Comparative Performance of Arm and Farm on a Normalised Datasets

Prachi Singh Thakur, Jitendra Agrawal

School of information technology, Rajiv Gandhi Technological University, Bhopal -462036, Madhya Pradesh, India

Prachi_singh19@yahoo.co.in, jitendra@rgtu.net

ABSTRACT

Association rule mining is basically used to generate association rules on a real life datasets. A well-known algorithm called apriori is used to generate the frequent pattern itemsets for a given transaction. Since real life dataset consist of nominal, continuous, integer attribute fields, to convert it into binary format some type of pre-processing has to be done on the dataset.

In this paper, we had evaluated the performance of two algorithms that is ARM(Association Rule Mining) and FARM(Fuzzy Association Rule Mining) on the basis of generation time by supplying different support and confidence values for data pre-processing, two methods are used: discretisation and normalisation. Discretisation converts the range of possible values of continuous data into subranges which is identified by a unique integer label. It also convert values associated with instances to corresponding integer label. Normalisation process converts values of nominal data into corresponding integer labels.

1. Introduction

Datamining [1] is known as knowledge discovery in databases (KDD) which is recognized as a new area for database research. It has attracted a great deal of attention in the information industry [5]. It has different meanings such as knowledge mining from data, knowledge extraction, data/pattern analysis, data archaeology, data dredging, different processing steps in KDD includes data cleaning, data integration, data selection, data transformation, data mining, pattern evaluation, knowledge presentation.

Association rule mining [3] is an important research topic in data mining and knowledge discovery which is introduced by R.Agrawal, T. imielinski and A. swami. Association rule (AR) [6] is of the form AB, where A (antecedent) and B (consequent) are disjoint sets of items. There are two interesting measures in AR they are support and confidence.

2. Association Rules

Association rule mining [6] is an important technique in data mining, which has been applied in industries of retailing, insurance and banking. The main task of association rule discovery is to extract frequent itemsets from data and to generate association rules from these frequent itemsets. Association rules [7] finds interesting association rules are kind of patterns representing correlation of attribute value (items) in a given set of data provided by a process of data mining system.
2.1 Definition of Association Rules

Association rules [8] can be defined as: Let I= \{I_1, I_2, I_3, ..., I_n\} be a set of items. Let D be a set of database transactions where each transaction T is a set of items such that T is a set of items such that T\text{ teach} transaction is associated with an identifier, called Tid. Let X be a set of items. A transaction T is said to contain X if implication of the form X \rightarrow Y, where X I, Y I and X \cap Y = \Phi. The rule X \rightarrow Y holds in the transaction set D with support s, where s is the percentage of transactions in D that contain XUY.

2.2 Phases of Association Rule Mining

Generally association rule mining process overcome with two problems they are:

1. First is the efficiency of algorithm. Most of the research is being done to find out an algorithm with less computation complexity.

2. Second is to find out the interesting rules from discovered rules.

2.3 Interestingness of Association Rules

To find the association rules there are two interestingness measures they are support and confidence. The rule A \rightarrow B [8] has confidence C in the transaction in D contains X that also contains Y. This is taken to be conditional probability (B | A). That is

\[
\text{Support} (A \rightarrow B) = \mu (A \cup B)
\]

\[
\text{Support} (A \rightarrow B) = \text{support} (A \cup B)
\]

\[
\text{Support} = \frac{|X \cup Y|}{|T|}
\]

\[
\text{Confidence} (A \rightarrow B) = \frac{\mu X \cap Y}{\mu A}
\]

Rules will be strong if it satisfies both a minimum support threshold (min_supp) and a minimum confidence threshold (min_conf).

2.3 Definition of Fuzzy Association Rules

Fuzzy association [9] rules use fuzzy logic to convert numerical attributes to fuzzy attributes, like “Income=High”, thus maintaining the integrity of information conveyed by such numerical attributes.

For a given database attribute it is necessary to utilize fuzzy sets so that meaningful association rules can be created. For domain D, let F be fuzzy set which is defined by a membership function of form

\[
\mu: D \rightarrow [0, 1]
\]

**Formal definition:** Fuzzy association [10] rule are defined as: Let T be a database, attribute I and fuzzy sets related to I are given. The fuzzy association rules are in the form of: X is A \rightarrow Y is B.

Let X = \{x_1, x_2, ..., x_p\} and Y = \{y_1, y_2, y_3, ..., y_q\} are subsets of itemset I and \Phi. Set A = \{f_{x_1}, f_{x_2}, f_{x_3}, ..., f_{x_p}\} and set B = \{f_{y_1}, f_{y_2}, f_{y_3}, ..., f_{y_q}\} include the fuzzy sets respectively related with X and Y. This first half of the rule X is known as the premise is B known as the result.

3. Data Pre-processing

Data pre-processing is an important step in data mining, it involves steps such as data cleaning, data integration, data transformation and data reduction.
The main task of data pre-processing is to discretize the items of itemset I and code the discretized data to meet the requirements of Apriori algorithm.

