ISSN: 2250–3676

[IJESAT] INTERNATIONAL JOURNAL OF ENGINEERING SCIENCE & ADVANCED TECHNOLOGY Volume-2, Issue-4, 1030 – 1034

EXTENDING DECISION TREE CLASIFIERS FOR UNCERTAIN DATA

M. Suresh Krishna Reddy[1], R. Jayasree[2]

Aditya Engineering College, JNTU Kakinada, Dept Of Information Technology, Surampalem, Ap, India,
[1]mskrishnareddy@Gmail.Com, [2] Jaya_Raji2004@Yahoo.Co.In

Abstract

Traditionally, decision tree classifiers work with data whose values are known and precise. We extend such classifiers to handle data with uncertain information. Value uncertainty arises in many applications during the data collection process. Example sources of uncertainty include measurement/quantization errors, data staleness, and multiple repeated measurements. With uncertainty, the value of a data item is often represented not by one single value, but by multiple values forming a probability distribution. Rather than abstracting uncertain data by statistical derivatives (such as mean and median), we discover that the accuracy of a decision tree classifier can be much improved if the "complete information" of a data item (taking into account the probability density function (pdf)) is utilized. [1] We extend classical decision tree building algorithms to handle data tuples with uncertain values. Extensive experiments have been conducted that show that the resulting classifiers are more accurate than those using value averages. Since processing pdf's is computationally more costly than processing single values (e.g., averages).

----------***----------

1. INTRODUCTION

Classification is a classical problem in machine learning and data mining. 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 tuple's feature vector. [2]One of the most popular classification models is the decision tree model. Decision trees are popular because they are practical and easy to understand. Rules can also be extracted from decision trees easily. Many algorithms have been devised for decision tree construction. These algorithms are widely adopted and used in a wide range of applications such as image recognition, medical diagnosis, credit rating of loan applicants, scientific tests, fraud detection, and target marketing. Data uncertainty arises naturally in many applications due to arious reasons. We briefly discuss three categories here:

- Measurement errors,
- Data staleness,
- Repeated measurements.

a) Measurement Errors: Data obtained from measurements by physical devices are often imprecise due to measurement errors. As an example, a tympanic (ear) thermometer measures body temperature by measuring the temperature of the ear drum via an infrared sensor. A typical ear thermometer has a quoted calibration error of +/- $0{:}2^0$C, which is about 6.7% of the normal range of operation, noting that the human body temperature ranges from 37^0C (normal) and to 40^0C (severe fever) that along with other factors such as placement and technique, measurement error can be very high. For example, it is reported in [5] that about 24% of measurements are off by more than 0.5^0C, or about 17% of the operational range. Another source of error is quantisation errors introduced by the digitisation process. Such errors can be properly handled by assuming an appropriate error model, such as a Gaussian error distribution for random noise or a uniform error distribution for quantisation errors.

b) Data Staleness: In some applications, data values are continuously changing and recorded information is always stale. One example is location-based tracking system. The where about of a mobile device can only be approximated by imposing an uncertainty model on its last reported location[6]. typical uncertainty model requires knowledge about the moving speed of the device and whether its movement is restricted (such as a car moving on a road network) or unrestricted (such as an animal moving on plains). Typically a 2D probability density function is defined over a bounded region to model such uncertainty.

c) Repeated Measurements: Perhaps the most common source of uncertainty comes from repeated measurements. For example, a patient's body temperature could be taken multiple times during a day; an anemometer could record wind speed once every minute; the space shuttle has a large number of heat sensors installed all over its surface. When we inquire about a patient's temperature, or wind speed, or the temperature of a certain section of the shuttle, which values shall we use? Or, would it be better to utilise all the information by considering

the distribution given by the collected data values? As a more elaborate example, consider the "Breast Cancer" dataset reported in [7]. This dataset contains a number of tuples. Each tuple corresponds to a microscopic image of stained cell nuclei. A typical image contains 10–40 nuclei. One of the features extracted from each image is the average radius of nuclei. We remark that such a radius measure contains a few sources of uncertainty: (1) an average is taken from a large number of nuclei from an image, (2) the radius of an (irregularly-shaped) nucleus is obtained by averaging the length of the radial line segments defined by the centroid of the nucleus and a large number of sample points on the nucleus' perimeter, and (3) a nucleus' perimeter was outlined by a user over a fuzzy 2D image. From (1) and (2), we see that a radius is computed from a large number of measurements with a wide range of values. The source data points thus form interesting distributions. From (3), the fuzziness of the 2D image can be modelled by allowing a radius measure be represented by a range instead of a concrete point-value.

