International Journal of Electronic Commerce Studies
Vol.13, No.3, pp.001-020, 2022
doi: 10.7903/ijecs.2061

Prediction of Web Browsing Behavior based on Sequential Data Mining

Li-Ching Ma*
National United University, Taiwan
lcma@nuu.edu.tw

Pei-Pei Hsu
I-Shou University, Taiwan
pphsu@isu.edu.tw

ABSTRACT

Discovering time-related transaction behavior or patterns is helpful for businesses in suggesting appropriate products to their customers. For web systems, it is important to understand customers' browsing behavior to design or recommend products or services that customers need. This study proposes an approach for predicting web browsing behavior that integrates the concepts of sequential data mining, Borda majority count, bit-string operation, and PrefixSpan algorithm. By incorporating the concept of Borda majority count and sequential data mining, the proposed approach can discover majority-based priorities of items for recommendation and improve prediction accuracy. In addition, the proposed approach employs the concept of bit-string operation and the PrefixSpan algorithm to increase computational efficiency. This research employs the concept of ensemble methods that combine multiple models to derive improved results. Compared to previous methods, the proposed approach can yield higher prediction accuracy. Moreover, the proposed approach can provide flexibility for decision-makers in adjusting a minimum support level and the number of items for recommendation. The proposed approach can also be applied to many fields.

Keywords: Data mining, prediction, web browsing behavior, sequential data mining, web recommendation

1. INTRODUCTION

Discovering time-related transaction behavior or patterns can help businesses in suggesting appropriate products to their customers. With the e-commerce explosion, the need for reliable recommendation systems has greatly increased to assist managers in analyzing customers' buying behavior and recommending next-items that customers are most likely to buy in the near future [1, 2]. For web systems, it is also important to understand customers' browsing behavior in order to design or recommend products or services that customers need.

Sequential pattern mining is an important data mining technique that is used to find frequent time-related behavior from a sequential database [3, 4]. Mining sequential patterns can reveal the sequential purchasing behavior of most customers from a large transaction database [5]. Agrawal and Srikant [6] first discussed the sequential pattern mining problems to deal with customer transactions involving different transaction times. They proposed two well-known Apriori-based algorithms: AprioriAll and AprioriSome [6]. This research aims to predict time-related web browsing behavior based on the concept of sequential data mining.

However, most sequential data mining algorithms are not very efficient because they need to repeatedly generate candidates or scan the database multiple times. In order to improve the efficiency of sequential data mining, Yen and Lee [7] employed bit-string operations to efficiently discover the association rules and sequential patterns; all sequential data are transformed into binary bit strings in advance, and template masks are generated and used to find sequential patterns. Pei et al. [5] introduced a PrefixSpan algorithm to increase the computational efficiency of sequential data mining; it divided the sequence database into several projected databases and displayed the underlying sequential pattern by examining local frequent patterns in each projected database. Both approaches can increase the computational efficiency of sequential data mining. This research incorporates the concepts of bit-string operation and PrefixSpan algorithm into sequential data mining to enhance computational efficiency.

In addition, many methods have been proposed to discover group priorities that can be used to sequentially order recommendations. For instance, Zahid and Swart [8] developed a Borda majority count method based on the concept of the Borda count [9] to discover the order of alternatives by calculating the sum of scores for each alternative. The Borda majority count is a simple way to determine the group priority of alternatives according to the intensity of user preference and counting results; however, users may attain no consensus regarding the final results. Chen and Cheng [10] proposed a consensus mining approach to explore maximum consensus sequences of alternatives among group users based on the concept of the Apriori algorithm [11]. However, the problems considered were not time-related. This research incorporates the advantages of the Borda majority count and consensus mining to improve the prediction accuracy of sequential data mining.

This study proposes an approach for predicting web browsing behavior that integrates the concepts of sequential data mining, Borda majority count, bit-string operation, and the PrefixSpan algorithm. The major advantages of the proposed approach are listed below:

(i) The proposed approach can discover majority-based priorities of items for recommendation and improve prediction accuracy by incorporating the concepts of Borda majority count and sequential data mining.

(ii) The proposed approach can increase computational efficiency by employing the concept of bit string operation and the PrefixSpan algorithm.

(iii) The proposed approach can provide flexibility for decision-makers in adjusting a minimum support level and the number of items for recommendation.

The rest of this paper is organized as follows. Section 2 briefly illustrates the related works. Section 3 introduces the proposed sequential data mining approach. Section 4 presents two numerical examples to demonstrate the proposed process and results. Section 5 makes comparisons and offers a discussion. Conclusions are presented in Section 6.

2. RELATED WORKS

This paper divides related works into two subsections: sequential data mining, as well as group priority and recommendations, as follows:

2.1 Sequential data mining

Sequential data mining is a data mining method used for discovering frequent sequential patterns in a time-related database. There are broad applications in using sequential pattern mining, such as predicting the next prescribed medications [12], discovering access patterns in Weblogs [13, 14], analyzing biological sequences [15], and finding interesting knowledge from customer transactions [7]. Apriori-based algorithms [6] are widely adopted for sequential data mining; however, most suffer from the tedious workload of excessive computation because of candidate generation and multiple scanning of database processes.

