Journal of Engineering and Applied Sciences 7 (7): 456-461, 2012
ISSN: 1816-949X

Query Based Biclustering of Web Usage Data

K. Thangavel and R. Rathipriya
Department of Computer Science, Periyar University, Salem, Tamil Nadu, India

Abstract: In this study, a biclustering algorithm based query model is proposed that is able to extract biclusters of web objects (i.e., users and pages) from web usage datasets. This Query Based Biclustering (QBB) algorithm is applied to the web usage data to recruit biclusters with respect to query which contain a certain users of similar browsing pattern across a subset of pages of a web site. By this way, one can target the right group of user for the focalized marketing strategy. In this application, the main goal is to identify group of web users or customers with similar behavior so that one can predict the customer's interest and make proper recommendations to improve their performance. To evaluate the efficiency of the proposed algorithm, the experiment is conducted on the CTI dataset. Results show that the proposed QBB algorithm is efficient in extracting the maximum similar bicluster based on the query.

Key words: Web usage mining, biclustering, query based biclustering, similarity bicluster, target marketing

INTRODUCTION

For the past one decade, many studies have been suggested to analyze the voluminous web log data by using web usage mining techniques. Cooley *et al.* (1999) discussed methods to preprocess the user log data and proposed the method for web page reference model. And also they proposed a user navigation behavior model using web server log (Srivastava *et al*., 2000).

Clustering is the one of the most popular approaches used in web usage data analysis. It is used to group users who have similar browsing interest under entire set of pages of a web site or to group pages based on the similar browsing interest of the users. It usually seeks a disjoint cover of the set of elements requiring that no user or page belongs to more than one cluster. In most of the papers, web clustering approaches were based on a distance function to identify the objects (either user or page) that are clustered together (similarity based) or to other probabilistic techniques called model based clustering. These clusters produce global pattern because clustering techniques are based on entire set of users or pages. And hence, these approaches are not able to discover the browsing patterns that are common to a group of users only under specific subset of pages. Discovering such local browsing patterns may be the key feature for e-Commerce application like target marketing that is not apparent otherwise. Therefore, it is highly desirable to move beyond the clustering paradigm and develop approaches capable of discovering local patterns in web usage data.

Consider the sample data matrix as shown in Fig. 1. If one considers all pages, users 2-4 do not seem to be have similarly since their values are not similar under page1 and 2. However, these users behave similarly under pages 3 and 4 since all their values are 1. A traditional clustering method will fail to recognize such a cluster since the method requires the three users to behave similarly under all pages which are not the case.

In order to address these problems, the biclustering concept was introduced. It was proposed by Hartigan (1972). Biclustering approach may be based on fuzziness, i.e., web data elements are assigned to one or more biclusters with different membership levels (Koutsonikola and Vakali, 2009). Only few paper in the literature that have been proposed for the biclustering of binary data. They are Bimax algorithm (Prelic *et al*., 2006) discover the biclusters of ones whose time complexity is very high.

1	0	0	1
0	1	1	1
1	0	1	1
0	1	1	1

Fig. 1: Sample data matrix

Corresponding Author: K. Thangavel, Department of Computer Science, Periyar University, Salem, Tamil Nadu, India

Cmnk biclustering algorithm (Koyuturk *et al*., 2004) was not able to extract the bicluster in the noisy background. The biclustering algorithm BicBin (Van Uitert *et al*., 2008) that is able to find biclusters in large scale sparse binary datasets but it does not produce meaningful biclusters with less non-zero element. Biclustering based on the self matrix multiplication is proposed in Qin *et al*. (2008) which identify either non-overlapped biclusters or overlapping biclusters of ones. These entire biclustering algorithms are failed to produce the bicluster with small proportion of zeros from sparse binary dataset. Sometime small proportion of zeros in the biclusters is also useful when one looking of coherent browsing pattern comprises of visited and not visited pages of a web site.

Given a query or user/page reference pattern and then calculate the similarity of every other user/page to the query to identify the users/pages most related to the query to report as results. This biclustering technique can be used to identify subgroups of customers with similar preferences or behaviors towards a subset of pages with the goal of performing target marketing. The information provided by the biclusters can also be used in recommendation systems and electoral data analysis. It provides a way to find users' behavior, their preferences and desires in order to provide each user excellent and personalized web based services at low costs.

MATERIALS AND METHODS

Preliminaries: Biclustering formulated in the context of gene expression data by Cheng and Church (2000). Since 2000, a number of papers have been published on biclustering. Each biclustering algorithm tend to discover different types of biclusters, it is quite difficult to get an objective comparison among the performance of different biclustering methods due to the different nature of the results they provide. One of the beauties of this family of analysis methods since they are able to provide different type of information.