Yet another source of uncertainty comes from the limitation of the data collection process. For example, a survey may ask a question like, "How many hours of TV do you watch each week?" A typical respondent would not reply with an exact precise answer. Rather, a range (e.g., "6–8 hours") is usually replied, possibly because the respondent is not so sure about the answer himself. In this example, the survey can restrict an answer to fall into a few pre-set categories (such as "2–4 hours", "4–7 hours", etc.). However, this restriction unnecessarily limits the respondents' choices and adds noise to the data.

Clustering of uncertain data has recently attracted interests from researchers. This is driven by the need of applying clustering techniques to data that are uncertain in nature, and a lack of clustering algorithms that can cope with the uncertainty. Uncertainty in data arises naturally due to random errors in physical measurements, data staling, as well as defects in the data collection models. For instance, when track locations with GPS devices, the reported location can have errors of a few metres. When attempting to cluster the location of objects tracked using GPS, the errors may affect the clustering result.

Traditional clustering approaches model objects as having accurately known positions. This model does not cope well with uncertain data. It does not take into account the uncertainty inherent in the data, and may lead to undesirable clustering results because information on the uncertainty is dropped.[1] Owing to this shortcoming as well as the practical need to deal with data with uncertainty, there has been growing interest in developing problem models and algorithms
to handle uncertain data.

Rather than a single point in space, the location of each object is represented by a probability density function (pdf) over the space Rm being studied. Given a set of such objects, we want to divide them into k clusters, minimising the total expected distance (ED) from the objects to their cluster centres. We focus on the case where ED is defined using MSE (mean squared error).

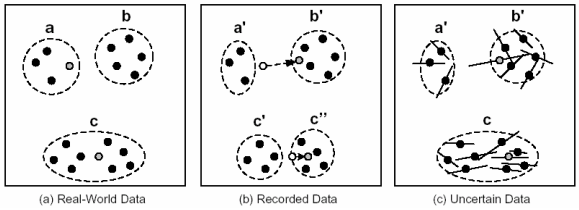

Figure 1. Data Clustering

1.1. Existing System

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. [4]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. Although the previous techniques can improve the efficiency of means, they do not consider the spatial relationship among cluster representatives, nor make use of the proximity between groups of uncertain objects to perform pruning in batch. [5]A simple way to handle data uncertainty is to abstract probability distributions by summary statistics such as means and variances. We call this approach Averaging. Another approach is to consider the complete information carried by the probability distributions to build a decision tree. We call this approach Distribution-based.

1.2 .Proposed System

We study the problem of constructing decision tree classifiers on data with uncertain numerical attributes. Our goals are
(1) to devise an algorithm for building decision trees from uncertain data using the Distribution-based approach;
(2) to investigate whether the Distribution-based approach could lead to a higher classification accuracy compared with the Averaging approach; and
(3) to establish a theoretical foundation on which pruning techniques are derived that can significantly improve the computational efficiency of the Distribution-based algorithms.

2. IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

2.1. Data Insertion:

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. With uncertainty, the value of a data item is often represented not by one single value, but by multiple values forming a probability distribution. This uncertain data is inserted by user.

2.2. Generate Tree:

Building a decision tree on tuples with numerical, point valued data is computationally demanding. A numerical attribute usually has a possibly infinite domain of real or integral numbers, inducing a large search space for the best "split point". [10]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

Pruning Algorithm: Although UDT can build a more accurate decision tree, it is not as efficient as AVG. As we have explained, to determine the best attribute and split point for a node, UDT has to examine k(ms - 1) split points, where k = number of attributes, m = number of tuples, and s = number of samples per pdf. (AVG has to examine only k(m - 1) split points.) For each such candidate attribute Aj and split point z, an entropy H(z;Aj) has to be computed. Entropy calculations are the most computation-intensive part of UDT. Our approach to developing more efficient algorithms is to come up with strategies for pruning candidate split points and entropy calculations.

Note that we are considering safe pruning here. We are only pruning away candidate split points that give sub-optimal entropy values.7 So, even after pruning, we are still finding optimal split points. Therefore, the pruning algorithms do not affect the resulting decision tree, which we have verified in our experiments. It only eliminates sub-optimal candidates from consideration, thereby speeding up the tree building process.