In order to improve the efficiency of sequential data mining, Pei et al. [5] developed a PrefixSpan algorithm. Based on the PrefixSpan algorithm, Chen and Hu [16] constructed a CFR-PostfixSpan algorithm to find all the frequent sequential patterns by considering compactness, frequency, and recency. Shyur et al. [17] developed a P-PrefixSpan algorithm incorporating time-probability constraints to obtain fewer but more reliable patterns. In addition, Yen and Lee [7] constructed bit-string operations to efficiently find sequential patterns.

This study combines the advantages of bit-string operations and the PrefixSpan algorithm to increase the efficiency of sequential data mining.

2.2 Group priority and recommendation

Many studies have attempted to discover the group priority of alternatives, thereby recommending the next items that customers are most likely to buy in the near future.

For predictions and recommendations, several approaches [18] have been developed, including content-based filtering [19], collaborative filtering [20, 21], rule-based approaches [2, 22, 23] and hybrid approaches [18, 24]. Rule-based approaches discover rules from a large database collected over time. For instance, Mishra et al. [22] proposed a web recommendation system considering sequential information. A next-page recommendation for users is generated based on the analysis of the sequential page visits of users. The similarity upper approximation method is employed to efficiently form clusters for prediction. However, the number of top similar clusters must be given in advance, which is usually difficult to determine. In addition, the prediction accuracy can be improved. This study proposes a prediction and recommendation approach based on sequential data mining, which can be categorized as rule-based approaches, where total score and frequency of occurrence are considered as recommendation rules.

A variety of methods have been proposed to discover group priorities or group consensus. Based on the concept of Borda count [9], Zahid and Swart [8] introduced a Borda majority count method to explore the order of alternatives by calculating the sum of natural numbers for each alternative. Given a finite language of strictly ordered linguistic grades, g_p , where $g_1 > g_2 > \ldots > g_p > \ldots > g_z$, the corresponding score of these grades, denoted as g_p^*, can be assigned as $g_1^* = z\text{-}1$, $g_2^* = z\text{-}2$, …, $g_z^* = 0$, where $g_1^* = z\text{-}1$ because the priority of g_1 is higher than z-1 linguistic grades. The Borda majority count is easy to employ; nevertheless, users might not achieve a consensus on the final results.

In order to obtain group consensus results, Chen and Cheng [10] proposed a consensus mining approach to discover the maximum consensus sequences of alternatives. Ma [25] developed a consensus-based ranking approach based on a rank-tracking algorithm and an optimization model. However, the candidate generation and the exhaustive searching process are the major disadvantages. In order to overcome the problems mentioned above, Ma [26] proposed a novel graphical approach to discover group consensus preferences based on consensus mining and Gower plots. For group ranking problems involving different intensities of preference, Ma [27] then developed a new consensus-based approach stemming from the concept of consensus mining. Nevertheless, these consensus mining approaches cannot solve time-related problems.

This research integrates the advantages of bit-string operation, PrefixSpan algorithm, Borda majority count, and consensus mining into sequential data mining to more effectively and accurately predict time-related browsing behavior.

3. THE PROPOSED SEQUENTIAL DATA MINING APPROACH

This study proposes a prediction and recommendation approach for web browsing behavior by integrating the concepts of sequential data mining, bit-string operation [7], the PrefixSpan algorithm [5], and the Borda majority count [8].

Denote $A = \{a_1, a_2, \ldots, a_n\}$ as a set of n items and $U = \{u_1, u_2, \ldots, u_m\}$ as m users in the group. Let S_i be the user sequence of u_i, which is composed of a series of items ordered by increasing the transactions or visiting time. Denote $L(S_i)$ as the length of sequence S_i indicating the number of items in this sequence. For example, suppose there are 4 users and 7 items $\{a_1, a_2, \ldots, a_7\}$, denoted as Example 1; the selecting sequences of 4 users $\{u_1, \ldots, u_4\}$ are listed in Table 1 in which $S_1 = \{a_5, a_1, a_2, a_4, a_3, a_6\}$ and $L(S_1) = 6$.

A flowchart of the proposed recommendation approach is shown in Figure 1. The proposed approach includes two parts: the model construction and the recommendation parts. The model construction process is illustrated in the following six steps:

Step 1. Input users' sequences.
Step 2. Find 1-itemset sequences based on the given min_support.
Step 3. Build a bit-string coding matrix.
Step 4. Construct a bit-string mask matrix.
Step 5. Perform the projected matrix developing algorithm.
Step 6. Develop the recommendation order.

The recommendation process is described in the following three steps:
Step 1. Input an item or a sequence.
Step 2. Check the recommendation order.
Step 3. Output the recommendation order.

