosh, Tomasz Imielinski, Bala Iyer, and Arun Swami. "An Interval Classifier for Database Mining Applications", *VLDB 92, Vancouver, British Columbia, Canada, 1992, 560-573*.

[5] Shalom Tsur, "Data Dredging", *IEEE Data Engineering Bulletin, 13, 4, December 1990, 58-63*.

[6] J. R. Quinlan, "Learning logical definitions from relations," Machine Learning, *vol. 5, pp. 239–266, 1990*.

[7] T. Elomaa and J. Rousu, "General and efficient multisplitting of numerical attributes," Machine Learning, *vol. 36, no. 3, pp. 201–244, 1999*.

[8] U. M. Fayyad and K. B. Irani, "On the handling of continuous-valued attributes in decision tree generation," *Machine Learning, vol. 8, pp.87–102, 1992*.

[9] T. Elomaa and J. Rousu, "Efficient multisplitting revisited: Optima preserving elimination of partition candidates," *Data Mining and Knowledge Discovery, vol. 8, no. 2, pp. 97–126, 2004*.

[10] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and Regression Trees", *Wadsworth, Belmont, 1984*.

[11] T. Elomaa and J. Rousu, "Knowledge and Information Systems", *vol. 5, no. 2, pp. 162-182, 2003*.

[12] J. Ross Quinlan, "Simplifying Decision Trees", *Int. J. Man-Machine Studies, 27, 1987, 221-234*.