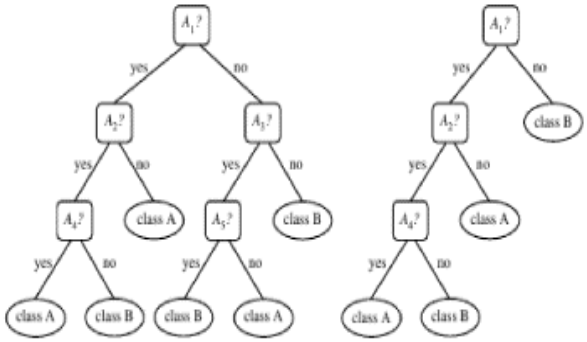

Figure 2. Un pruned and Pruned Decision tree

2.3. Averaging:

A simple way to handle data uncertainty is to abstract probability distributions by summary statistics such as means and variances. We call this approach Averaging. A straight-forward way to deal with the uncertain 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. AVG is a greedy algorithm that builds a tree top-down. When processing a node, we examine a set of tuples S. The algorithm starts with the root node and with S being the set of all training tuples. At each node n, we first check if all the tuples in S have the same class label. If so, we make n a leaf node and set Pn(c) = 1, Pn(c') = 0 for all c' not equal c. Otherwise, we select an attribute A_{jn} and a split point z_n and divide the tuples into two subsets: "left" and "right". All tuples with $v_{i;jn}<=z_n$ are put in the "left" subset L; the rest go to the "right" subset R. If either L or R is empty (even after exhausting all possible choices of A_{jn} and z_n), it is impossible to use the available attributes to further discern the tuples in S. In that case, we make n a leaf node. Moreover, the population of the tuples in S for each class label induces the probability distribution Pn.

In particular, for each class label c belongs to C, we assign to Pn(c) the fraction of tuples in S that are labelled c. If neither L nor R is empty, we make n an internal node and create child nodes for it. We recursively invoke the algorithm on the "left" child and the "right" child, passing to them the sets L and R, respectively. To build a good decision tree, the choice of A_{jn} and z_n is crucial. At this point, we may assume that this selection is performed by a blackbox algorithm BestSplit,

which takes a set of tuples as parameter, and returns the best choice of attribute and split point for those tuples. We will examine this blackbox in details. Typically, BestSplit is designed to select the attribute and split point that minimises the degree of dispersion.

The degree of dispersion can be measured in many ways, such as entropy (from information theory) or Gini index[3]. The choice of dispersion function affects the structure of the resulting decision tree.2 In this paper we assume that entropy is used as the measure since it is predominantly used for building decision trees.

2.4. Distribution Based:

An approach is to consider the complete information carried by the probability distributions to build a decision tree. We call this approach Distribution-based. Our goals are,
(1) To devise an algorithm for building decision trees from uncertain data using the Distribution-based approach;
(2) To investigate whether the Distribution-based approach could lead to a higher classification accuracy compared with the Averaging approach;
(3) To establish a theoretical foundation on which pruning techniques are derived that can significantly improve the computational efficiency of the Distribution-based algorithms.

For uncertain data, we adopt the same decision tree building framework as described above for handling point data. After an attribute A_{jn} and a split point z_n has been chosen for a node n, we have to split the set of tuples S into two subsets L and R. The major difference from the point-data case lies in the way the set S is split. Recall that the pdf of a tuple ti belongs to S under attribute A_{jn} spans the interval $[a_{i;jn}; b_{i;jn}]$. If $b_{i;jn} <= z_n$, the pdf of ti lies completely on the left of the split point and thus ti is assigned to L. Similarly, we assign ti to R if $z_n < a_{i;jn}$. If the pdf properly contains the split point, i.e., $a_{i;jn} <= z_n < b_{i;jn}$, we split ti into two fractional tuples tL and tR in the same way and add them to L and R, respectively. We call this algorithm UDT (for Uncertain Decision Tree).