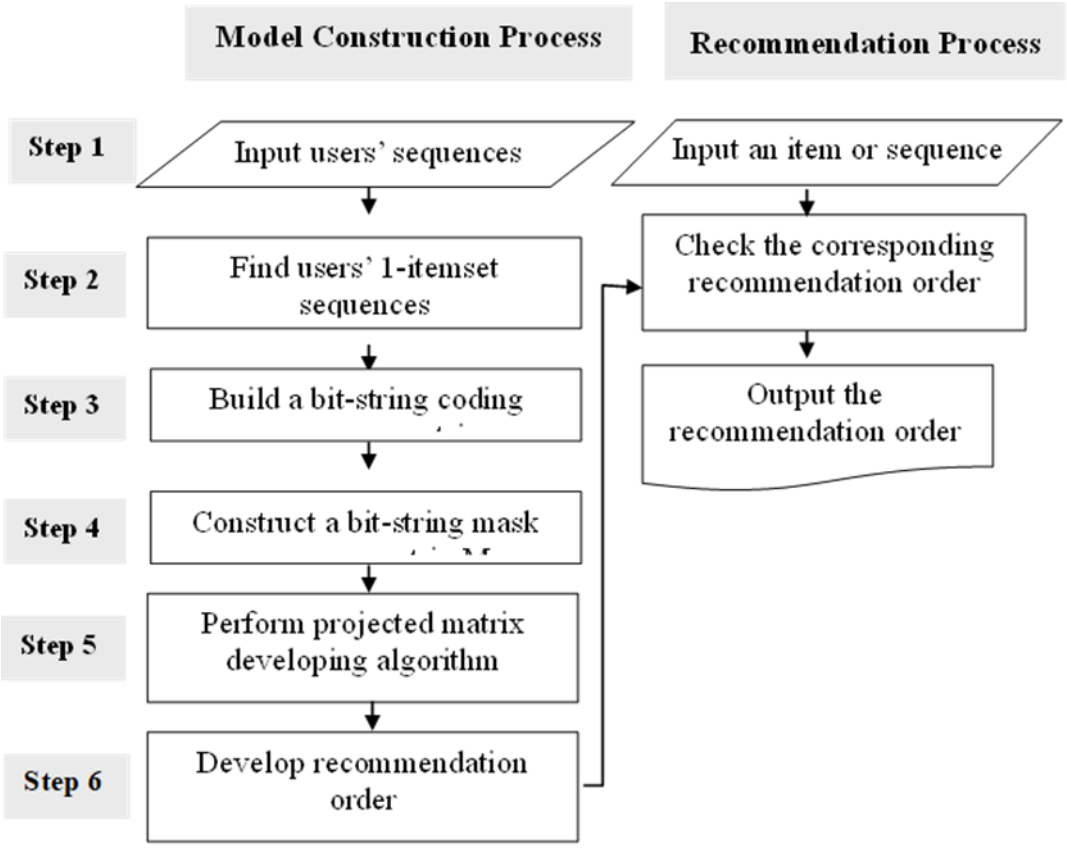

Figure 1. A flowchart of the proposed recommendation approach involving sequential information

The detailed descriptions of each step are illustrated below. Denote support(a_j) as the number of users' sequences containing a_j divided by the total number of users' sequences. Denote min_support as the minimum support specified by a decision-maker. If the support of an item a_j is greater than or equal to the min_support, then a_j is called a frequent item. For instance, the supports of a_1 and a_6 in Example 1 are 0.75 and 0.25, respectively. In this example, if min_support is given as 30%, the 1-itemset list, denoted as I^1, is $\{a_1, a_2, a_3, a_4, a_5\}$ where a_6 and a_7 are filtered out because they are not frequent items. Based on the 1-itemset list, we can derive users' 1-itemset sequences by filtering out infrequent items. Denote 1-itemset sequence for user u_i as S_i^1. Take Example 1 for instance: $S_1^1=\{a_5, a_1, a_2, a_4, a_3\}$. The 1-itemset sequences for each user are listed in Table 2. Let *MaxL* be the max value of $L(S_i^1)$ for all *i*. In Example 1, *MaxL* = 5.

This study employs the bit-string operation [7] to efficiently discover sequential patterns. First, users' 1-itemset sequences are mapped into a bit-string coding matrix containing bit-strings with equal length *MaxL*. For each item a_j in 1-itemset list I^1, if a user's 1-itemset sequence contains the item a_j in the q^{th} position, then the q^{th} bit of the bit-string is set to 1 and the remaining bits are set to 0; otherwise, all bits are set to 0. For example, for user u_1 with $S_1^1 = \{a_5, a_1, a_2, a_4, a_3\}$, the bit-string coding for item a_1 is "01000" because a_1 is in the second position of S_1^1; the bit-string for a_2 is "00100"

because a_2 is in the third position. The bit-string coding matrix is denoted as B, where element $B_{i,j}$ represents the bit-string coding of item a_j for user u_i.

Next, a bit-string mask is constructed to find subsequent items. When a bit-string coding $B_{i,j}$ is determined, it is scanned from left to right until a bit with value 1 is found. This bit and all bits prior to this bit are set to "0", and all bits posterior to this bit are set to "1" in the bit-string mask. For instance, for $B_{1,2}$ = "00100", the corresponding bit-string mask is "00011". The bit-string mask matrix is denoted as M, where each element $M_{i,j}$ stands for the bit-string mask of item a_j for user u_i.