4. Apriori Algorithm

Apriori [11] is a widely used algorithm for mining frequent itemsets; this algorithm is based on the Apriori property, which means that all non-empty subsets of a frequent itemset must also be frequent. The frequent itemset [12] and association rules mining problem has received a great deal of attention and many algorithms have been proposed to solve this problem.

Generally Apriori algorithm generates\((k+1)\)-candidates by joining frequent \(k\)-itemset. The Apriori algorithm[2] is a seminal algorithm for mining frequent itemsets for Boolean association rules. It explores the level-wise mining Apriori property that all nonempty subsets of a frequent itemset must also be frequent. At the \(k^{th}\) iteration \((k>=2)\), it forms frequent itemset must also be frequent.

\begin{verbatim}
k←1
nextlevelFlag=true;
generate candidate k-itemsets
loop
count support values for candidate k-itemsets
prune unsupported k-itemsets
k←2
generate candidate k2 itemsets from previous level
if no k2 itemsets break
end loop
\end{verbatim}

Basic Apriori Algorithm

5. Apriori-T Algorithm

Apriori-T algorithm[13] is an association rule mining(arm) algorithm, developed by the LUCS-KDD research team. This algorithm mainly consist of support-confidence framework. The input to the software, is the binary valued dataset \(D\). The set \(S\) comprises a set of \(R\) records such that each record\(R\) consist of \(A\) attributes. The given dataset has \(m\) columns and \(n\) rows. This software is implemented in java which makes it highly portable. This software compares four source files and with three application classes.

6. Fuzzy Apriori-T Algorithm

Fuzzy association rule mining [14] is intended to address the crisp boundary problem encountered in traditional ARM. The principal idea is that ranged...
values can belong to more than one sub-ranges. The total membership degree that associates it with each available sub-range is the total membership degree that associates it with each available sub-range.

6.1 Crisp Boundary Problem

Since traditional association rule mining (ARM) algorithms are concerned with finding the useful or interesting pattern from the binary valued datasets. Classical ARM requires all attributes to be binary valued for example, yes, no, true-false, 0-1, etc. but real datasets consist of combinations of different attributes. These datasets are not binary valued.

6.2 Principle of Fuzzy Apriori-T Algorithm

To overcome the crisp boundary problem in traditional ARM association rule mining (ARM) is introduced. The principle behind this is to range the values that belong to more than one sub-range, we say that this value has a membership degree that associates it with each available sub-ranges. The total of membership degree for single attribute equals to one. Fuzzy apriori-T algorithm is software this software is the fuzzy version of apriori-T algorithm. The support for a 1-itemset is the sum of the membership degree values divided by the number of records in the dataset.

7. Proposed Method

In proposed methodology we had calculated the performance of Apriori-T and Fuzzy Apriori-T algorithm. This software consist of ARM algorithms that uses support-confidence framework. The input to the software is in binary valued dataset D. Performance of our algorithm is based on generation time of creating association rules with different support and confidence values. For this we had used 3 datasets of UCI data repository.

7.1 Data Pre-processing

Since real datasets comprise of nominal data, continuous data and integer data. To convert it into binary data, its attribute fields must be converted into binary fields. Therefore, real datasets require preprocessing step. To do this, we require discretisation and normalisation process i.e., conversion into a binary valued format. We define discretisation and normalisation as follows:

Discretisation is the process of converting the range of possible values associated with a continuous data item into a number of sub-ranges each identified by a unique integer label and converting all the values associated with instances of this data item to the corresponding integer labels. Normalisation is a process of converting values associated with nominal data items so that they correspond to unique integer labels.

![Figure 1. Real dataset applied to discretisation/normalisation process](image-url)
6. Experimental Results

We have experimented our algorithms on 3 datasets of UCI repository of machine learning databases. We have evaluated the efficiency of algorithms by calculating the generation time on different support/confidence values. Characteristics of the datasets for experiment is given below:

<table>
<thead>
<tr>
<th>Normalize DVR dataset</th>
<th>No. Of Records</th>
<th>No. Of Columns</th>
<th>No. Of Columns (Schema)</th>
<th>No. of attributes</th>
<th>Density (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abalone</td>
<td>2148</td>
<td>9</td>
<td>9</td>
<td>51</td>
<td>17.65</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>155</td>
<td>20</td>
<td>20</td>
<td>388</td>
<td>5.15</td>
</tr>
<tr>
<td>Housing</td>
<td>506</td>
<td>14</td>
<td>14</td>
<td>93</td>
<td>15.05</td>
</tr>
</tbody>
</table>

Table 1: Details of Dataset Used For Comparison

Table 1 shows the details of three datasets i.e Abalone, Hepatitis, Housing dataset which is taken from UCI data repository. Firstly these datasets are pre-processed by using discretisation and normalization. Process consists of number of records, number of columns, number of columns in schema, number of attributes, and density in percent.

