Application of Matrix Clustering to Web Log Analysis and Access Prediction

Shigeru Oyanagi, Kazuto Kubota and Akihiko Nakase
Parallel Application TOSHIBA Laboratory, Real World Computing Partnership
R&D Center, Toshiba Corp., Komukai Toshiba-cho, Saiwai-ku, Kawasaki, 212-8582, Japan
{oyanagi, kazuto, nakase}@isl.rdc.toshiba.co.jp

Abstract

Matrix clustering is a new data mining method which extracts a dense sub-matrix from a large sparse binary matrix. We propose an efficient algorithm named the ping-pong algorithm which enables real-time mining of a large sparse matrix.

This article describes the application of matrix clustering to Web usage mining. Matrix clustering can be applied to Web access log analysis by representing relationships between pages and users in a binary matrix. An experiment with a practical WWW access log shows that page clusters can be extracted by applying matrix clustering. The extracted page clusters are compared with those obtained by association rule mining. The result shows that matrix clustering is more powerful in finding various types of page clusters.

The page clusters extracted by matrix clustering can be applied to web access prediction. We have also compared the matrix clustering with association rule mining and sequence pattern mining with respect to access prediction. The result shows that matrix clustering has a higher hit rate than these methods when the session length is long.

1 Introduction

Rapid increase of electronic commerce on the Web has been shifting the strategy for marketing toward mass customization. Now, personalization techniques such as recommendation are getting attracted on various EC sites [5].

Basic approaches for customization can be classified into two categories: web usage mining approaches and collaborative filtering approaches. Web usage mining applies data mining techniques to the WWW access log to discover Web usage patterns for customization[7]. Associative rule mining [3] and sequence pattern mining [4]are typical mining methods used for Web usage mining. On the other hand, collaborative filtering approaches [5][6] use user profiles which include preferences for individual products by individual consumers. Collaborative filtering is powerful when the user profile is available, whereas Web usage mining can be applied to a wider range.

We have proposed a new mining method named matrix clustering[12]. Matrix clustering extracts a dense sub-matrix from the base sparse matrix. It can be applied to a Web access log by representing the relationship between users and Web pages in a binary matrix. The result of matrix clustering can be explained as a set of users and a set of Web pages related to each other. The basic representation is similar to the collaborative filtering, however matrix clustering restricts the value of matrix to a binary value. Hence, it is easier to get the matrix value automatically from user's action, which makes it easy to adapt matrix clustering to a wider application area than multi-valued collaborative clustering.

We have also proposed an efficient algorithm named the ping-pong algorithm for matrix clustering. The ping-pong algorithm is especially efficient when the matrix is large and sparse. Our experiment shows that matrix clustering is efficient enough for real-time processing of practical size matrix.

This paper proposes an application of matrix clustering to Web usage analysis and access prediction. It shows that efficiency and wide applicability of matrix clustering can be a suitable technique for personalization on EC sites.

This paper consists of the following sections. Section two defines matrix clustering and the ping-pong algorithm. Section three explains the application of matrix clustering to a Web access log. The extracted page clusters are compared with those by association rule mining. Section four explains the

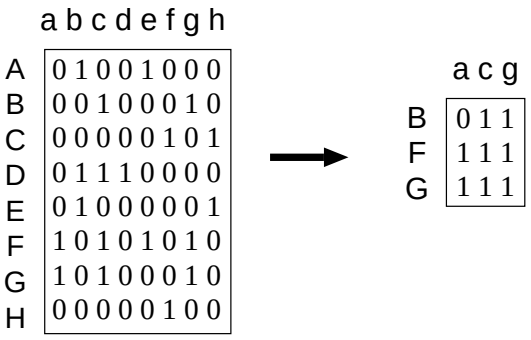

Figure 1: Extraction of dense sub-matrix

application of extracted page clusters to Web access prediction. The result is compared with those of association rule mining and sequence pattern mining.

2 Matrix Clustering

2.1 Definition of Matrix Clustering

Matrix clustering is proposed for CRM(customer relationship management)[12]. A target matrix is shown in Fig.1. In Fig.1, row represents customers and column represents items. Matrix element Aij represents whether customer-i bought an item-j or not. Namely, Aij = 1 means that a customer-i has bought an item-j, and Aij = 0 means that a customer-i has not bought an item-j. Generally, matrix Aij is a large-scale sparse matrix.