The subsequent item can be found by performing a logical AND operation on bit-string $M_{i,j}$ and $B_{i,j}$. If the number of "1" in the resultant bit-string is not zero, then there is at least one a_k posterior to a_j in the user's 1-itemset sequence S_i^1. The position of "1" in the resultant bit-string indicates the order of subsequent items. If the position of "1" in the resultant bit-string is at the r^{th} position of "1" in $M_{i,j}$, then a_k is at the r^{th} position subsequent to a_j. If we want to find which items are subsequent to a_1 in user u_1's 1-itemset sequence S_1^1, we carry out an AND operation between $M_{1,1}$ and $B_{1,k}$ for all $1 \leq k \leq MaxL$. For instance, in Example 1, $M_{1,1}$ = "00111" and $B_{1,3}$="00001", $M_{1,1}$ AND $B_{1,3}$ = "00001"; because the position of "1" in the resultant bit-string is at the third position of the "1"s in $M_{1,1}$, we can find a_3 at the third position subsequent to a_1 in S_1^1. All subsequent items of a_1 in S_1^1 can be found by the following AND operations: $M_{1,1}$ AND $B_{1,1}$ = "00000", $M_{1,1}$ AND $B_{1,2}$ = "00100", $M_{1,1}$ AND $B_{1,3}$ = "00001", $M_{1,1}$ AND $B_{1,4}$ = "00010", $M_{1,1}$ AND $B_{1,5}$ = "00000". We can find that a_2, a_4, and a_3 are at the first, second, and third positions after a_1, respectively.

From the concept of Borda majority count [8], this study posits that if the position of "1" in the resultant bit-string is found at the r^{th} position of "1" in the bit-string mask, then the corresponding score (priority value) is set as *MaxL*-*r*; otherwise, the score is set as 0. The range of *r* is between 1 and *MaxL* -1. The smaller the *r* value, the higher the corresponding score yields. This means that if two items are after a specific item a_j, the item with a smaller *r* value is closer to a_j with a higher priority value. Take Example 1 as an example; for a user's 1-itemset sequence S_1^1, the priority value of a_2 corresponding to a_1 is set as 4 because *MaxL* = 5 and *r* = 1. Similarly, the priority values of a_4 and a_3 corresponding to a_1 are set as 3 and 2, respectively.

Next, this study constructs a projected matrix developing algorithm based on the concept of the PrefixSpan algorithm [5], which divides the sequence database into several projected databases and examines local frequent patterns in each projected database to increase the computational efficiency of the sequential data mining. Each item in 1-itemset list I^1 is treated as a prefix item. Several projected matrices (PM) are generated based on different prefix items, and then local frequent relationships are examined in each projected matrix. Denote PM_j as a projected matrix with the prefix item a_j. For each a_j, the corresponding PM_j can be formed by the following algorithm.

Projected matrix developing algorithm

INPUT M and B
FOR i = 1 to n
{ FOR k = 1 to $MaxL$
{ IF the position of "1" in ($M_{i,j}$ AND $B_{i,k}$) is at the r^{th} position of "1"s in $M_{i,j}$,
THEN $PM_j(i, k) = MaxL - r$;
ELSE $PM_j(i, k) = 0$
}
}
OUTPUT PM

The score of a_k corresponding to a_j in PM_j is defined as the sum of $PM_j(i, k)$ for all users i. The total score is employed to decide the priorities of items to be recommended. It is worth noting that if an alternative a_j appears more than once in a user's 1-itemset sequence S_i^1, only the first a_j is treated as a prefix item in this study.

4. NUMERICAL EXAMPLES

Two examples are illustrated here. The first example is used to demonstrate the whole process of the proposed approach. The second example is adopted to make comparisons with other methods.

Example 1

Suppose there are seven items $\{a_1, a_2, \ldots, a_7\}$ in a store for sale; assume each customer buys one item at the same time. The purchasing sequences of four customers $\{u_1, \ldots, u_4\}$ are listed in Table 1. Given min_support = 30%, the 1-itemset list $I^1 = \{a_1, a_2, a_3, a_4, a_5\}$ and the corresponding 1-itemset sequences are listed in Table 2. The bit-string coding matrix B is listed in Table 3, and the bit-string mask matrix is listed in Table 4.

Based on the projected matrix developing algorithm, $PM_1, \ldots, PM_5$ are listed in Tables 5(a)~(e), respectively. The second row from the bottom of Tables 5(a)~(e) represents the frequency of occurrence (freq), and the bottom row shows the total score of subsequent items. The "freq" is used to judge whether an item is frequent, and the total score is adopted to determine the priorities of items.

Take PM_1 in Example 1 as an example; the total scores of a_1, a_2, a_3, a_4, a_5 are 0, 5, 8, 9, 4, respectively. This indicates that a_4 yields the highest priority after a_1. The remaining order is a_3, a_2, a_5 and a_1. That is, for a customer with the last bought item a_1, the recommendation order for the next item is a_4, a_3, and a_2. It is worth noting that items a_5 and a_1 with occurrence frequency less than 2 are removed from the recommendation list because they are not frequent items (min_support =30%). The recommendation order of Example 1 is listed in Table 6.

Table 1. Users' sequences of Example 1

	Users' sequences
S_1	$a_5, a_1, a_2, a_4, a_3, a_6$
S_2	a_1, a_3, a_4
S_3	a_2, a_4, a_3, a_7, a_2
S_4	a_1, a_5, a_4, a_3, a_2

Table 2. Users' 1-itemset sequences of Example 1

	Users' 1- itemset sequences
S_1^1	a_5, a_1, a_2, a_4, a_3
S_2^1	a_1, a_3, a_4
S_3^1	a_2, a_4, a_3, a_2
S_4^1	a_1, a_5, a_4, a_3, a_2

Table 3. The bit-string coding matrix ($B_{i,j}$) of Example 1

	a_1	a_2	a_3	a_4	a_5
u_1	01000	00100	00001	00010	10000
u_2	10000	00000	01000	00100	00000
u_3	00000	10010	00100	01000	00000
u_4	10000	00001	00010	00100	01000