Notation: Given data matrix A (X, Y) with set of rows X and set of columns Y, a bicluster is a submatrix (i, j) where i is a subset of the rows X and j is a subset of columns Y and a_{ij} is the value in the matrix A corresponding to row i and column j (Madeira and Oliveira, 2004).

Preprocessing of web usage data: Let A (U, P) be an n× m matrix of 0's and 1's where $U = \{U_1, U_2, \ldots, U_n\}$ be a set of users and $P = \{P_1, P_2, \ldots, P_m\}$ be a set of pages of a web site. It is used to describe the relationship between web pages and users who access these web pages. The element a_{ij} of A(U, P) represents whether the U_i of U visit the P_j of P during a given period of time:

$$a_{ij} = \begin{cases} 1 \text{if } P_j \text{ is visited by } U_i \\ 0 \text{ otherwise} \end{cases} \quad (1)$$

For user subset U' of U and page subset P' of P, A (U', P2') denotes the submatrix called bicluster of A(U, P) that contain only the elements a_{ij} satisfying U'⊆U and P2'⊆P (Rathipriya *et al*., 2011).

Liu and Wang (2007) introduced maximum similarity biclustering concept in gene expression analysis and it is popularly known as MSBE. This methodology initiated to apply the same idea to web usage data for extraction of maximum similarity biclusters using Hamming similarity. The fundamental principles for using biclustering in the analysis of web usage data are; similar users may exhibit similar behaviors only under a subset of pages, not all pages; users may show interest for >1 category of pages, resulting in one browsing pattern in one context and a different pattern in another. Using biclustering algorithms, one can obtain sets of users that are co-regulated under subsets of pages.

Query: An information need is expressed by an end user in a recognized form is usually called a query. Query plays a vital role in extracting the biclusters from the web usage data. It is also known as reference pattern vector. The query based biclustering approach helps to answer the following questions:

- Which users are interested in what pages of a web site?
- Given a set of known users/pages as a reference pattern, how to select new candidate users/pages that may be linked to the same reference pattern?
- Which users/pages are (logically) related to the seed users/pages and which pages/users are relevant for this browsing pattern?

Selection of query: Let A (U, P) be an user access matrix of n×m . There are four different way of selecting the query from A. They are:

- User query; given matrix A, end user selects any one of user sessions as a user query or user reference pattern
- Page query; given matrix A, end user selects any one of pages as a page query or page reference pattern
- Random selection query; random selection of user/page query from A
- Random query; given matrix A of size n x m, end user sets the user/page query with random values. User query is the sequence of the random values of length m. Page query is the sequence of the random values of length n

Similarity measures: The measure reflects the degree of closeness of the web objects (i.e., user or page) to the given query and should correspond to the characteristics that are believed to distinguish the biclusters embedded in the data. In many cases, these characteristics are dependent on the data or the problem context at hand and there is no measure that is universally best for all kinds of biclustering problems.

Moreover, choosing an appropriate similarity measure is also crucial for bicluster analysis, especially for a particular type of biclustering algorithms. In general, similarity measures map the distance or similarity between the symbolic descriptions of two objects into a single numeric value. In QBB, similarity between query and objects is measured to find bicluster with maximum similarity with respect to the given query. In this research, two types of similarity measures are used for extraction of biclusters with browsing patterns. They are:

- Liu similarity measure (A similarity measure defined by Liu)
- Hamming similarity measure

Liu similarity measure: Let us consider A (U, P) be user access matrix, a_{ij} be an element of A (U, P), user query be I*U and page query be j*P, user query be i*U and page query be j*P. Liu similarity measure computes the distance between every elements of A and element of the query and constructs a similarity matrix for the query. The distance between the element of every user of A and element of the user query is defined as:

$$d1_{ij} = |a_{ij} - a_{i*j}| \text{ for } i = 1, 2\ldots n \text{ and } j = 1, 2, \ldots, m \quad (2)$$

The distance between the element of every page of A and element of the page query is defined as:

$$d2_{ij} = |a_{ij} - a_{i*j}| \text{ for } i = 1, 2\ldots n \text{ and } j = 1, 2,\ldots, m \quad (3)$$

d_{avg} is the average distance value of all elements in the distance matrix and it is defined as:

$$d_{avg} = \frac{\Sigma_{i\in U, j\in p} dk_{ij}}{|I| * |J|} \quad (4)$$

Where k = 1, 2 and |U| and |P| are number of users and pages, respectively.

Thershold is $\alpha.d_{avg}$ where α is a constant. If $d_{ij} > \alpha.d_{avg}$ then the two elements a_{ij} and a_{i*j} are not similar and set the similarity s_{ij} to be 0 otherwise the similarity score is defined as:

$$s_{ij} = \begin{cases} 0 & \text{if } d_{ij} > \alpha.d_{avg} \\ 1 - \dfrac{d_{ij}}{\alpha.d_{avg}} + \beta, & \text{otherwise} \end{cases} \quad (5)$$

Where β is used to increase the value for small d_{ij} and ignore d_{ij}'s that are greater than $\alpha.d_{avg}$. Here, S (U, P) is used to denote the n×m similarity matrix containing the set of rows U and set of columns P with every element s_{ij} computed.