Since the order of customers and items is not meaningful, rows and columns can be exchanged arbitrarily. Fig.1 shows the extracted dense sub-matrix from the given base matrix. As shown in Fig.1, a dense sub-matrix can be extracted which consists of customers B, F and G and items a, c and g. This sub-matrix can be explained as follows. Customers B, F and G have a common feature with respect to the purchase of items a, c and g. Items a, c and g have a common feature with respect to the purchase by the customers B, F and G. There is a possibility that customer B will buy the item a in future.

This hypothesis should be associated with statistical meaning. Here, we introduce the concepts of support and confidence which are proposed in the Apriori algorithm [3] for finding association rules. In the matrix clustering, support is defined as an area of extracted sub-matrix, and confidence is defined as the density of extracted sub-matrix. Matrix clustering is defined as the following two problems with these definitions.

1. find a sub-matrix of maximum density whose area is greater than a user specified value (support).
2. find a sub-matrix of maximum area whose density is greater than a user specified value (confidence).

2.2 Ping-pong Algorithm

A naive algorithm to exchange rows and columns iteratively takes a lot of computation when the matrix size is large. Here, we propose a new fast algorithm which is intended to reduce the execution time by utilizing the sparseness of a matrix. The ping-pong algorithm uses marker propagation. Marker propagation is generally defined as a computation model to represent a unit of processing as a node and a relation between nodes as a link. It activates processing by passing a marker from an activated node through a link successively. In the ping-pong algorithm, rows and columns are represented as nodes. When a matrix element is 1, the correspondent row and column are connected by a bi-directional link. When a node receives a marker, it is accumulated, and the total count of the received markers is used for pruning. The structure of the ping-pong algorithm is shown below.

```
While (convergence) {
      row_to_col();
      prune_col();
      col_to_row();
      prune_row();
}
```

The algorithm iterates marker propagation between rows and columns until the state where the activated rows and columns are not changed. The processing of the ping-pong algorithm is explained with Fig.2. The ping-pong algorithm starts with the specification of the starting row as row_B. At first, row_B is activated and it sends a marker to the linked columns (c, g). Next, pruning of columns is performed. Since the counts of markers are 1 for all columns (c, g), all of them are activated without pruning. Then, activated columns (c, g) propagate markers to linked rows. For example, column_c sends a marker to rows (B,D,F,G), and column_g sends a marker to rows (B,F,G). Then, each row calculates the count of received markers. For example, row_B, row_F and row_G have received 2 markers at this step, but row_D has received 1 marker. Then the pruning of rows is performed. In this case, row_D is pruned and the other rows(B,F,G) are activated. Next, rows(B,F,G) send markers

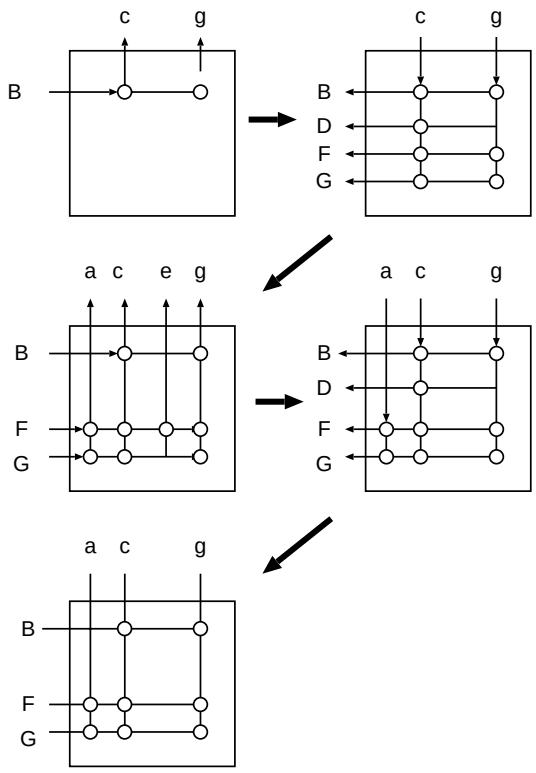

Figure 2: Ping-pong algorithm

and columns(a,c,e,g) receive markers. Since column_e receives only 1 marker, column_e is pruned and the other columns(a,c,g) are activated. Finally, rows(B,F,G) and columns(a,c,g) are activated to form the result.