Table 4. The bit-string mask matrix ($M_{i,j}$) of Example 1

	a_1	a_2	a_3	a_4	a_5
u_1	00111	00011	00000	00001	01111
u_2	01111	00000	00111	00011	00000
u_3	00000	01111	00011	00111	00000
u_4	01111	00000	00001	00011	00111

Table 5. The projected matrices of Example 1

(a) PM_1

	a_1	a_2	a_3	a_4	a_5
u_1	0	4	2	3	0
u_2	0	0	4	3	0
u_3	0	0	0	0	0
u_4	0	1	2	3	4
freq	0	2	3	3	1
total score	0	5	8	9	4

(b) PM_2

	a_1	a_2	a_3	a_4	a_5
u_1	0	0	3	4	0
u_2	0	0	0	0	0
u_3	0	2	3	4	0
u_4	0	0	0	0	0
freq	0	1	2	2	0
total score	0	2	6	8	0

(c) PM_3

	a_1	a_2	a_3	a_4	a_5
u_1	0	0	0	0	0
u_2	0	0	0	4	0
u_3	0	4	0	0	0
u_4	0	4	0	0	0
freq	0	2	0	1	0
total score	0	8	0	4	0

(d) PM_4

	a_1	a_2	a_3	a_4	a_5
u_1	0	0	4	0	0
u_2	0	0	0	0	0
u_3	0	3	4	0	0
u_4	0	3	4	0	0
freq	0	2	3	0	0
total score	0	6	12	0	0

(e) PM_5

	a_1	a_2	a_3	a_4	a_5
u_1	4	3	1	2	0
u_2	0	0	0	0	0
u_3	0	0	0	0	0
u_4	0	2	3	4	0
freq	1	2	2	2	0
total score	4	5	4	6	0

Table 6. The recommendation order for each prefix item of Example 1

Prefix item	Recommendation order
a_1	a_4, a_3, a_2
a_2	a_4, a_3
a_3	a_2
a_4	a_3, a_2
a_5	a_4, a_2, a_3

Example 2

The second example, referred to in Mishra et al. [22], is the MSNBC web navigation data set collected from the UCI dataset repository. The MSNBC dataset is a benchmark dataset consisting of weblogs from msnbc.com and msn.com for one day. Each weblog is represented as a sequence of page views of a user. According to Mishra et al. [22], only user sessions whose length is 6 are considered because the average length of the web user session is 5.7. There are 17 categories in the data set including "front-page", "news", "tech", …, "MSN-sports", which are transferred into 1, 2, 3, …, 17, respectively. A sample of 20 users' sequences from the MSNBC dataset is listed in Table 7.

To make a comparison with the results of Mishra et al. [22], a data set whose size is 5000 sequences is taken for the model construction process (training), and a data set with 2000 sequences is used for the recommendation process (testing). The concept of ten-fold cross-validation is employed for validation. For example, for testing subsample size = 100, 100 subsamples are randomly sampled from the 2000 original testing dataset, and the process is then repeated 10 times. The 10 results from the folds can then be averaged to produce a single estimation.

Min_support is a parameter supplied to the Apriori algorithm in order to prune candidate rules by specifying a minimum lower bound for the Support measure of resulting association rules. The setting value of min_support may vary according to different products or industry characteristics. If the min_support value is too big, nothing might be found in a database, whereas a small min-support might generate many uninteresting association rules. In order to make comparisons, two commonly used min_support values (1% and 0.1%) are tested. Given min_support = 1%, the 1-

item set list I^1 = {1, 2, …, 15} and the corresponding recommendation table are listed in Table 8. Given min_support = 0.1%, the 1-item set list I^1 = {1, 2, …, 17} and the corresponding recommendation table are listed in Table 9. This study uses accuracy as the metric to evaluate the recommendation system, rather than a statistical significance test, because accuracy is a simple and popular metric in the field of data mining, regardless of data distribution. Accuracy is defined as the ratio of the number of correct recommendations to the number of total recommendations.

Table 7. List of 20 training sequences of Example 2

No	Sequences	No	Sequences	No	Sequences
1	6, 11, 6, 12, 12, 12	**8**	14, 14, 14, 14, 14, 14	**15**	1, 1, 14, 14, 14, 14
2	13, 13, 14, 14, 14, 14	**9**	8, 9, 9, 4, 4, 4	**16**	1, 1, 1, 1, 1, 1
3	4, 4, 4, 4, 4, 4	**10**	6, 6, 6, 6, 7, 6	**17**	1, 1, 1, 1, 14, 1
4	1, 1, 1, 11, 1, 1	**11**	6, 13, 14, 14, 14, 14	**18**	1, 2, 17, 2, 1, 1
5	2, 2, 2, 2, 10, 2	**12**	9, 13, 13, 9, 9, 9	**19**	1, 7, 7, 7, 7, 1
6	1, 1, 1, 1, 10, 1	**13**	1, 1, 7, 7, 7, 7	**20**	6, 7, 6, 7, 6, 6
7	6, 7, 7, 7, 6, 6	**14**	14, 14, 2, 2, 14, 14		