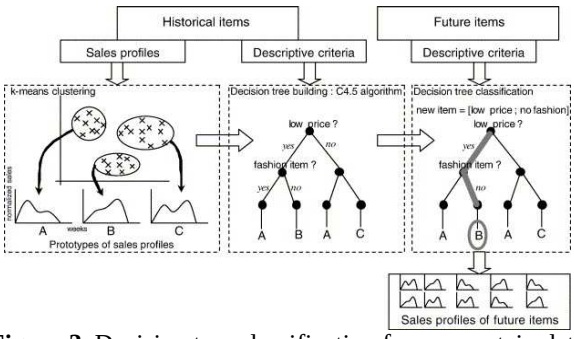

Figure 3. Decision tree classification from uncertain data

Again, the key to building a good decision tree is a good choice of an attribute Ajn and a split point zn for each node n. With uncertain data, however, the number of choices of a split point given an attribute is not limited to m - 1 point values. This is because a tuple ti's pdf spans a continuous range [ai;j ; bi;j]. Moving the split point from ai;j to bi;j continuously changes the probability pL and pR . This changes the fractional tuples tL and tR, and thus changes the resulting tree. If we model a pdf by s sample values, we are approximating the pdf by a discrete distribution of s points. In this case, as the split point moves from one end-point $a_{i;j}$ to another end-point $b_{i;j}$ of the interval, the probability pL changes in s steps. With m tuples, there are in total ms sample points. So, there are at most (ms -1) possible split points to consider. Considering all k attributes, to determine the best (attribute, split-point) pair thus require us to examine k(ms - 1) combinations of attributes and split points. Comparing to AVG, UDT is s time more expensive. Note that splitting a tuple into two fractional tuples involves a calculation of the probability pL, which requires an integration. We remark that by storing the pdf in the form of a cumulative distribution, the integration can be done by simply subtracting two cumulative probabilities.

CONCLUSION

We have extended the model of decision-tree classification to accommodate data tuples having numerical attributes with uncertainty described by arbitrary pdf's. We have modified classical decision tree building algorithms to build decision trees for classifying such data. We have found empirically that when suitable pdf's are used, exploiting data uncertainty leads to decision trees with remarkably higher accuracies. We therefore advocate that data be collected and stored with the pdf 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.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A. N. Swami, "Database mining: A performance perspective," IEEE Trans. Knowl. Data Eng., vol. 5, no. 6, pp. 914–925, 1993.

[2] J. R. Quinlan, "Induction of decision trees," Machine Learning, vol. 1, no. 1, pp. 81–106, 1986.

[3] C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993, ISBN 1-55860-238-0.

[4] M. Chau, R. Cheng, B. Kao, and J. Ng, "Uncertain data mining: An example in clustering location data," in PAKDD, ser. Lecture Notes in Computer Science, vol. 3918. Singapore: Springer, 9–12 Apr. 2006, pp. 199–204.

[5] R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J. S. Vitter, "Efficient indexing methods for probabilistic threshold queries over uncertain data," in VLDB. Toronto, Canada: Morgan Kaufmann, 31 Aug.–3 Sept. 2004, pp. 876–887.

[6] W. K. Ngai, B. Kao, C. K. Chui, R. Cheng, M. Chau, and K. Y. Yip, "Efficient clustering of uncertain data," in ICDM. Hong Kong, China: IEEE Computer Society, 18–22 Dec. 2006, pp. 436–445.

[7] S. D. Lee, B. Kao, and R. Cheng, "Reducing UK-means to K-means," in The 1st Workshop on Data Mining of Uncertain Data (DUNE), in conjunction with the 7th IEEE International Conference on Data Mining (ICDM), Omaha, NE, USA, 28 Oct. 2007.

[8] H.-P. Kriegel and M. Pfeifle, "Density-based clustering of uncertain data," in KDD. Chicago, Illinois, USA: ACM, 21–24 Aug. 2005, pp. 672–677.

[9] C. K. Chui, B. Kao, and E. Hung, "Mining frequent itemsets from uncertain data," in PAKDD, ser. Lecture Notes in Computer Science, vol. 4426. Nanjing, China: Springer, 22-25 May 2007, pp. 47–58.

[10] C. C. Aggarwal, "On density based transforms for uncertain data mining," in ICDE. Istanbul, Turkey: IEEE, 15-20 Apr. 2007, pp. 866– 875.

[11] O. O. Lobo and M. Numao, "Ordered estimation of missing values," in PAKDD, ser. Lecture Notes in Computer Science, vol. 1574. Beijing, China: Springer, 26-28 Apr. 1999, pp. 499–503.

[12] L. Hawarah, A. Simonet, and M. Simonet, "A probabilistic approach to classify incomplete objects using decision trees," in DEXA, ser. Lecture Notes in Computer Science, vol. 3180. Zaragoza, Spain: Springer, 30 Aug.-3 Sep. 2004, pp. 549–558.

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

BIOGRAPHIES

M Suresh Krishna Reddy, II year MTech, Dept of Information Technology, Aditya Engineering College, Surampalem, Andhra Pradesh, India
mskrishnareddy@gmail.com

R. Jayasree, Asst Professor, Dept of Information Technology, Aditya Engineering College, Surampalem, Andhra Pradesh, India,
jaya_raji2004@yahoo.co.in