Here we notice that area and density of sub-matrix which consists of the activated rows and columns at each step can be calculated easily. For example, area of sub-matrix at final stage can be calculated by 3 x 3 = 9, because the count of activated rows is 3 and the count of activated columns is 3. Density can be calculated by dividing the total count of received markers at activated nodes by the area. The density of sub-matrix at the final stage is 8 / 9 = 0.89.

The important issue in this algorithm is the pruning strategy. Pruning is performed by comparing the count of received markers with a threshold value. If the threshold value is low, then the resulting sub-matrix is large but the density is low. On the other hand if the threshold value is high, then the density of the resulting sub-matrix is high but the area is small. We propose to calculate the threshold value at each step from a user specified value. Namely, when the problem is to find a sub-matrix with maximum density whose area is greater than the user specified value (support), the threshold value is calculated at each step by using this support value. When the problem is to find a sub-matrix with maximum area whose density is greater than the user specified value (confidence), the threshold value is calculated at each step by using this confidence value. In addition, constraints on the form of sub-matrix (minimum rows and columns) can be represented in the calculation of the threshold value.

Next, we will discuss the ping-pong algorithm from the viewpoint of a database. The matrix of customers and items can be represented as a relation whose attributes correspond to customers and items. The ping-pong algorithm corresponds to iterate the execution of SELF JOIN and GROUP BY operations to the relation (customer, item) with pruning. This processing is quite similar to the Apriori algorithm[3] which also uses SELF JOIN and GROUP BY operations to a relation (transaction#, item) with pruning. The major difference concerns the JOIN term. The JOIN term of the Apriori algorithm is always at the attribute of transaction#, but that of the ping-pong algorithm changes each time between attributes of customer and item.

2.3 Implementation

We have implemented the ping-pong algorithm. The program specifies parameters listed below prior to the execution.

- Execution mode : to maximize area or to maximize density
- Minimum support : specify minimum area in the case of maximize density
- Minimum confidence : specify minimum density in the case of maximize area
- Minimum rows / minimum columns : specify restriction to the form of sub-matrix
- Start rows / columns : specify rows or columns as the start point of the algorithm
- Inhibit rows / columns : specify rows or columns that should be excluded from the result

In general, the size of matrix should be large and sparse, hence the data structure to store the matrix is important. Major functions to access the matrix in the ping-pong algorithm is to find locations of elements with value 1 in the specified row or column.

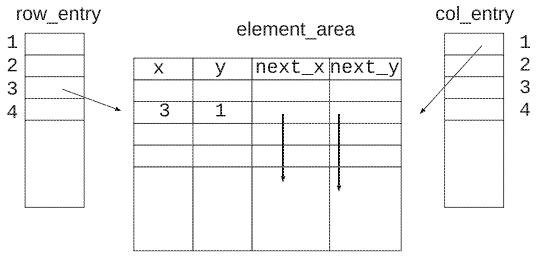

Figure 3: Data structure

We have implemented data structure for the matrix as shown in Fig.3.

In Fig.3, each row and column has an entry (row_entry, col_entry). Matrix elements with value 1 are stored in a large array (element_area), which consists of the location of the element (x,y), a pointer to the next element in the same row(next_row), and a pointer to the next element in the same column(next_col). This data structure makes it possible to access matrix elements of value 1 in the specified row or column efficiently through the pointers. The required memory size to store the data structure is only dependent on the total number of matrix elements with value 1, which means that it is suitable for storing large sparse matrices.