Table 8. The recommendation table of Example 2 with Min_support = 0.01

Prefix	Order of recommendation	Prefix	Order of recommendation
1	2, 12, 14, 4, 11, 7, 1, 6, 3, 10, 5, 15, 8	9	9, 7, 4, 1, 3, 2
2	2, 3, 4, 12, 14, 6, 10, 7, 1	10	10, 2
3	3, 2, 4	11	11, 2
4	4, 7, 2, 14, 6	12	12, 2, 3, 14, 4, 6
5		13	14, 13, 7
6	6, 7, 2, 15, 4, 9, 10, 3	14	14
7	7, 4, 1	15	15
8	8		

Table 9. The recommendation table of Example 2 with Min_support = 0.001

Prefix	Order of recommendation	Prefix	Order of recommendation
1	2, 12, 14, 4, 7, 1, 6, 11, 3, 10, 5, 8, 15, 9, 13, 16, 17	10	10, 2, 6, 3, 11, 4, 14, 1, 7, 15, 5, 9
2	2, 3, 4, 12, 6, 14, 10, 7, 1, 11, 5, 15, 8, 9, 13, 17	11	11, 2, 12, 14, 3, 4, 6, 10, 17, 1, 5, 8, 15, 9
3	3, 2, 4, 12, 9, 14, 10, 1, 6, 11, 7, 5, 15, 8, 13, 17	12	12, 2, 3, 14, 9, 4, 1, 6, 10, 11, 7, 5, 13, 8, 15, 17
4	4, 7, 2, 14, 9, 3, 6, 1, 12, 11, 13, 10, 8, 5, 15	13	14, 13, 7, 9, 1, 4, 8, 3, 6, 12, 11, 2
5	5, 2, 15, 3, 4, 6, 1, 11, 14, 9	14	14, 2, 1, 3, 13, 12, 4, 7, 6, 11, 10, 9, 8, 5
6	6, 7, 2, 15, 4, 10, 9, 3, 14, 1, 8, 12, 11, 13, 5	15	15, 7, 6, 10, 2, 11, 5, 4, 3, 12, 8, 9, 1
7	7, 4, 1, 2, 14, 3, 13, 6, 8, 10, 15, 12, 9, 11	16	16
8	8, 2, 4, 7, 6, 13, 9, 1, 12, 3, 11, 15, 10, 5	17	17, 2, 4, 11, 7, 6, 14
9	9, 4, 1, 7, 3, 2, 13, 12, 6, 8, 14, 11, 5, 10, 15		

Table 10. The prediction accuracy with different testing sizes (Number of recommendations = 1)

Size of testing samples	The proposed approach		
	Min_support = 1%	Min_support = 0.1%	Average
100	52.50%	52.80%	52.65%
500	51.78%	51.96%	51.87%
1000	52.62%	52.82%	52.72%
2000	52.40%	52.63%	52.52%
Average	52.33%	52.55%	52.44%

Four different sizes of testing samples are examined: 100, 500, 1000, and 2000. The accuracy of different testing sizes is listed in Table 10 in which the average recommendation accuracy is 52.44%. Different sizes of testing samples and different settings of min_support values yield similar accuracy values. The accuracy of different numbers of recommendation items from 1 to 3 with 1000 and 2000 testing samples are listed in Tables 11 and 12, respectively. Different sizes of testing samples still yield similar results. In Table 12, the average accuracy for 1, 2 and 3 recommendation items are 52.52%, 61.48% and 64.78%, respectively.

Table 11. The prediction accuracy with different recommendation numbers (Testing size = 1000)

Number of recommendation items	The proposed approach		
	Min_support = 1%	Min_support = 0.1%	Average
1	52.62%	52.82%	52.72%
2	60.07%	61.69%	60.88%
3	63.08%	65.50%	64.29%
Average	58.59%	60.00%	59.30%

Table 12. The prediction accuracy with different recommendation numbers (Testing size = 2000)

Number of recommendation items	The proposed approach		
	Min_support = 1%	Min_support = 0.1%	Average
1	52.40%	52.63%	52.52%
2	60.75%	62.21%	61.48%
3	63.69%	65.87%	64.78%
Average	58.95%	60.24%	58.59%

5. COMPARISONS AND DISCUSSION

The results of three approaches: Mishra et al. [22], random method, and the proposed approach are compared and discussed herein.

Taken from Mishra et al. [22], the given number of clusters = 20; the accuracies for five different testing samples of the MSNBC dataset are listed in Table 13. The average accuracy of 1, 2, and 3 recommendation items are 12.9%, 19.76%, and 28.86%, respectively. Take Example 2 with 2000 testing as a sample, the average accuracy of Mishra et al. [22], the random method, and the proposed approach are shown in Table 14 and Figure 2. In Figure 2, the *x*-coordinate axis represents the recommendation quantity, while the *y*-coordinate axis represents the prediction accuracy. As depicted in Figure 2, the prediction accuracy rises when the number of recommendation items increases for all three methods; moreover, the proposed approach yields the highest prediction accuracy.

Comparisons among the three approaches are listed in Table 15. The major concepts of the three approaches are soft clustering and singular value decomposition for Mishra et al. [22], simple guess for the random method, sequential data mining,

Borda majority count, bit-string operation, and the PrefixSpan algorithm for the proposed approach. The prediction accuracy of the proposed approach is much higher than those of Mishra et al. [22] and the random method. The approach of Mishra et al. [22] requires decision-makers to set the number of clusters in advance; however, different settings may result in different results. Only the proposed approach can provide flexibility for decision-makers to adjust the preferred minimum support level.

