A New Sequential Pattern Discovery Algorithm for Web Usage Mining.

BUNYAT THIPMOUD, WORAPOJ KREESURADEJ
Data Mining and Data Exploration Laboratory,
Faculty of Information Technology,
King Mongkut’s Institute of Technology Ladkrabang,
Ladkrabang, Bangkok 10520
THAILAND

Abstract: - In this paper, we propose a new algorithm that the main objective of discovery rules used in prediction model. Focusing on latest-substring rules that are the order and adjacency information that model experiences a monotonically increasing precision curves. Meanwhile, the reduced rule could be done at the same time as searching. The rule discovered would be recorded in Full Table. Then rule is reduced by confidence factor and support factor and recorded in Trim table. Last procedure is pruning rules that will discard redundant and inconsistent rules then the rest will have high accuracy.

Key-Words: - Web usage mining, web request prediction, sequential pattern discovery algorithm

1. Introduction

One important data source for this study is the web log that traces the user’s web browsing. The data collected by these traces could predict the next step as the idea of repeated behavior. Since web logs contain data that describe patterns of web pages usage, such as IP addresses, sizes, date and time, one can discover user access patterns to web servers by analyzing the web logs, and then express the patterns in form of web-document prediction models. This paper proposes sequential pattern discovery model that can be advantage for prediction with high accuracy of the user’s next requests based on web-log data.

During recent years, there has been research on the other choices to build a classifier. Pitknow et al [5] suggested a way to make predictions based on K\textsuperscript{th}-order Markov models. Because they thought longer paths were more reliable than shorter paths, and could display user’s access pattern better. However, their algorithm is deficient in that longer paths are rarer in web log history; thus the noise in longer paths could be higher than is shorter paths. This can result in the undesirable effect of reduced accuracy. Su et al [7] also proposed an intuitive way to build the model from multiple N-gram and select the best prediction by applying a smoothing algorithm called “Cascading” model, which trusted longer N-gram rules. About using web-document prediction is personalizing the web experience for user [1]. WebWatcher “follows” a user browses the web and identifies links that is potentially interesting to the use. WUM (Web Utilization Miner) is application in business intelligence. The area of using prediction model is aiming at improving the performance of Internet system, such as bringing down the network traffic, shortening user’s perceived waiting period, and reducing network latency and web server workloads [8]. The hybrid Markov model proposed in [9] to establish a decision-theoretic WWW presending model; their model reduce the time Internet users have to wait while network transmission costs rise.

Problem Decomposition: Referring to problem in [6], regarding to discover and reduce all association rules. Size of slide window must be specified and this would affect in rule discovery. If the specified size is too small, the longer rule will not match the rule then some rule might disappear. Assuming that the size of slide window = 3, the long path could not be discovered such as LogT = \{A, B, C, D, E, F\} then we have the rule size as \{A, B, C\} \rightarrow D but it will not discover \{A, B, C, D\} \rightarrow E, \{A, B, C, D, E\} \rightarrow F. In case that the size of slide window is too long. For example slide window = 4, it could not discover the smaller rule such as LogT =\{A,B,C,D,E\}, we would have \{A,B,C,D\} \rightarrow E then the lost rule is \{A\} \rightarrow B, \{B\} \rightarrow C, \{A,B\} \rightarrow C
It is noticeable that the problem stated above as rule of 1, 2, 3.. n sequential pattern rule could not reach decision criteria. It, then, could not be considered in any further stages. The problem solving in this paper that the size of slide window, is adaptive window size could confidentially solve the problem described above.

2. Rule Representation Method

Given a web log. We consider web log data as a sequence of distinct web pages where subsequences, such as users’ sessions can be observed within unusually long gaps between consecutive requests.

<table>
<thead>
<tr>
<th>Time</th>
<th>User ID</th>
<th>Requested Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:00:01</td>
<td>U1</td>
<td>A</td>
</tr>
<tr>
<td>00:00:02</td>
<td>U2</td>
<td>B</td>
</tr>
<tr>
<td>00:00:03</td>
<td>U2</td>
<td>C</td>
</tr>
<tr>
<td>00:00:04</td>
<td>U3</td>
<td>D</td>
</tr>
<tr>
<td>00:00:05</td>
<td>U1</td>
<td>E</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

Table 1 Example user visit sequence

This sequence is divided into user sessions according to user Ids Shown in table 2.

<table>
<thead>
<tr>
<th>User ID</th>
<th>Session Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>A, E,…</td>
</tr>
<tr>
<td>U2</td>
<td>B, C,…</td>
</tr>
<tr>
<td>U3</td>
<td>D,…</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

Table 2 Extracted user sessions

We now discuss how to extract rules of the form LHS⇒RHS from the log table. As it was mentioned before in this paper, the RHS in each association rule is the next page requested by the user. This is a different method to extract rule.

