Decision Tree: Review of Techniques for Missing Values at Training, Testing and Compatibility

Sachin Gavankar
Department of Computer Engineering
Datta Meghe College of Engineering, Mumbai University
Navi Mumbai, India
sachingavankar@yahoo.co.in

Sudhirkumar Sawarkar
Department of Computer Engineering
Datta Meghe College of Engineering, Mumbai University
Navi Mumbai, India
sudhir_sawarkar@yahoo.com

***Abstract* — Data mining rely on large amount of data to make learning model and the quality of data is very important. One of the important problem under data quality is the presence of missing values. Missing values can occur in both at the time of training and at the time of testing. There are many methods proposed to deal with missing values in training data. Many of them resort to imputation techniques. However, Very few methods are there to deal with the missing values at testing/prediction time. In this paper, we discuss and summarize various strategies to deal with this problem both at training and testing time. Also, we have discussed the compatibility between various methods at training and testing to achieve better results.**

Keywords* — *data mining, induction, decision tree, missing values, training data, testing data, compatibility

I. Introduction

In many problems data may have missing attribute values. Data could be missing due to error or limitation of measuring instrument, high cost for data acquisition, non-disclosure of data, etc. The existing data mining methods which converts data into information primarily based on assumption that the data is complete. As 'missingness' of attributes increases, it affects the prediction accuracy. Decision Tree is one of the algorithm commonly used in Data Mining. This 'missingness' problem can be there in both training data sets and testing data sets. There has been lot of work on handling the 'missingness' in training data. It affects the accuracy of classification model itself. There is comparatively less work for handling missing values in Test data.

The commonly used approaches to address this problem at the time of training and testing are discussed in next sections.

In most of the research these methods have been studied independently. In this paper we have tried to establish the relationship between training and testing methods and their compatibility to achieve better prediction accuracy.

II. Dealing with Missing Values at Training

A. Ignoring Data with Missing Attribute Values

The simplest approach to deal with missing values is to ignore the data having missing values. One of the way is to ignore all the records having missing values. This method is called as complete case analysis. In second method, do not consider the instances and/or attributes with high level of missing data, called as discarding instances and/or attributes. Deletion of data can result in the loss of large amount of valuable information. These methods should be applied only if missing data that are Missing Completely At Random (MCAR), as missing data that are not MCAR can bias the result.

B. Most Common Attribute Value

The most commonly occurred attribute value is imputed in place of missing values. The CN-2 algorithm [26] uses this concept.

C. Most Common Attribute Value within the class

This method is similar to above method except it selects most common attribute value with the class.

D. Method of assigning All possible Values

In this case, multiple training records are created each with possible attribute value in case of missing value. This introduces additional information in training data which is not true and hence induces bias and importance of actual known true value is reduced.

E. Method of assigning All possible Values of Class

The method is same as above only difference is instead of crating training records for all possible attribute values, only the possible attribute values for the specific class are used. It limits the bias to the possible attribute values for the class.

F. A Special LEM2 Algorithm

At the time of building block for a particular attribute, all the records having missing values are discarded. Then, a set of rules is induced by using the original LEM2 (Learnable Evolution Model) method [27].

G. Hot Deck and Cold Deck

In hot and desk method, an unknown value is substituted by a value from an estimated distribution from the current data. Initially data is segregated into clusters and then

978-1-4673-8675-3/15 $31.00 © 2015 IEEE
DOI 10.1109/AIMS.2015.29

complete cases in the cluster are used to substitute the unknown values. In cold deck method, the data set must be different than the current data set.

H. Null Value Strategy

In 'Null value' method missing values are treated as a regular value, 'Null' in tree construction and testing process. As values might be missing for certain reason, it might be good idea to assign a special value. There are two drawbacks of this method. Firstly, it does not try to identify the original known values as missing values are considered as equally as a original value. Another disadvantage is, in case of more than one actual missing values and replacing all of them by one value 'null' may not be correct. Also, subtrees can be built under the null branch, oddly suggesting that the missing value is more discriminating then the known value. The advantage of this method is that all the records can be used for tree building and it is very simple to implement. This approach handles the missing at the training time as well.

I. Prediction Model

In this method, prediction model is created to predict values that will replace the unknown values. The attribute with unknown values is considered as a class, and remaining attributes are uses as training data (attributes) for the predictive model. Attributes might have relationships (correlations) among themselves. The main drawback of this approach is that the predicted values are likely to be more consistent with this set of attributes than the true (missing) values would be. Another drawback is that if there are no correlation among the attributes, then approach may not be precise for predicting missing attribute values.