Hamming similarity measure: Hamming distance is simply defined as the number of bits that are different between two bit vectors. In the context of biclustering, Hamming distance is used to capture the distance between the element of every user of A and element of the query. Construct distance matrix using the following equations. The distance between the element of every user of A and element of the user query is defined as:

$$d1_{ij} = |a_{ij} - a_{i*j}| \text{ for } i = 1, 2,\ldots, n \text{ and } j = 1, 2, \ldots, m \quad (6)$$

The distance between the element of every page of A and element of the page query is defined as:

$$d2_{ij} = |a_{ij} - a_{i*j}| \text{ for } i = 1, 2,\ldots,n \text{ and } j = 1, 2,..., m \quad (7)$$

If $dk_{ij} = 0$ then the two elements a_{ij} and a_{i*j} are not similar and set the similarity s_{ij} to be 0 otherwise the similarity score is defined as:

$$s_{ij} = \begin{cases} 1, & \text{if } d_{ij} = 0 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Here, S (U, P) is used to denote the n×m similarity matrix containing the set of rows U and set of columns P with every element s_{ij} computed as in Eq. 8.

Similarity score for a bicluster: Let s(U, P) be a n×m similarity matrix and S(U', P') be a bicluster of S(U, P). The similarity score of each row i*U' in the bicluster is defined as:

$$s(i, P') = \Sigma_{j\times p'} s_{ij} \quad (9)$$

The similarity score of each column j∈P is defined as:

$$s(U', j) = \Sigma_{i\times U'} s_{ij} \quad (10)$$

The similarity score of a whole bicluster S (U', P') is:

$$\text{Min } \{\min_{i\times U'} s(i, P'), \min_{j\times P2'} s(U', j)\} \quad (11)$$

Table 1: Comparison between QBB and MSBE algorithms

Query Driven Biclustering (QBB) of binary data	MSBE
Specified user (i*) or page j* is used as query	Every gene is considered as a reference gene i*
Data preprocessing is required for conversion of web usage data into binary data	Not applicable
Number of iterations is very less than m+n-2	Number of iterations is m+n - 2
Multiple node deletion method is used	Single node deletion method is used
The time required to extract single bicluster is approximately 3.5 sec	The time required to extract one bicluster is approximately is 68 sec
It is used to analysis the user's behavior related to the query	It was applied for gene expression analysis
Applications are target marketing and recommendation systems	

Find minimum similarity score for both row and column of the bicluster. The similarity score of the bicluster is the minimum of these two similarity scores.

Query based biclustering algorithm: This study describes the proposed query based biclustering algorithm to recruit maximum similarity bicluster with respect to query. The proposed biclustering algorithm recruits the maximum similarity bicluster with local browsing pattern. Initialize the query according to the nature of the problem. Query is nothing but reference vector which may be user reference vector or page reference vector. If there is no information about query (i.e., reference user and page) then randomly select a set of rows and a set of columns as user or page query. Algorithm 1 is used to compute the distance matrix that related to the given query using either Lui similarity or Hamming similarity measure. Using the distance matrix, compute similarity matrix using either Eq. 5 or 8.

Algorithm 1: Query based biclustering algorithm

Input: A n×m user access matrix A(U, P), similarity_measure
Output: A maximum similarity bicluster $A(U_A, P_A)$

i. Select the either user or page reference pattern vectoras $U_{i^*} = \{U_{i^*1}, U_{i^*2},\ldots,U_{i^*n}\}$ or $P_{j^*} = \{P_{1j^*}, P_{2j^*}, \ldots, P_{mj^*}\}$
ii. Compute the similarity matrix S(U, P) with respect to referenced pattern vector
iii. Initialize whole similarity matrix as first bicluster $S(U_1, P_1) = S(U, P)$
iv. k = 1
v. Repeat
   a. Calculate row similarity for $i'\times U_k$ using Eq. 11
   b. Set I as indexes of minimum row similarity score and rowval = $\min_{i\times U_k} s(i, P')$
   c. Calculate column similarity for $j\times P_k$ using Eq. 13
   d. Set J as indexes of minimum column similarity score and colval = $\min_{j\times P_k'} s(U', j)$
   e. If rowval < = colval then
      Set $U_{k+1} = U_k - \{I\}$ and $P_{k+1} = P_k$
      else
      Set $U_{k+1} = U_k$ and $P_{k+1} = P_k - \{J\}$
      End (if)
   f. Calculate the similarity score of a bicluster $A(U_{k+1}, P_{k+1})$
   g. k = k+1