The representation is called the latest-substring rules. This is also known as n-gram rules in some literature [8]. These rules are not only take into account the order and adjacency information, but also the recent information about the LHS string.

<table>
<thead>
<tr>
<th>LHS-RHS</th>
<th>RHS</th>
<th>Extracted Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, B, C</td>
<td>D</td>
<td>&lt;A, B, C&gt;⇒E, &lt;B, C&gt;⇒D, &lt;A&gt;⇒D, &lt;C&gt;⇒D</td>
</tr>
</tbody>
</table>

Table 3 The latest-substring rule

For each rule of the form LHS-RHS, we define the support and confidence as follows:

\[
\text{sup} = \frac{\text{count}(\text{LHS}, \text{RHS})}{\text{count}(\text{Table})} \quad \text{conf} = \frac{\text{sup}(\text{LHS}, \text{RHS})}{\text{sup}(\text{LHS})}
\]

For the equations above, the function Count (Table) is the record set in the log table. The way of pruning here is exactly the same as in all association mining algorithms [4]. The approaches of association rule mining, two parameters: minimum support and minimum confidence can be pre-defined to filter out some rules with very low support and/or minimum confidence [4].

The latest substring representation interested references the experiment such as [6].

2.1 Extract processing of log file table:

- Firstly, the system will read all data from Log file then transaction log file incurred as Transaction Log File \( Tlog = \{R1, R2, R3, \ldots Rn\} \).
- Data cleaning is the next step. The method of cleaning server log is to eliminate irrelevant items of web log analysis. The output processed by cleaning model will result in untruth data, the best use of cleaning model applied to HTML log file.
- The output clean model could be interpreted as web transaction table. Web transaction table is written as \( \text{WebT} = \{R1, R2, R3, \ldots Rn\} \).
- Lastly, reading each records as \( \text{Read} \leftarrow \{P1, P2, \ldots, Pn\}\) goes by IP number. The result is kept in variable.

3. Algorithm

Definitions:
- **Support**: the support of a rule “X ⇒ Y” is the probability (or attribute sets) X and Y occurring together in the same transaction
- **Confidence**: the confidence of a rule “X ⇒ Y” is defined as the probability of occurrence of X and Y together in all transaction in which X already occurs.
- **Minsup**: Minsup is an input parameter to the algorithm for generating association rules
- **Minconf**: Minconf is an input parameter that defines the minimum level of confidence that a rule must process
- **Web Transaction Table**: is where all transactions on web that were already passed cleaning model
- **Moving window pair (LHS-RHS)**: To capture the sequential and time limited nature of predictions that composted of two adjacent windows.
- **Antecedent window (LHS)**: It holds all visited
page(s) within a given number of user requests and up to a current time.

**Consequent window (RHS):** It holds all future visited page(s) or predict page within a number of user requests from the current time instant

**Antecedent Rule (A):** Rules recorded in Trim Table redundant routes were eliminated and counted as one rule

**iCW:** is a number of current windows that was use when algorithm slide moving pair window

**Full Table:** is structure array \([m][n] = \{\text{LHS}, \text{LHS-RHS}, \text{sup}, \text{conf}\}\)

**Trim Table:** is structure array \([m][n] = \{\text{LHS}, \text{LHS-RHS}, \text{sup}, \text{conf}\}\) after Rules from Full Table were reduce then were keep in Trim Table. Trim Table was use for comparing in next step.

**Example 1:**

Let \(iCW = 1\) Moving window pair = \(D \rightarrow E\)

Let \(iCW = 2\) Moving window pair = \(C, D \rightarrow E\)

Therefore: \(RHS = "E"\)

\[\text{LHS}_{iCW=1} = "D"\] and \(\text{LHS}_{iCW=2} = "C, D"\)

\(\text{A}_{R(iCW-1)} = D \rightarrow E\) when \(iCW = 2\)

Figure 3 describes the First Only algorithm. This algorithm works in case of \(iCW_1\) only because Rule that comes from LHS-RHS at the beginning calculated to find out Confidence and Support. LHS-RHS used in this calculation must not be compared with Trim Table before. It is noticeable that Trim Table is firstly established after \(iCW_1\) then used in comparison for the first time on \(iCW_2\).

Figure 1 shows the main algorithm. At the first step, the web transaction table will be created by the application of data from log file. Next step runs to Cleaning Model, all irrelevant data will be eliminated, also the applicable of new technique of web page as the web programming; ASP, PHP, for example.