3 Application to Web access log analysis

3.1 Extraction of clusters

We have applied matrix clustering to Web access log analysis. The log data contain cookies to identify users and URLs. These cookies and URLs are extracted to form a matrix, namely URLs correspond to rows and cookies correspond to columns. The number of rows is 480, the number of columns is 1,223, and the number of matrix elements with value 1 is 6,936. The density of the matrix is 1.4%. We iterated the ping-pong algorithm for each row as a start row to extract clusters. The parameters are set to find the maximum area with density greater than 80% with minimum rows and columns to 3. Some of the interesting results are given below. In addition, Fig.4 shows the pattern of cluster B.

- Clusters with large rows and columns

cluster-name	area(row x col)	density
A	38 x 25 = 950	86%
B	24 x 57 = 1392	86%
C	20 x 46 = 920	80%

- Clusters with large columns

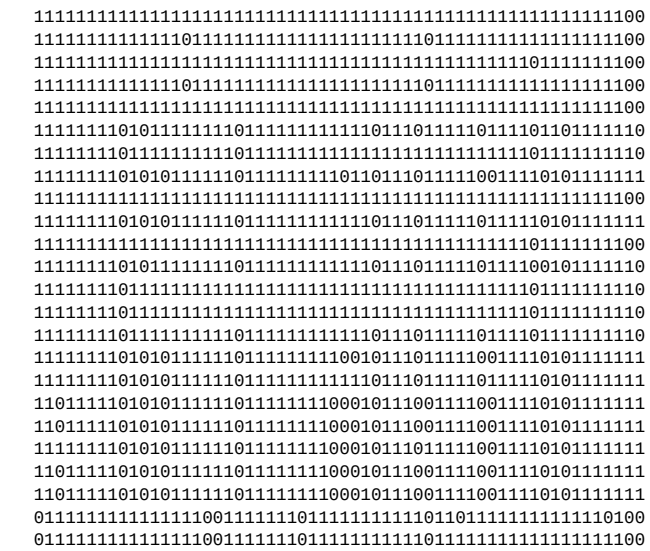

Figure 4: Pattern of a dense sub-matrix (cluster B)

cluster-name	area(row x col)	density
D	5 x 195 = 975	90%
E	3 x 229 = 687	90%
F	6 x 111 = 666	100%

- Clusters with large rows

cluster-name	area(row x col)	density
G	18 x 3 = 54	100%
H	16 x 3 = 48	100%

We can see that various kinds of sub-matrices can be extracted. These patterns show that Web access is not balanced, and large clusters of users and URLs exist.

However, these clusters involve a lot of frequently accessed pages, which are apparent to the Web log analysts. It is more important for them to find relations between Web pages which are not accessed frequently. Therefore, we experimented with the ping-pong algorithm using inhibit parameters. Namely, frequently accessed pages are inhibited prior to the execution in order to find clusters that consist of infrequently accessed pages. The resulting clusters are listed below.

Cluster-name	area(row x col)	density
P	18 x 3 = 54	100%
Q	16 x 3 = 48	100%
R	4 x 6 = 24	90%
S	11 x 3 = 33	100%

3.2 Comparison with Apriori algorithm

The Apriori algorithm[3] can be applied to Web log analysis by treating a cookie as a transaction and

a URL as an item. We also performed experiments to find clusters of pages by using the Apriori algorithm with the same data in order to compare the results. The Apriori algorithm finds all the sets of items whose frequency is greater than the minimum support. If the minimum support is high, then the execution time is short, but the amount of the result is small. On the other hand if the minimum support is low, then the execution time is long, but the amount of the result is large. We determined the minimum support such that the execution time was similar to the total execution time of iterating the ping-pong algorithm for every row. The Apriori algorithm starts to find a pair of items which appears frequently, then increases the number of items in a set gradually. Therefore, all the subsets of a large item set are also included in the solution. For example, when (A,B,C) is included in the solution, (A,B),(A,C),(B,C) are also included in the solution. Since our purpose is to find large item sets, these subsets were removed from the solution. The result is shown in Table 1.

Table 1: Set of URLs obtained by Apriori algorithm

length of item set	total (A)	subset (B)	solution (A) - (B)
2	177	172	5
3	705	701	4
4	2066	2065	1
5	4465	4463	2
6	7278	7278	0
7	9104	9103	1
8	8819	8819	0
9	6623	6622	1
10	3828	3828	0
11	1673	1673	0
12	535	533	2
13	118	118	0
14	16	15	1
15	1	0	1

Table 1 shows that most of the item sets are subsets of other item sets. Only a small number of large clusters are extracted as a solution. This is partly because the minimum support is high. We can extract larger clusters to set the minimum support smaller, but the execution time grows quite large.