The computational time of the proposed approach mostly depends on the projected matrix developing algorithm with time complexity $O(nm^2)$, where n is the number of records or users, and m is the number of items. Although the computational time seems to increase greatly when the number of items grows, the proposed bit-string operation can improve the limitation by simple binary operations. The computational time of Example 2 by the proposed approach is less than 5 seconds. The computational environment in this study is a personal computer with Intel Core i7-6700 CPU and 8 GB RAM, and the programming language used is Java.

Table 13. The prediction accuracy of Mishra et al. [22]

Number of recommendation items	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Average
1	11.76%	12.50%	9.38%	17.24%	13.64%	12.90%
2	23.53%	18.75%	18.75%	24.14%	13.64%	19.76%
3	24.91%	37.50%	28.13%	31.03%	22.73%	28.86%
Average	20.07%	22.92%	18.75%	21.14%	16.67%	20.51%

Table 14. Results of different approaches (Testing size = 2000)

Number of recommendation items	Mishra et al. [22]	Random	The proposed approach
1	12.90%	6.19%	52.52%
2	19.76%	10.39%	61.48%
3	28.86%	13.02%	64.78%
Average	20.51%	9.87%	59.59%

Table 15. Comparisons of different approaches

	Mishra et al. [22]	**Random**	**The proposed approach**
Major concepts	Soft clustering Singular value decomposition	Guess	Sequential data mining Borda majority count Bit-string operation PrefixSpan algorithm.
Recommendation Accuracy	Medium	Low	High
Predefined number of clusters	Yes	No	No
Flexibility for adjusting minimum support level	No	No	Yes

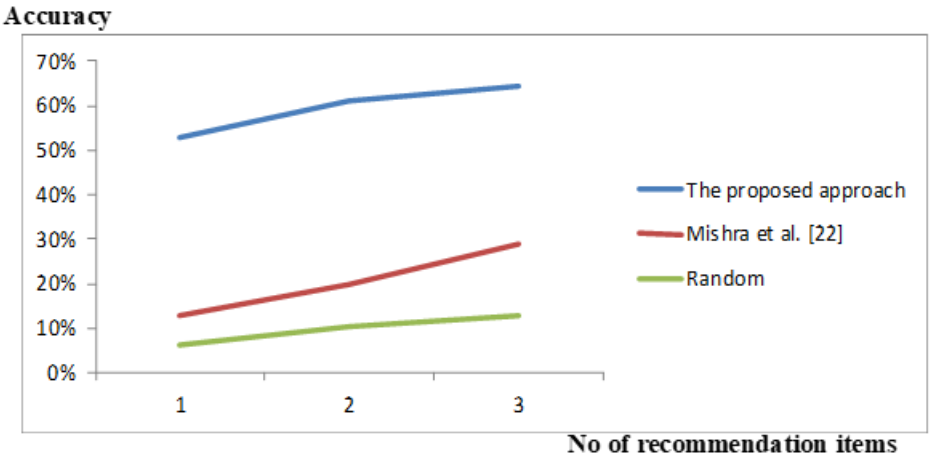

Figure 2. Results of different approaches (testing size = 2000)

6. CONCLUSIONS

This study proposes a prediction and recommendation approach for web browsing behavior by integrating the concept of sequential data mining, Borda majority count, bit-string operation, and the PrefixSpan algorithm. First, users' 1-itemset sequences are mapped into a bit-string coding matrix. The subsequent item can be found by performing a logical AND operation on a bit-string coding matrix and mask matrix. The projected matrix developing algorithm was developed to construct the projected matrix for each prefix item. By calculating the total score and frequency of subsequent items for each item, the recommendation order can be achieved.

The proposed approach can be widely applied to solve real-world problems, such

as recommending the next video for renting, next commodity for promotion, and next webpage for browsing. In the era of e-commerce, there is a great demand for a reliable recommendation system to help companies recommend the next product that customers may soon buy. The proposed approach can explore customers' sequential purchasing patterns, thereby helping companies understand customers' buying behavior and recommend suitable products.

This research employs the concept of ensemble methods that combine multiple models to produce improved results. Ensemble methods usually yield better solutions than a single model would in the field of machine learning. The major contributions of this research include: (1) proposing a next item recommendation approach with higher recommendation accuracy and computational efficiency. (2) providing decision-makers with the flexibility to adjust the minimum support level, which can be used for different products or industry characteristics.

One of the main limitations of this research is that it is difficult to obtain real datasets from businesses, so only a benchmark dataset is used for validation and comparisons. In addition, for parameter settings, only two commonly used min_support values are tested and compared. Further research can address improving these limitations or developing next-group recommendation methods.

Acknowledgement

This work was supported by the Ministry of Science and Technology of the Republic of China [grant number: MOST 108-2410-H-239-011-MY3].

7. REFERENCES

[1] R. Burke, "Hybrid recommender systems: survey and experiments," *User Modeling and User-Adapted Interaction*, Vol. 12, pp. 331-370, 2001.

[2] N. A. Desai, and A. Ganatra, "Buying scenario and recommendation of purchase by constraint based sequential pattern mining from time stamp based sequential dataset," *Procedia Computer Science*, Vol. 45, pp. 166-175, 2015.