From these step processed, web transaction according to user’s identification (IP) showing user’s behavior focusing on web page visit is generated. This table shows each user’s behavior in consequence of web pages visit. The consequence resulted from time stamping by the time that user visit each of web page and the next stop.

At the second step, the quantity of all web transaction’s record set is known. On the case that \(iCW_i\), the Algorithm First only will be processed that result in rule as record in KeepRule then Same Table. This Same Table is used for both IP and Rule record on the purpose of inspection. If rule incurred repeatedly from same IP were counted as one. When the calculation is completed, values resulted by calculation could be recorded in Full Table, used in record and all calculation such as confidence and support.

Lastly, step 3 enhanced by split window as \(iCW\) will identify the size of split windows. Rule that incurred would be recorded in Same Table then process again on Discover Rule Processing. The input of this part is Full Table as Trim Table. The procedure of this stage is Rule Reduction by discard rule with support value that is lower than specified. Then rule with confidence = 100% will be recorded. By this method of rule’s record, it has never been applied to any other rule comparison.

---

**Begin main:**

1. Open Web Log File;
   - Extract web log record in to a main table.
   - Data Cleaning
   - Extract Web Transaction according to user’s identification (IP).
   - Grouping web transaction based on web page visited. The record will be stamped according to the time consequence.
   - Web transaction Table according to IP separation resulted from previous grouping. However, this table is in order of IP that visit web page consequently, according to order of web page.

2. While \((Rs!=NULL)\) do /*Record set=1 to EOF
3. Select record set from Web Transaction Table
   - Let \(\{P_1, P_2, \ldots, P_n\}\) be the set of visited pages, where \(P_n\) is a predict page
   - For \((iCW = 2 \text{ to } N)\) do /* Current Window start at 1 to N
   - For \((\text{LHS-RHS=2 to N})\) do /*Extend window size
     i. KeepRule \(\leftarrow (\text{LHS-RHS})\)
     ii. SameTable \([1..n]\) \(\leftarrow\) KeepRule /*Remark;
     iii. Function: Discover Rule Processing
     END for
   
   Function: Reduce Rule Processing
   END for
   
   END while

END;

---

**Remark:**

Remark: record on the purpose of whether rule that incurred from same IP or not

**Fig. 1:** Algorithm for Discover and Reduce rule of Web Usage Mining

---
**Function** Discover Rule Processing:

**Input:** KeepRule saves rule from Moving window pair.

**Output:** Full Table records analyzed value.

**Analysis KeepRule:**

\[
\text{KeepRule} = \{\text{Antecedent window (LHS), Antecedent Rule (Ar), Moving window pair (LHS-RHS)}\};
\]

**While** (Trim Table [iCW-1][1..n] do

- \(A_r = \text{TrimTable}[iCW-1][1..n]\) do
  - Count number of (LHS);
  - Count number of (LHS-RHS);
  - Confidence = sup(LHS-RHS)/sup(LHS)
  - Support = count (LHS-RHS)/count(Table)
  - FullTable[iCW][n] \(\leftarrow \{\text{LHS, LHS-RHS, Confidence, Support}\};\)

**End if**

**End while**

**End function** Discover Rule Processing:

---

**Function** Reduce Rule Processing:

**Input:** Full Table [iCW][n].

**Output:** Trim Table [iCW][n].

**While** (EOF Full Table) do

- Token \(\leftarrow\) Full Table [iCW][n];
  - if \((\text{token}\_\text{support} > \text{Support}_{\text{min}})\) do /* Reduction Rule by discard
    - if \((\text{token}\_\text{confidence} != 100\%)\) do /* Reduction Rule by create node
      - Trim Table [iCW][n] \(\leftarrow\) Full Table [iCW][n];
    - end if
  - end if

**End if**

**End while**

**End function** Reduce Rule Processing:

---

**Fig. 2:** Algorithm Discover Rule Processing

---

**Fig. 3:** Algorithm Reduce Rule Processing

---

Begin:

Do (iCW = 1)

- KeepRule \(\leftarrow\) Moving window pair /* An variable record in moving window pair
  - Same Table \(\leftarrow\) KeepRule /* Remark 1:
  - KeepRule = \{Antecedent window (LHS), Moving window pair (LHS-RHS)\}
  - Analysis Confidence and Support Value
  - FullTable[iCW][n] \(\leftarrow\) Conf., Sup.