We compared the solutions in Table.1 with the page clusters extracted by the ping-pong algorithm. We found that all of the large item sets (length is greater than 9) in Table.1 are contained in cluster-A or cluster-B as described in the previous section. It means that the solutions of the Apriori algorithm can be obtained from the solutions of the ping-pong algorithm by deleting rows and columns which contains elements of value 0. The Apriori algorithm is useful for finding a cluster with large rows and small columns, (also with large columns and small rows by transposing the matrix) but it is not efficient for finding a cluster with large rows and large columns. On the other hand, matrix clustering can find various sizes of clusters efficiently.

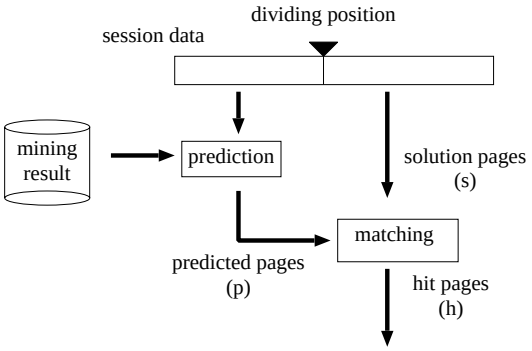

Figure 5: Experiment of prediction

4 Web access prediction

4.1 Experiment

Web page clustering can be applied to Web page prediction by extracting page clusters from already accessed pages. In addition, association rule mining and sequence pattern mining can be applied to Web page prediction[10]. This section compares matrix clustering with association rule mining and sequence pattern mining with respect to the capability of Web access prediction.

We performed an experiment of Web access prediction by using two kinds of log data. The data size is given in Table.2. Data used in experiment 1 are the same as those used in section 3.

Log data with the same user-id are grouped into a session. A session is a sequence of accessed pages ordered by the access time. Duplicated access to the same page within a session is removed except for the first access. The experiment is performed as shown in Fig.5.

Table 2: Data size for experiment

	experiment 1	experiment 2
no. records	14,416	180,249
no. users	1,223	40,932
no. pages	482	4,866
density	1.4%	0.09%

Each session is divided into two parts. The predicted pages are generated by applying the mining result to the former part of a session. Then, the predicted pages are compared with the pages in the latter part of the session. In order to evaluate the capability for prediction, predicted pages are sorted by the confidence value.

Next, conditions for web usage mining are explained. Minimum support value and minimum confidence value are important parameters for executing association rule mining and sequence pattern mining. We restricted the execution time for each mining method within a similar order so as to compare the capability of mining methods. Execution time for association rule mining depends on the minimum support value and size of item set. We set minimum support value as small as possible and restricted the length of item sets to 5, because we thought that better prediction can be obtained by setting minimum support value small. We set minimum confidence value to 0, because confidence value is not heavily dependent on the execution time and predicted pages are ordered by the confidence value.

In the case of sequence pattern mining, we set minimum support value to the same value as in association rule mining. Since sequence pattern mining does not generate as many intermediate results as association rule mining, we need not restrict the length of item sets. The total amount of rules generated by association rule mining and sequence pattern mining for each experiment are shown in Table 3 and 4.

Table 3: Size of generated rules for experiment 1

item set length	association	sequence
1	103	103
2	915	1193
3	9055	5715
4	76432	6417
5	-	2504
6	-	402
7	-	4

4.2 Evaluation method

Evaluation is performed by comparing predicted pages with the latter part of sessions. In order to evaluate the access prediction, the following two parameters are important.

