Binary Journal of Data Mining & Networking 1 (2010) 12-14

http://www.arjournals.org/index.php/bjdmn/index

Research Article

ISSN: 2229 -7170

Calculate missing value using association rules mining

Anil Rajput[1], Manmohan singh[2], Pooja Shrivastava[3]

***Corresponding author:**
Anil Rajput
[1] Department of Mathematics and Computer Science Govt. PG Nodal College, Sehore (M. P.) India.
[2]Mewar University, Rajasthan.
[3]Barkatullah University, Bhopal. India

Abstract

Discovering hidden knowledge from hug amount of data in form of association rules mining have become very popular in scaling field of data mining. One several algorithms have been also developed for mining association rules. All those algorithms can be effectively applied on all as dataset where data has not any time granularity means non-temporal dataset. The quality of training data for knowledge discovery in databases (KDD) and data mining depends upon so many factors, but also handling missing values is considered to be a crucial factor in whole data quality. Today in real world datasets contains missing values due to human, in operational error, hardware malfunctioning and many other factors.

Keywords: Data Mining, Temporal Frequent Pattern and Missing Value Quality of training data, association rules mining,

Introduction

A missing value can signify a number of different different things. Perhaps the field was not applicable, the event did not happen. It could be that the person who entered the data did not know the right value, or did not care if a field was not filled in.

There are many data mining scenarios in which missing values provide important information. The meaning of the missing values depends largely on context. For example, a missing value for the date in a list of invoices has a meaning substantially different from the lack of a date in column that indicates an employee hire date. Generally, Analysis Services treats missing values as informative and adjusts the probabilities to incorporate the missing values into its calculations. By doing so, you can ensure that models are balanced and do not weight existing cases too heavily.

Therefore, Analysis Services provides two distinctly different mechanisms for managing and calculating missing values. The first method controls the handling of nulls at the level of the mining structure. The second method differs in implementation for each algorithm, but generally defines how missing values are processed and counted in models that permit null values.

Specifying Handling of Nulls

To the data mining algorithm, missing values are informative. In case tables, Missing is a valid state like any other. Moreover, a data mining model can use other values to predict whether a value is missing. In other words, the fact that a value is missing is not an error. When you create a mining model, a Missing state is automatically added to the model for all discrete columns.

Adjusting Probability for Missing States

In addition to counting values, Analysis Services calculates the probability of any value across the data set. The same is true for the Missing value. For example, the following table shows the probabilities for the cases in the example:

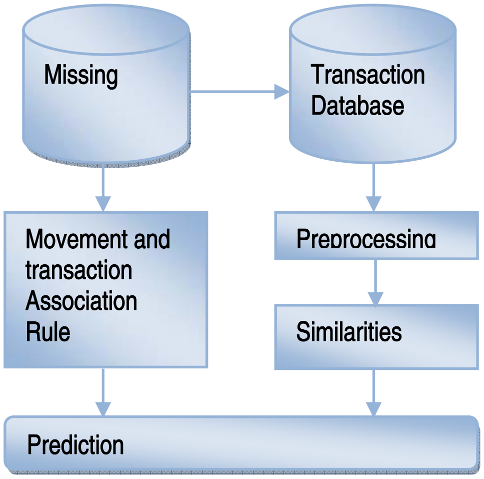

Figure1: User next prediction

Table1

Value	Cases	Probability
0	9296	50.55%
1	9098	49.42%
Missing	0	0.03%

It may seem odd that the probability of the Missing value is calculated as 0.03%, when the number of cases is 0. In fact, this behavior is by design, and represents an adjustment that lets the model handle unknown values gracefully. In general, probability is calculated as the favorable cases divided by all possible cases. In this example, the algorithm computes the sum of the cases that meet a particular condition ([Bike Buyer] = 1, or [Bike Buyer] = 0), and divides that number by the total count of rows. However, to account for the Missing cases, 1 is added to the number of all possible cases. As a result, the probability for the unknown case is no longer zero, but a very small number, indicating that the state is merely improbable, not impossible.

The addition of the small Missing value does not change the outcome of the predictor; however, it enables better modeling in scenarios where the historical data does not include all possible outcomes.