**End do**

Trim Table [iCW = 1][n] \(\leftarrow\) Full Table [iCW = 1][n];

**End**

---

**Remark**

**Remark 1:** record on the purposed of whether rule that incurred from same IP or not

---

**4. Experiments**

The rules established at the beginning of Algorithm experiment. These are D, E \(\rightarrow\) Y, X1 \(\rightarrow\) C and C, D, E \(\rightarrow\) Z. These rules were set to compile with log table then additional information is filled out. Lastly algorithm principle mentioned in this paper was applied and resulted as following.

Table 4; Web Transaction Table, illustrates all data of 7 IP. Given each users have different access pattern calling as P1 to P5, all of these data were already processed Cleaning Model. After we have data as shown in Web Transaction Table 4, all data will be processed called Algorithm First Only. This procedure involves iCW with Count Table = 7 at the beginning, we would get data as Table. Moving Window Pair which has same identification that is incurred from same user by IP inspection method would be recorded as one record. The repeated access pattern would be discarded.

\[
\begin{array}{cccccc}
\text{IP} & \text{P1} & \text{P2} & \text{P3} & \text{P4} & \text{P5} \\
1 & D & E & D & E & Y \\
2 & X4 & X5 & D & E & Y \\
3 & X1 & C & D & E & Z \\
4 & X1 & C & D & E & Z \\
5 & X1 & C & D & E & Z \\
6 & E & D & E & Y & X \\
7 & D & E & Y & X1 & X2 \\
\end{array}
\]

**Table 4 Web transaction table**

For example: IP1 has Access Pattern; D \(\rightarrow\) E, E \(\rightarrow\) D, D \(\rightarrow\) E, E \(\rightarrow\) Y. It is noticeable that D \(\rightarrow\) E incurred twice, and then access pattern of the repeated on is discarded. The count method is only one time; LHS \(\rightarrow\) RHS = 1 and LHS = 2. The benefit of unrepeated count is described earlier, hence the end of this Algorithm First Only generated data showing in Table 5; Full Table 1

\[
\begin{array}{cccccc}
\text{RULE} & \text{LHS} & \text{LHS-RHS} & \text{Sup} & \text{Conf} \\
D & E & 8 & 7 & 1 & 0.875 \\
E & D & 9 & 2 & 0.285 & 0.222 \\
E & Y & 9 & 4 & 0.571 & 0.444 \\
X4 & X5 & 1 & 1 & 0.142 & 1 \\
X5 & D & 1 & 1 & 0.142 & 1 \\
X1 & C & 4 & 3 & 0.428 & 0.75 \\
C & D & 3 & 3 & 0.428 & 1 \\
E & Z & 9 & 3 & 0.428 & 0.333 \\
\end{array}
\]
Based on this Table, LHS, LHS – RHS, Support, and Confidence are recoded with the obviously that $C \rightarrow D$ has Confidence = 100%. This means whenever Access User starts at C, it would absolutely result in Consequent Window (RHS) as D without any questions. When data of Full Table 1 completed the process of Reduce Rule (filter out), data in Trim Table 1 appeared which some rules that value of support is less then threshold will filter out (if support $\leq 0.142$), such as $X_4 \rightarrow X_5$, $X_5 \rightarrow D$, $Y \rightarrow X$, $Y \rightarrow X_1$, $X_1 \rightarrow X_2$ while the rest rules are recorded in Trim Table 1 would be $D \rightarrow E$, $E \rightarrow D$, $E \rightarrow Y$, $X_1 \rightarrow C$, $E \rightarrow Z$ as shown in Table 6.

Table 6 Full Table 2

Table 6 works as following procedure. When iCW2 slide window with the value that compares with rule recorded in Table of iCW2-1 or Trim Table 1, LHS-RHS from slide window would be split into 3 major parts called LHS, LHS – RHS and $A_R$. Where as LHS and LHS – RHS applied in support and confidence value calculation, $A_R$ takes an important role to record value within itself compared to rule. If the result of this comparison states the same value, it means LHS – RHS currently be able to process in the next step. In case that $A_R$ could not find any value in Trim Table 1, shift window moving pair to {P1, P2, …, Pn} continues to incur. The last value recorded in Full Table 2 will be filter out and then the rest are kept in Trim Table 2 for further next process. It is obviously seen that the rest value is quite small with the Rule of $D \rightarrow E$, $D \rightarrow Y$.

Table 7 Trim Table 2 and Full Table 3

As you may noticed that the more difficulties access pattern has, the less of support value incurred, Table 6 with additional iCW=4, compared with Trim Table 3 resulted in none identical value hence no Rule in Full Table 4. This could be called as the end of process.