Table 4: Size of generated rules for experiment 2

item set length	association	sequence
1	82	82
2	762	900
3	2865	1546
4	6321	1011
5	9557	348
6	10205	64
7	7622	3
8	3834	-
9	1211	-
10	210	-
11	14	-

- dividing position within a session stream
- size of prediction

The dividing position defines the amount of information to be used for prediction and the amount of solutions for the prediction which corresponds to the latter part of a session. Setting the dividing position near to the beginning of the session means that prediction is executed with a small number of accessed pages and the solution page space for the prediction is large. On the other hand setting the dividing position near to the ending of the session means that prediction is executed with a large number of accessed pages and the solution page space for the prediction is small. Hence, it is feasible to set the dividing position at the center of the stream.

When the size of prediction pages is large, it is expected that most of the solution pages will be included in the predicted pages. Since different mining methods may generate different sizes of the set of predicted pages, it is necessary to consider the ratio of matched pages to the total predicted pages.

Now, we define two kinds of hit rate for prediction.

- p-hit rate = h / p
- s-hit rate = h / s

where h : hit pages, p : predicted pages, and s : solution pages.

Namely, p-hit rate represents the hit rate to the predicted pages, and s-hit rate represents the hit rate to the accessed pages. In general, high s-hit rate can be obtained by setting the size of prediction pages large, whereas, p-hit decreases. Since the size of predicted pages is different for each mining method, these two hit rates are not adequate for comparing the performance of prediction. Therefore, we introduce the normalized hit rate(n-hit rate) by setting the number of predicted pages equal to the number of solution pages.

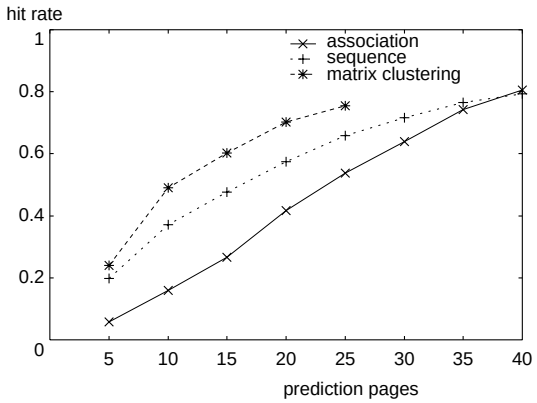

Figure 6: s-hit rate by changing the size of predicted pages in experiment 1

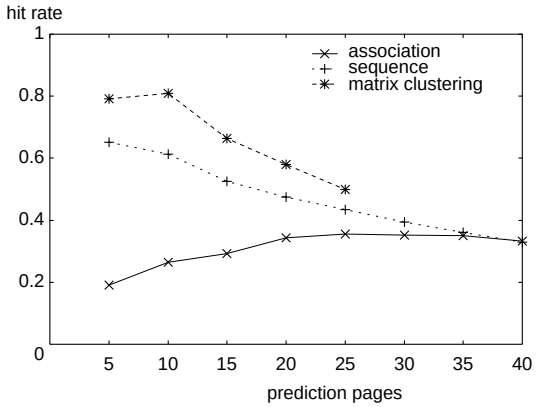

Figure 7: p-hit rate by changing the size of predicted pages in experiment 1

- n-hit rate = h / s
  (where s = p)

4.3 Result of Experiment 1

Session length of data used in experiment 1 is rather long. Fig.6 to Fig.8 show the results of experiment 1 where session lengths are between 25 and 50. Fig.6 shows s-hit rate and Fig.7 shows p-hit rate at dividing position 15 by changing the number of prediction pages from 5 to 40. Fig.8 shows n-hit rate by changing the dividing position to 5,10,15, and 20.

In the case of matrix clustering, the number of predicted pages is between 20 and 30, and is not dependent on the dividing position. In Fig.6 and Fig.7, the predicted pages are limited to 25 at matrix clustering. In the case of association rule mining, the number of predicted pages is always about 70, and are not dependent on the dividing position. In the case of sequence pattern mining, the number of predicted pages is always about 60. From Fig.6

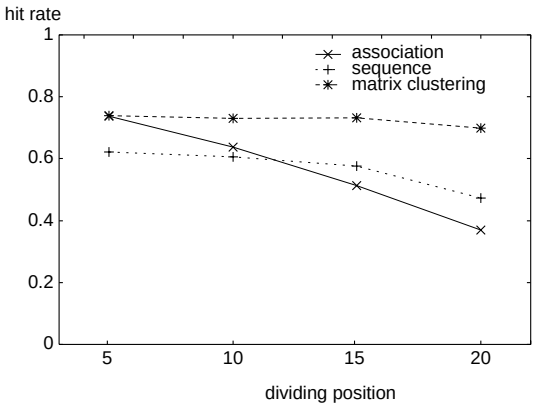

Figure 8: n-hit rate by changing the dividing position in experiment 1

and Fig.7, it is apparent that matrix clustering is superior to the other two methods when the number of predicted pages is less than 25.

