**Abstract**

Discovering hidden knowledge from huge amounts of data in the form of association rules mining has become very popular in the scaling field of data mining. Several algorithms have been developed for mining association rules. All those algorithms can be effectively applied on all datasets where data has no time granularity means non-temporal datasets. The quality of training data for knowledge discovery in databases (KDD) and data mining depends upon so many factors, but 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, 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 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:

<table>
<thead>
<tr>
<th>Transaction Database</th>
<th>Movement and transaction Association Rule</th>
<th>Preprocessing</th>
<th>Similarities</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing</td>
<td></td>
<td></td>
<td></td>
<td>Figure 1: User next prediction</td>
</tr>
</tbody>
</table>
Table 1

<table>
<thead>
<tr>
<th>Value</th>
<th>Cases</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>9296</td>
<td>50.55%</td>
</tr>
<tr>
<td>1</td>
<td>9098</td>
<td>49.42%</td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0.03%</td>
</tr>
</tbody>
</table>

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

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:

\[
\text{StateProbability} = \frac{(\text{NodePriorProbability}) \times (\text{StateSupport} + 1)}{\text{NodeSupport} + \text{TotalStates}}
\]

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:

\[
\text{StateProbability} = \begin{cases} 
0.0 & \text{if that state has 0 support in the training set} \\
\text{ELSE} \frac{(\text{NodePriorProbability}) \times (\text{StateSupport} + 1)}{\text{NodeSupport} + \text{TotalStatesWithNonZeroSupport}} & \text{otherwise}
\end{cases}
\]

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