J. Imputation with k-Nearest Neighbour

k-nearest neighbor algorithm is used to estimate and impute missing data. k-nearest neighbor method can predict both qualitative attributes (the most frequent value among k-nearest neighbors) and quantitative attributes (the mean among k-nearest neighbors). The drawback of this method is that, whenever the k-nearest neighbor looks for the most similar instances, the algorithm searches through the complete data set.

K. C4.5 Strategy

C4.5 [12] does not replace the missing values. At the time of selection an attribute at splitting node, all instances with known value of that attribute are used for information gain calculation. After selection of an attribute, instance with known attribute values are split as per actual values and instances with unknown attribute value are split in proportionate to the split off known values. At the time of testing, a test instance with missing value is split into branches according to the portions of training examples falling into those branches and goes down to leaves. This has an advantage over the methods of discarding all incomplete instances in that fewer instances are being discarded when computing the best attribute.

L. The EM Algorithm

Expectation-Maximization (EM) algorithm introduced by Dempster, Laird and Rubin [15]. Here, data source is assumed to be from a certain (mixture of) parametric models. EM iteratively performs the following two steps. Estimation (E) step – Estimate the parameters in the model for the data source by using the known attribute-values and estimate of the missing attributes values obtained in the previous iteration of the M-step. Maximization(M) step – Fill in the missing values so as to maximize the likelihood function that is refined in each iteration by the E-step. There are two drawbacks of this method. The first one is that it assumes that the data source came from some parametric model(s) with Gaussian (k-Gaussians) being the most commonly used. Hence, almost all EM applications are applicable only for numerical attributes. The second one is while EM can be proved to converge, with the parametric model and numerical attribute assumption, nevertheless the convergence process tends to be very slow.

III. Dealing with Missing Values at Testing

A. Discard Testing Instance

This is the simplest approach of discarding test cases with unknown attribute values. However, in practice, it may not be acceptable to reject few cases for prediction.

B. Imputation

Imputation a class of methods by which an estimation of missing value or of its distribution is used to generate prediction. An unknown value is replaced by an estimation of the value.

C. C4.5 Strategy

In this approach, the distribution of possible missing value is estimated and corresponding model predictions are combined probabilistically. At the time of testing, a test instance with missing value is split into branches according to the portions of training examples falling into those branches and goes down to leaves.

D. Null Strategy

In Null Value strategy as discussed in training time, 'Null' is considered as a special value both at training and testing time.

E. Lazy Decision Tree (Reduced Feature Models/Known Value Strategy)

Friedman [8] suggested lazy decision tree approach where the prediction model is constructed at testing time based on the available test instance values. This is also known as 'Known values strategy'. During tree construction it uses only attributes whose values are known at testing. Hence it naturally handles the missing values at testing. The main drawback of this approach is its high computational cost as

different trees may be constructed for different test examples. We can also save few trees and make use of it at running time.

IV. Compatibility: Training vs. Testing Method

In existing literature missing value handling at Training and Testing have been considered separately. However, compatibility of methods at these two phases is very important for the prediction accuracy. We have summarized the same in following table.

TABLE I. Compatibility of Training and Testing Methods.

Testing Method	Training Method		
	C 4.5	Null	Imputation
C 4.5	Compatible	Partially Compatible	Compatible
Lazy	Compatible	Partially Compatible	Compatible
Null	Not Compatible	Compatible	Not Compatible
Imputation	Compatible	Partially Compatible	Compatible

A. Compatibility with Testing Method – C4.5

1) Training Method - C4.5 – Compatible: C4.5 uses similar technique of splitting missing value instances across multiple branches as per probability. However, there is no relation between splitting of training instance and testing instance. Both methods are compatible to each other. Both methods helps to improve the prediction accuracy individually.

2) Training Method - Null - Partially Compatible: C4.5 at training will split the missing value instance into multiple branches and all the branches with 'Null' values in classification model created at training will never be utilized.

3) Training Method - Imputation – Compatible: Imputation at training time, helps to select single path without splitting the training instance. It works well with C4.5 technique of splitting instance across multiple branches at testing. The combination helps to improve the accuracy individually.