Fig.6 shows that hit rate of matrix clustering reaches 80% with 25 pages of prediction, whereas sequence pattern mining requires 35 pages and association rule mining requires 40 pages. Fig.7 shows that hit rates of matrix clustering and sequence pattern mining decrease as the number of prediction pages increases, whereas that of association rule mining increases. It means that pages with high priorities may not be hit well, which represents that ordering by the confidence value in association rule mining is inadequate for predicting page access.

Fig.8 shows that performance of prediction by matrix clustering is almost constant in the case of changing the dividing position. On the other hand, performance of prediction by association rule mining and sequence pattern mining decreases as the dividing position gets near to the end. Since the number of predicted pages decreases as the dividing position gets near to the end, the difference in ordering of the predicted pages greatly affects the performance.

By comparing these predicted pages, we can see that most of pages predicted by matrix clustering are included in the pages predicted by the other two methods. It means that fundamental prediction capabilities are similar among these three methods. The superiority of matrix clustering to the association rule mining and sequence pattern mining comes from the ordering of predicted pages.

4.4 Result of Experiment 2

The mean length of session streams in experiment 2 is rather short. We selected the longest 130 streams (length between 15 and 28) as test data. Fig.9 to

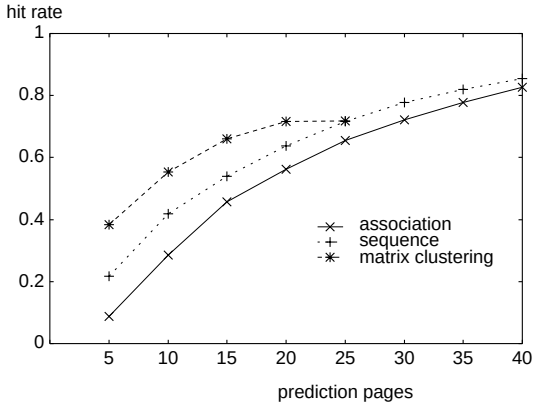

Figure 9: s-hit rate by changing the size of predicted pages in experiment 2

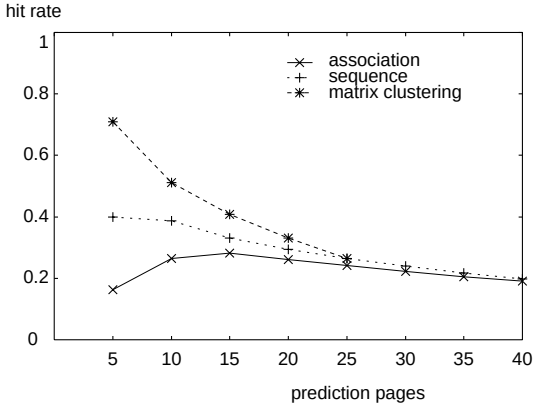

Figure 10: p-hit rate by changing the size of predicted pages in experiment 2

Fig.11 show the results of experiment 2. Fig.6 shows s-hit rate and Fig.7 shows p-hit rate at dividing position 8 by changing the number of prediction pages from 5 to 40. Fig.8 shows n-hit rate by changing the dividing position to 3,6,9, and 12.

In the case of matrix clustering, the number of predicted pages is about 20, and about 70 in the cases of association rule mining and sequence pattern mining. Global tendency of prediction performance is similar to that in the experiment 1, but hit rate is lower. It is because the session length is shorter. Fig.9 and Fig.10 show that matrix clustering is superior to association rule mining and sequence pattern mining when the number of predicted pages is less than 20. Fig.10 shows that hit rates of matrix clustering and sequence pattern mining decrease as the number of prediction pages increases, whereas that of association rule mining increases. It means that ordering by the confidence value in association rule mining is inadequate for predicting page access.