Special Handling of Missing Values

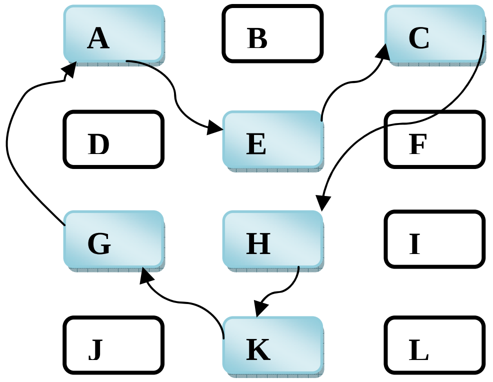

Figure 2: Missing Values

The Microsoft Decision Trees algorithm calculates probabilities for missing values differently than in other algorithms. Instead of just adding 1 to the total number of cases, the decision trees algorithm adjusts for the Missing state by using a slightly different formula. In a decision tree model, the probability of the Missing state is calculated as follows:

StateProbability = (NodePriorProbability)* (StateSupport+1)/ (NodeSupport + TotalStates)

Moreover, in SQL Server 2012 Analysis Services (SSAS), the Decision Trees algorithm provides an additional adjustment that helps the algorithm compensate for the presence of filters on the model, which may result in many states to be excluded during training.

In SQL Server 2012, if a state is present during training but just happens to have zero support in a certain node, the standard adjustment is made. However, if a state is never encountered during training, the algorithm sets the probability to exactly zero. This adjustment applies not only to the Missing state, but also to other states that exist in the training data but have zero support as result of model filtering.

This additional adjustment results in the following formula:

StateProbability = 0.0 if that state has 0 support in the training setELSEStateProbability=(NodePriorProbability)* (StateSupport + 1) / (NodeSupport + TotalStatesWithNonZeroSupport)

The net effect of this adjustment is to maintain the stability of the tree.

Conclusion

Information about additional time periods that you expect to see in the data. By default, time series models will try to automatically detect a pattern in the data. if you already know the expected time cycle, providing a periodicity hint can potentially improve the accuracy of the model. However, if you provide the wrong periodicity hint, it can decrease accuracy; therefore, if you are not sure what value should be used?

References

[1]. Tseng VS, Tsui CF, "Mining Multi-Level and Location-Aware Associated Service Patterns in Mobile Environments, "IEEE Trans. Systems, Man and Cybernetics: Part B, vol. 34, no. 6, pp. 2480-2485, Dec. 2004.

[2]. Tseng VS Lin WC, "Mining Sequential Mobile Access Patterns Efficiently in Mobile Web Systems," Proc. Int'l Conf. Advanced Information Networking and Applications, pp. 867-871,Mar. 2005.

[3]. Tao Y, Faloutsos C, Papadias D, Liu B, "Prediction and Indexing of Moving Objects with Unknown motion patterns," Proc. ACM SIGMOD Conf. Management of Data, pp. 611-622, June 2004.

[4]. Patel JM, Chen Y, Chakka VP, "Stripes: An Efficient Index for Predicted Trajectories," Proc. ACM SIGMOD Conf. Management of Data, pp. 635-646, June 2004.

[5]. Lu Y, "Concept Hierarchy in Data Mining: Specification, Generation and Implementation," master's thesis, Simon Fraser Univ., 1997.

[6]. Yin X, Han J, Yu PS, "LinkClus: Efficient Clustering via Heterogeneous Semantic Links," Proc. Int'l Conf. Very Large Data Bases, pp. 427-438, Aug. 2006.

[7]. Jeh G, Widom J, “SimRank: A Measure of Structural-Context Similarity,” Proc. Int’l Conf. Knowledge Discovery and Data Mining, pp. 538-543, July 2002.

[8]. Xin D, Han J, Yan X, Cheng H. “Mining Compressed Frequent-Pattern Sets,” Proc. Int’l Conf. Very Large Data Bases, pp. 709-720, Aug. 2005.

[9]. Han J. Fu Y. “Discovery of Multiple-Level Association Rules in Large Database,” Proc. Int’l Conf. Very Large Data Bases, pp. 420-431, Sept. 1995.

[10]. Chen MS, Park JS, Yu PS, “Efficient Data Mining for Path Traversal Patterns,” IEEE Trans. Knowledge and Data Eng., vol. 10, no. 2, pp. 209-221, Apr. 1998.

[11]. Jeung H, Liu Q, Shen HT, Zhou X. “A Hybrid Prediction Model for Moving Objects,” Proc. Int’l Conf. Data Eng., pp. 70-79,Apr. 2008.

[12]. Yun CH, Chen MS. “Mining Mobile Sequential Patterns in a Mobile Commerce Environment,” IEEE Trans. Systems, Man, and Cybernetics, Part C, vol. 37, no. 2, pp. 278-295, Mar. 2007.

[13]. Tao Y, Papadias D, Sun J. “The tpr*-tree: An Optimized Spatio-Temporal Access Method for Predictive Queries,” Proc. Int’l Conf. Very Large Data Bases, pp. 790-801, Sept. 2003.

[14]. Han J, Pei J, Yin Y. “Mining Frequent Patterns without Candidate Generation,” Proc. ACM SIGMOD Conf. Management of Data, pp. 1-12, May 2000.