B. Testing Method – Lazy Approach

1) Training Method - C4.5 – Compatible: C4.5 Training logic utilizes missing value instances by splitting them into multiple branches and lazy works on only know attribute values at testing. The knowledge imparted by using missing value at training and Lazy technique both help to improve the accuracy.

2) Training Method - Null - Partially Compatible: The knowledge imparted by instances with 'Null' values at training will never be utilized as Lazy testing method utilizes only known values. Lazy approach will work independently at training and it helps to improve the accuracy.

3) Training Method - Imputation- Compatible: Imputation at training time, helps to select single path without splitting the training instance. Lazy works for only know values. These two approaches work independently and both will contribute to improve overall accuracy.

C. Testing Method – Null

1) Training Method - C4.5 & Imputation – Not Compatible: In this combination, since C4.5 or Imputation method never creates any 'Null' branch, 'Null' approach is not compatible for testing.

2) Training Method - Null – Compatible: Treating null as a special value at training and testing are compatible with each other. It is best suited when there is any specific reason behind null value. It act as an attribute value.

D. Testing Method – Imputation

1) Training Method - C4.5 – Compatible: C4.5 training logic splits instances with missing value and utilizes during tree building. This additional knowledge is definitely useful when imputed value in testing instance takes particular path. Both methods helps to improve the prediction accuracy individually.

2) Training Method - Null – Partially Compatible: The instances with 'Null' values at training will never be utilized. Imputation testing method will impute specific value and will select appropriate branch at testing.

3) Training Method - Imputation – Compatible: Though similar technique, it works independently at training and testing and helps to improve prediction accuracy at both steps.

V. Experimental Comparisions

A. Data Set

We used WEKA J48 from the machine learning software WEKA [29]. We selected four data sets from UCI Machine Learning Repository [28]. We have divided data set into training data and testing data. 10% missing data introduced in both test and training data separately.

TABLE II. Data Sets Used in the Experiments

Data Set	No. of Attributes	Training Set	Testing Set
Breast	10	236	50
Credit	19	950	50
Diabetes	8	718	50
Iris	4	150	50

B. *Comparision of Handling Missing Values at Training*

To start with the data without missing value is classified and results are considered as baseline. We introduced 10% 'missingness' in data and kept testing data complete. Training data set is changed for different experiments by removing records with missing values, imputation by default value method (mean/mode) and data with missing values for testing accuracy of C4.5.

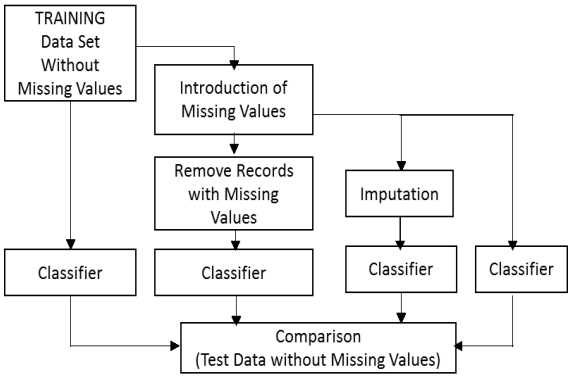

Figure 1. Preparation of Data Sets for handling the missing values in training data

TABLE III. PREDICTION ACCURACY OF 4 MISSING VALUE HANDING TECHNIQUES FROM TRAINING DATA.

Data Sets	Complete Data	Delete Records	Imputation	C4.5
Breast	78%	76%	76%	78%
Credit	70%	64%	74%	74%
Diabetes	80%	72%	74%	76%
Iris	96%	90%	96%	96%

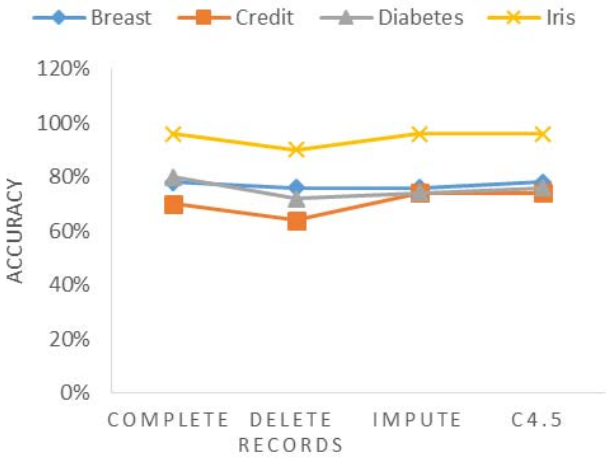

Figure 2. Comparison of Prediction accuracy of 4 missing value handing techniques from training data.