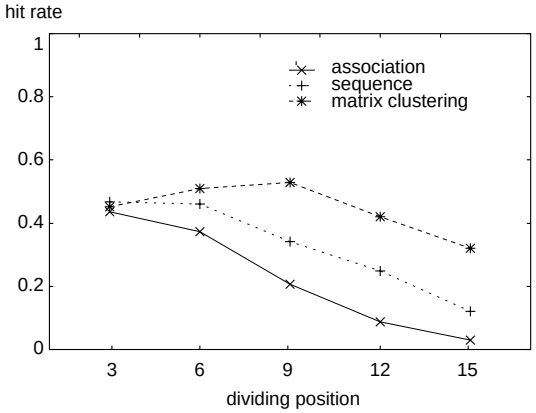

Figure 11: n-hit rate by changing the dividing position in experiment 2

Fig.11 shows that performance of prediction by matrix clustering is superior to those of association rule mining and sequence pattern mining, however performance of prediction decreases as the dividing position gets near to the end. It is because the session length is too short to predict correctly.

These results show that the superiority of matrix clustering to association rule mining and sequence pattern mining is not dependent on the Web site structure. We conclude that matrix clustering is more suitable for Web access prediction when the session length is long enough.

5 Conclusion

This article has described matrix clustering and its application to Web access prediction. Matrix clustering is a simple and powerful mining method to extract page clusters of various sizes. The experiment shows that page clusters generated by association rule mining are contained in the page cluster extracted by matrix clustering. We conclude that matrix clustering is more powerful than association rule mining with respect to finding various kinds of page clusters.

By applying the extracted page clusters to Web access prediction, better prediction capability can be obtained than in the cases of using association rule mining and sequence pattern mining. The result of experiment suggests that long history of Web access may match with page clusters extracted by matrix clustering. Matrix clustering is also efficient enough to execute in real time. It is important for frequently modified Web sites to realize personalized services.

It should be noted that prediction performance is insufficient when the session length is short. Ob-

viously, this is a property common to data mining methods. Hence, we need to integrate web usage mining methods with other methods such as utilizing user profiles for practical personalization systems. We are planning to develop personalized Web systems based on matrix clustering.

References

[1] U.Fayyad, E.Simoudis : Data Mining and KDD: An Overview, Third Intl. Conf. on Knowledge Discovery & Data Mining, (1997)

[2] M.Berry, G.Linoff : Data Mining Techniques for Marketing, Sales, and Customer Support, John Wiley and Sons, (1997)

[3] R.Agrawal, R.Srikant : Fast Algorithms for Mining Association Rules, Proc. 20th VLDB Conf.,pp,487-499 (1994)

[4] R.Agrawal, R.Srikant :Mining sequential patterns, Proc.Intl.Conf. Data Engineering (1995)

[5] J.Schafer, J.Konstan, J.Riedl : E-Commerce Recommendation Applications, ACM Conference on EC,(2000)

[6] U.Shardanand, P.Maes : Social Information Filtering: Algorithms for Automating "Word of Mouth", Proceedings of CHI95,(1995)

[7] J.Srivastava, R.Cooley, M.Deshpande, P.Tan : Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, SIGKDD Explorations, Vol.1,Issue 2(2000)

[8] M.Spiliopoulou : Web Usage Mining for Web Site Evaluation, Comm.ACM, vol.43, No.8, pp.127-134 (2000)

[9] B.Mobasher, R.Cooley, J.Srivastava : Automatic Personalization Based on Web Usage Mining, Comm.ACM, Vol.43, No.8, pp.142-151 (2000)

[10] B.Lan, S.Bressan, B.Ooi : Making Web Servers Pusher, WEBKDD-99,(1999)

[11] Y.Fu, K.Sandhu, M.Shih : Clustering of Web Users Based on Access Patterns, WEBKDD-99, (1999)

[12] S.Oyanagi, K.Kubota, A.Nakase : Matrix Clustering: a New Data Mining Algorithm for CRM, Trans.IPSJ, (2001) (in Japanese)