Table 8 Full Table 3

We construct the Tree according to the following relations:
- Each rule is represented by a node in the Tree.
- The node representing the direct parent rule is the parent node of the node(s) representing the direct children rule(s).
- The root of the Tree representing the default rule.

The pruning process just traverses the tree using post-order traverse. If a not has a lower confidence than that of its direct parent nodes, or it predicts the same class as its direct parent node, it can be pruned and all of its children nodes (if any) promoted to be the children of its direct parent node [6]. Figure 5 is a diagram tree created from Table 4 after Discovery and reduce rule completed. Rules left as show in Figure 6 that every single nodes have value of Conf. According to the concept of pruning tree method stated earlier, tree that completed the process will be as Figure 7 while;

- $E \rightarrow Y_{Conf} < Root_{Conf}$ then pruned and promote $D, E \rightarrow Y$ that node will not be pruned because the value of conf is more than root. It is essential that $E, D \rightarrow Y$ was pruned because prediction is same as prediction of parent that is node $D \rightarrow Y$.
- $E \rightarrow D_{Conf} < Root_{Conf}$ was pruned.
- $E \rightarrow D$ and $C, D \rightarrow E$ was pruned because the prediction is same as parent prediction.
- $E \rightarrow Z_{Conf} < Root_{Conf}$ was pruned then promoted $D, E \rightarrow Z$ to be parent of $C, D, E \rightarrow Z$.
- $D, E \rightarrow Z_{Conf} < Root_{Conf}$ was pruned then promoted $C, D, E \rightarrow Z$. 

We construct the Tree according to the following relations:
- Each rule is represented by a node in the Tree.
- The node representing the direct parent rule is the parent node of the node(s) representing the direct children rule(s).
- The root of the Tree representing the default rule.

The pruning process just traverses the tree using post-order traverse. If a not has a lower confidence than that of its direct parent nodes, or it predicts the same class as its direct parent node, it can be pruned and all of its children nodes (if any) promoted to be the children of its direct parent node [6]. Figure 5 is a diagram tree created from Table 4 after Discovery and reduce rule completed. Rules left as show in Figure 6 that every single nodes have value of Conf. According to the concept of pruning tree method stated earlier, tree that completed the process will be as Figure 7 while;

- $E \rightarrow Y_{Conf} < Root_{Conf}$ then pruned and promote $D, E \rightarrow Y$ that node will not be pruned because the value of conf is more than root. It is essential that $E, D \rightarrow Y$ was pruned because prediction is same as prediction of parent that is node $D \rightarrow Y$.
- $E \rightarrow D_{Conf} < Root_{Conf}$ was pruned.
- $E \rightarrow D$ and $C, D \rightarrow E$ was pruned because the prediction is same as parent prediction.
- $E \rightarrow Z_{Conf} < Root_{Conf}$ was pruned then promoted $D, E \rightarrow Z$ to be parent of $C, D, E \rightarrow Z$.
- $D, E \rightarrow Z_{Conf} < Root_{Conf}$ was pruned then promoted $C, D, E \rightarrow Z$. 

We construct the Tree according to the following relations:
- Each rule is represented by a node in the Tree.
- The node representing the direct parent rule is the parent node of the node(s) representing the direct children rule(s).
- The root of the Tree representing the default rule.

The pruning process just traverses the tree using post-order traverse. If a not has a lower confidence than that of its direct parent nodes, or it predicts the same class as its direct parent node, it can be pruned and all of its children nodes (if any) promoted to be the children of its direct parent node [6]. Figure 5 is a diagram tree created from Table 4 after Discovery and reduce rule completed. Rules left as show in Figure 6 that every single nodes have value of Conf. According to the concept of pruning tree method stated earlier, tree that completed the process will be as Figure 7 while;

- $E \rightarrow Y_{Conf} < Root_{Conf}$ then pruned and promote $D, E \rightarrow Y$ that node will not be pruned because the value of conf is more than root. It is essential that $E, D \rightarrow Y$ was pruned because prediction is same as prediction of parent that is node $D \rightarrow Y$.
5. Conclusion

This algorithm introduced in this paper is a discovery rule used in prediction model. The idea of reduction irrelevant data and counting precisely could show the capacity of the rule. While the reduce rule could be done at the same time as searching. The rule is reduced, by confidence and support, and recorded in Trim table. Last procedure is pruning rule that will discard redundant and inconsistent rule [6] then the rest will have high accuracy. This prediction helps in either e-commerce to provide client’s choice of products or offer the fast view of next page to satisfy those who visit website applying this algorithm.

References:
[6] Ian Tian Yi Li. Web-document prediction and presenting using association rule sequential classifiers, Simon Fraser University, S2001