We can draw following conclusions from the results. First of all, C4.5 managed to predict with high accuracy followed by Imputation method. Secondly, for dataset Diabetes prediction accuracy came down for both the methods and for credit dataset overall accuracy is increases. It indicates that the performance of method is also depends on the dataset. Third, delete records strategy clearly not useful for handling missing values in training data.

C. *Comparision of Handling Missing Values at Testing*

In case of testing the accuracy for missing values at the time of testing we established the baseline with complete data. We introduced 20% 'missingness' in data and kept training data complete. Testing data set is changed for different experiments by imputation and using C4.5 method.

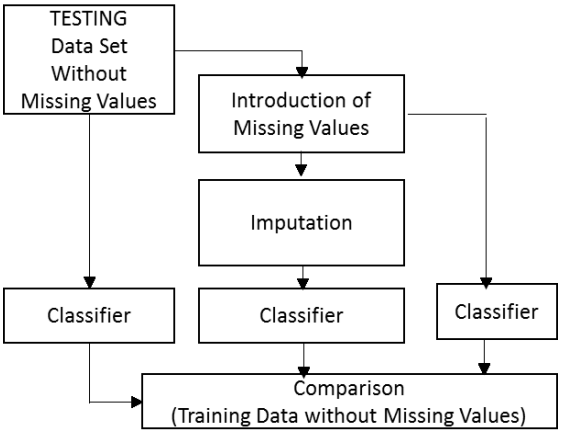

Figure 3. Preparation of Data Sets for handling the missing values in testing data

TABLE IV. PREDICTION ACCURACY OF 3 MISSING VALUE HANDING TECHNIQUES FROM TESTING DATA.

Data Sets	Complete Data	Imputation	C4.5
Breast	78%	80%	76%
Credit	70%	66%	74%
Diabetis	80%	72%	78%
Iris	96%	96%	92%

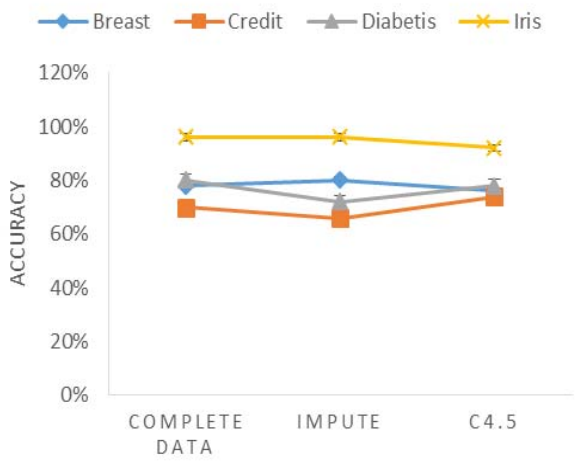

Figure 4. Comparison of Prediction accuracy of 3 missing value handing techniques from testing data.

Reduction in accuracy is more in case of Imputation method -0.13% compared to C 4.5 where it is -0.03%. Clearly C4.5 outperforms Imputation method to handle missing values in testing example.

VI. Conclusion and Future Work

Missing value problem has been studied in details mainly for training data and comparatively little work is done for testing data. This paper presented a comprehensive review of various methods for handling the issue at both training and testing time. C4.5 provides better prediction accuracy compared to other methods. We have attempted to evaluate the compatibility between few training and testing method. In future work, we plan to apply the lazy methods for performance evaluation and address the current drawbacks of lazy approaches.

References

[1] Jiawei Han, Michline Kamber, 'Data Mining Concepts and Technique' , Kaufmann Pulications, 2001

[2] David Hand, Heiki Mannila, Padhraic Smyth, 'Principles of Data Mining', The MIT Press.

[3] Tom Mitchell, 'Machine Learning', Mc-GrawHill Publications

[4] Rubin, D.B., (1976) Inference and Missing Data. Biometrika 63 581-592

[5] Schafer, J.L., (1997) The Analysis of Incomplete Multivariate Data. Chapman & Hall

[6] S. Zhang "Missing is Useful: Missing Values in Cost-Sensitive Decision Trees", IEEE Transactions on Knowledge and Data Engineering, Vol 17, No. 12, 2005.