<table>
<thead>
<tr>
<th>(Support-confidence)%</th>
<th>Execution Time(secs)</th>
<th>(Support-confidence)%</th>
<th>Execution Time(secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5%-35%)</td>
<td>0.05</td>
<td>(0.05%-0.35%)</td>
<td>0.04</td>
</tr>
<tr>
<td>(10%-40%)</td>
<td>0.03</td>
<td>(0.10%-0.40%)</td>
<td>0.03</td>
</tr>
<tr>
<td>(15%-45%)</td>
<td>0.03</td>
<td>(0.15%-0.45%)</td>
<td>0.02</td>
</tr>
<tr>
<td>(20%-50%)</td>
<td>0.02</td>
<td>(0.20%-0.50%)</td>
<td>0.01</td>
</tr>
<tr>
<td>(25%-55%)</td>
<td>0.00</td>
<td>(0.25%-0.55%)</td>
<td>0.00</td>
</tr>
<tr>
<td>(30%-60%)</td>
<td>0.00</td>
<td>(0.30%-0.60%)</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 2 Results For Abalone Dataset Showing the Execution Time

Figure 3 shows the results for abalone dataset showing the execution time in seconds by supplying different support-confidence values. Figure 3 shows the graph between Apriori-T and Fuzzy Apriori-T in abalone dataset. The graph shows that execution time of Apriori-T and Fuzzy Apriori-T is faster than Apriori-T.
Table 3: Results for Hepatitis Dataset Showing the Execution Time

<table>
<thead>
<tr>
<th>(Support-confidence)%</th>
<th>Apriori-T Execution Time(secs)</th>
<th>Fuzzy Apriori-T (Support-confidence)%</th>
<th>Execution time(secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5%-35%</td>
<td>4.86</td>
<td>0.05%-0.35%</td>
<td>3.32</td>
</tr>
<tr>
<td>10%-40%</td>
<td>2.06</td>
<td>0.10%-0.40%</td>
<td>2.01</td>
</tr>
<tr>
<td>15%-45%</td>
<td>1.25</td>
<td>0.15%-0.45%</td>
<td>1.00</td>
</tr>
<tr>
<td>20%-50%</td>
<td>0.39</td>
<td>0.20%-0.50%</td>
<td>0.21</td>
</tr>
<tr>
<td>25%-55%</td>
<td>0.13</td>
<td>0.25%-0.55%</td>
<td>0.06</td>
</tr>
<tr>
<td>30%-60%</td>
<td>0.14</td>
<td>0.30%-0.60%</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Table 3: Results for Housing Dataset Showing the Execution Time

<table>
<thead>
<tr>
<th>(Support-confidence)%</th>
<th>Apriori-T Execution Time(secs)</th>
<th>Fuzzy Apriori-T (Support-confidence)%</th>
<th>Execution time(secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5%-35%</td>
<td>0.42</td>
<td>0.05%-0.35%</td>
<td>0.36</td>
</tr>
<tr>
<td>10%-40%</td>
<td>0.31</td>
<td>0.10%-0.40%</td>
<td>0.25</td>
</tr>
<tr>
<td>15%-45%</td>
<td>0.05</td>
<td>0.15%-0.45%</td>
<td>0.04</td>
</tr>
<tr>
<td>20%-50%</td>
<td>0.03</td>
<td>0.20%-0.50%</td>
<td>0.02</td>
</tr>
<tr>
<td>25%-55%</td>
<td>0.01</td>
<td>0.25%-0.55%</td>
<td>0.01</td>
</tr>
<tr>
<td>30%-60%</td>
<td>0.0</td>
<td>0.30%-0.60%</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 3 and Figure 4 shows the results for hepatitis dataset showing the execution time in seconds by supplying different support-confidence values. Figure 3 shows the graph between Apriori-T and Fuzzy Apriori-T in abalone dataset. The graph shows that execution time of Apriori-T and Fuzzy Apriori-T is faster than Apriori-T.

Table 4 and Figure 5 shows the results for housing dataset showing the execution time in seconds by supplying different support-confidence values. Figure 3 shows the graph between Apriori-T and Fuzzy Apriori-T in abalone dataset. The graph shows that execution time of Apriori-T and Fuzzy Apriori-T is faster than Apriori-T.
7. CONCLUSION

In this paper, we had evaluated the performance of two algorithms that is Apriori-T and fuzzy Apriori-T on the basis of execution time of frequent itemsets occurrence by supplying support and confidence values. We had experimented our algorithms on three datasets of UCI repository. We had used pre-processing technique to convert the real dataset into binary format for this discretisation and normalisation process is used. Discretisation is used to convert continuous data item into number of sub ranges which is identified by a unique integer label. We had used normalisation to convert nominal data items into unique integer labels. At last we had concluded that fuzzy Apriori-T algorithm takes lesser computation time than Apriori-T algorithm which is shown in the graph form for different datasets.

REFERENCES