[3] Y. H. Hu, F. Wu, and Y. J. Liao, "An efficient tree-based algorithm for mining sequential patterns with multiple minimum supports," *The Journal of Systems and Software*, Vol. 86, No. 5, pp. 1224-1238, 2013.

[4] Z. Zhang, Y. Liu, W. Ding, W. Huang, Q. Su, and P. Chen, "Proposing a new friend recommendation method, FRUTAI, to enhance social media providers' performance," *Decision Support Systems*, Vol. 79, pp. 46-54, 2015.

[5] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. C. Hsu, "PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern

growth," in Proc. of Int. Conf. on Data Engineering, Heidelberg, Germany, 2001, pp. 215-224.

[6] R. Agrawal, and R. Srikant, "Mining sequential pattern," in Proc. 11th Int. Conf. on Data Engineering, Taipei, Taiwan, 1995, pp. 3–14.

[7] S. J. Yen, and Y. S. Lee, "An efficient data mining approach for discovering interesting knowledge from customer transactions," *Expert Systems with Applications*, Vol. 230, No. 4, pp. 650-657, 2006.

[8] M. A. Zahid, and H. D. Swart, "The Borda majority count," *Information Sciences*, Vol. 295, pp. 429-440, 2015.

[9] J. C. Borda, "Mémoire sur les élections au scrutiny," Histoire de l'Academie Royale de Sciences, Paris. Available, in: I. McLean, A. Urken (Eds.) (1995) Classics of Social Choice, Ann Arbor: University of Michigan Press, 1981.

[10] Y. L. Chen, and L. C. Cheng, "Mining maximum consensus sequences from group ranking data," *European Journal of Operational Research*, Vol. 198, pp. 241-251, 2009.

[11] R. Agrawal, and R. Srikant, "Fast algorithms for mining association rules," in Proc 20th Int. Conf. on VLDB, Santiago de Chile, Chile, 1994, pp. 487-499.

[12] A. P. Wright, A. T. Wright, A. B. McCoy, and D. F. Sittig, "The use of sequential pattern mining to predict next prescribed medications," *Journal of Biomedical Informatics*, Vol. 53, pp. 73-80, 2015.

[13] R. Mishra, and P. Kumar, "Clustering web logs using similarity upper approximation with different similarity measures," *International Journal of Machine Learning and Computing*, Vol. 2, No. 3, pp. 219-221, 2012.

[14] P. SenKul, and S. Salin, "Improving pattern quality in web usage mining by using semantic information," *Knowledge and Information Systems*, Vol. 30, No. 3, pp. 527-541, 2012.

[15] V. C. C. Liao, and M. S. Chen, "Dfsp: a depth-first spelling algorithm for sequential pattern mining of biological sequence," *Knowledge and Information Systems*, Vol. 38, No. 3, pp. 623-639, 2014.

[16] Y. L. Chen, and Y. H. Hu, "Constraint-based sequential pattern mining: the consideration of recency and compactness," *Decision Support Systems*, Vol. 42, pp. 1203-1215, 2006.

[17] H. J. Shyur, C. Jou, and K. Chang, "A data mining approach to discovering

reliable sequential patterns," *The Journal of Systems and Software*, Vol. 86, pp. 2196-2203, 2013.

[18] K. Choi, D. Yoo, G. Kim, and Y. Suh, "A hybrid online-product recommendation system: combining implicit rating-based collaborative filtering and sequential pattern analysis," *Electronic Commerce Research and Applications*, Vol. 11, pp. 309-317, 2012.

[19] M. Pazzani, and D. Billsus, "Learning and revising user profile: the identification of interesting web sites," *Machine Learning*, Vol. 27, No. 3, pp. 313-331, 1997.

[20] K. W. Cheung, J. T. Kwok, M. H. Law, and K. C. Tsui, "Mining customer product ratings for personalized marketing," *Decision Support Systems*, Vol. 35, No. 2, pp. 231-243, 2003.

[21] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H. P. Kriegel, "Probabilistic memory-based collaborative filtering," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 1, pp. 56-69, 2004.

[22] R. Mishra, P. Kumar, and B. Bhasker, "A web recommendation system considering sequential information," *Decision Support Systems*, Vol. 75, pp. 1-10, 2015.

[23] Y. Wang, W. Dai, and Y. Yuan, "Website browsing aid: a navigation graph-based recommendation system," *Decision Support Systems*, Vol. 45, No. 3, pp. 387-400, 2008.

[24] M. Salehi, and I. N. Kamalabadi, "Hybrid recommendation approach for learning material based on sequential pattern of the accessed material and the learner's preference tree," *Knowledge-Based Systems*, Vol. 48, pp. 57-69, 2013.

[25] L. C. Ma, "A new group ranking approach for ordinal preferences based on group maximum consensus sequences," *European Journal of Operational Research*, Vol. 251, No. 1, pp. 171-181, 2016.

[26] L. C. Ma, "Discovering consensus preferences visually based on Gower plots," *International Journal of Information Technology & Decision Making*, Vol. 17, No. 3, pp. 741–761, 2018.

[27] L. C. Ma, "A new consensus mining approach to group ranking problems involving different intensities of preferences," *Computers & Industrial Engineering*, Vol. 131, pp. 320-326, 2019.