[7] Maytal Saar-Tsechansky, Foster Provost, "Handling Missing Values when Applying Classification Models", Journal of Machine Learning Research 8 (2007) 1625-1657.

[8] J. Friedman, Y. Yun, and R. Kohavi, "Lazy Decision Tree," Proc. 13th Nat'l Conf. Artificial Intelligence, pp. 717-724, 1996.

[9] K.M. Ali and M.J. Pazzani, "Hydra: A Noise-Tolerant Relational Concept Learning Algorithm," *Proc. 13th Int'l Joint Conf. Artificial Intelligence (IJCAI93),* R. Bajesy, ed., pp. 1064-1071, 1993.

[10] C.J. Date and H. Darwen, "The Default Values Approach to Missing Information," *Relational Database Writings 1989-1991,* pp. 343-354, 1989.

[11] Charles X. Ling , Qiang Yang , Jianning Wang , Shichao Zhang, "Decision trees with minimal costs," Proc. 21st Int'l Conf. Machine Learning (ICML 04),2004.

[12] J.R.Quinlan, 'C4.5 Programs for Machine Learning', Morgan Kaufmann Publications, San Mateo, CA, 1993.

[13] Maytal Saar-Tsechansky "Handling Missing Values when Applying Classification Models", Journal of Machine Learning Research 8 (2007) 1625-1657

[14] Gustavo, Batista, Maria Monard "An Analysis of Four Missing Data Treatment Methods for Supervised Learning" Applied Artificial Intelligence: An International Journal Vol. 17, Issue 5-6, 2003.

[15] A.P. Dempster, N.M. Laird, D. B. Rubin. Maximum-likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, B39: 1-38, 1977.

[16] Amitava Karmaker, Stephen Kwek "Incorporating as EM-Approach for Handling Missing Attribute- Values in Decision Tree Induction". 5th Int'l Conf on Hybrid Intelligent Systems(HIS), 2005.

[17] Shichao Zhang, Xindong Wu, Manlong Zhu "Efficient missing data imputation for supervised learning", Proc. 9th Int'l Conf. on Cognitive Informatics (ICCI'10), 2010.

[18] Jun Wu, Yo Seung Kim, Chi-Howa Song, Won Don Lee "A New Classifier to Deal with Incomplete Data", Proc. 9th ACIS Int'l Conf. on Soft Engg, AI, Networking and Parallel/Distributed Computing. 2008.

[19] Kiran Siripuri, M Venugopal Reddy, "Classification of Uncertain Data using Decision Trees", IJARCSSE, Vol-3, Issue-10,Oct-2013.

[20] Jerzy W, Grzymala-Buss, Ming Hu, "A Comparison of Several Approaches to Missing Attribute Values in Data Mining" RSCTC 2000.

[21] Twala, B. E. T. H.; Jones, M. C. and Hand, D. J. "Good methods for coping with missing data in decision trees" Pattern Recognition Letters, 29(7), pp. 950–956, 2008.

[22] Smith Tsang, Ben Kao, Kevin Y. Yip, Wai-Shing Ho and Sau Dan Lee, "Decision Trees for Uncertain Data", IEEE, VOL. 23, NO. 1,1-15, 2011.

[23] Masahiro Sugimoto, Masahiro Takada, Masakazu Toi, " Comparison of robustness against missing values of alternative decision tree and multiple logistic regression for predicting clinical data in primary breast cancer" 35th Int'l. Conf. of the IEEE EMBS, 2013.

[24] Charu Aggarwal, Chen Chen, Jiawei Han, "On the Inverse Classification Problem and its Applications", ICDE'06, Georgia, USA.

[25] N. Landwehr, M. Hall, and E. Frank. Logistic model trees. Machine Learning, 59(1-2):161-205, 2005.

[26] Clark, P. Niblett, T.: The CN2 induction algorithm. Machine Learning 3 (1989)261-283.

[27] Grzymala-Busse, J. W. and Wang A.Y.: Modified Algorithms LEM1 and LEM2 for rule induction from data with missing attribute values. Proc. of the Fifth International Workshop on Rough Sets and Soft Computing (RSSC'97) at the Third Joint Conference on Information Sciences (JCIS'97), Research Triangle Park, NC, Mar 2-5, 1997, 69-72.

[28] UCI data summary

[29] I. H. Witten, E. Frank. Nuts and bolts: Machine Learning alogrithms in java. In:Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation, pp.265-320. Morgan-Kaufmann, 2000.