Until ((isempty (U_k) = true) || isempty (P_k) = true)

vi. Return bicluster A (U_A, P_A) with maximum similarity score as maximum similarity bicluster

It is a greedy approach to extract the maximum similarity bicluster. Initialize the whole matrix as the bicluster. Repeat the following steps until there is one element in the similarity matrix. Calculate the similarity score for all rows and columns of the similarity matrix using the Eq. 9-10. Delete the set of rows or columns whose similarity score is the smallest among all the rows and columns in the current bicluster. This process is repeated until there is no element in the similarity matrix. During this process, k biclusters are obtained where k<n+m-2. Among these biclusters, choose the bicluster that has the maximum similarity score (Table 1).

RESULTS AND DISCUSSION

The purpose of this study is to show the performance and effectiveness of the proposed method on CTI dataset. All the experiments were carried out on a 2.4 GHz, 256 MB RAM, Pentium-IV machine running on Microsoft Windows 7 and the code was developed on Matlab toolbox.

CTI dataset description: The dataset, CTI is from a university web site log and was made available by the researchers of Mobasher (2004) and Zhang *et al.* (2005). The data is based on a random collection of users visiting university site for a 2 weeks period during April, 2002. After data preprocessing, the filtered data consisted of 13745 sessions and 683 pages. Further preprocessed CTI dataset where the root pages were considered in the page view of a session. This preprocessing step resulted in total of 16 categories; namely, search, programs, news, admissions, advising, courses, people, research, resources, authenticate, CTI, pdf, calendar, shared, forums and hyperlink. These page views were given numeric labels as 1 for search, 2 for programs and so on.

Table 2 shows the complete list of numeric coded web pages. Each row of CTI dataset describes the hits of a single user. The session length in the dataset ranges from 2-68. Since comparing very long sessions with small sessions would not be meaningful, researchers considered only sessions of length between 3 and 7. Finally, 5915 user sessions are taken for the experimentation.

Types of measures: Biclustering algorithms are evaluated on the basis of their ability to find different types of

Table 2: Numeric code of web pages in CTI dataset

Web page name	Coding	Web page name	Coding
Search	1	Resources	9
Programs	2	Authenticate	10
News	3	Cti	11
Admissions	4	PDF	12
Advising	5	Calendar	13
Courses	6	Shared	14
People	7	Forums	15
Research	8	Hyperlink	16

biclusters. A perfect bicluster can take different forms: Constant value, constant rows/columns or coherent values on rows/columns. Various measures are available in the literature to evaluate the quality of the different types of biclusters. They are Mean Squared Residue (MSR) score and Average Correlation Value (ACV).

Mean squared residue score: Cheng and Church (2000) defined a bicluster as a subset of rows and subset of columns which has low Mean Squared Residue (MSR) score:

$$MSR(B) = \frac{1}{nm}\sum_{i}\sum_{j}(b_{ij} - \overline{b_i} - \overline{b_j} + \overline{b_{ij}})^2 \qquad (12)$$

Where:

$\overline{b_i}$ = The average value of row I

b_j = The average value for column j

Average correlation value: Average Correlation Value (ACV) (Teng and Chan, 2007) is used to evaluate the homogeneity of a bicluster. But, ACV is not applicable for constant bicluster. Matrix B = (b_{ij}) has the ACV which is defined by the following function:

$$ACV(B) = \max\left\{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}|row_{ij}| - n}{n^2 - n}, \frac{\sum_{k=1}^{m}\sum_{l=1}^{m}|col_{kl}| - m}{m^2 - m}\right\} \qquad (13)$$

Where:

row_{ij} = The correlation between row i and row j

col_{kl} = The correlation between column k and I

A high ACV suggests high similarities among the rows or columns. ACV can tolerate translation as well as scaling. ACV can be used as merit function to find non-perfect biclusters.

A perfect bicluster may have different forms, they are constant value, constant rows/columns or coherent values on rows/columns. There is no uniform quality measure for coherence. In this study, proposed biclustering method finds the biclusters with low MSR and high ACV, i.e., maximum value of ACV is 1. A bicluster with high ACV suggests high correlation on either dimension (or both dimensions) of the data matrix. ACV measures more types of biclusters comparing to the residue related criteria since ACV can tolerate transformations like translation and scaling.

Table 3: Biclusters obtained by using Hamming similarity measure

	Bicluster			
User query id	No. of users	No. of pages	ACV	MSR
3442	700	16	1	0
8	5871	16	1	9
1999	52	15	1	0
23	882	16	1	0
5300	5477	15	1	0

Table 4: Biclusters obtained by using Liu similarity measure

	Bicluster			
User query id	No. of users	No. of pages	ACV	MSR
3442	700	16	1	0
8	117	16	1	4844
1999	252	15	1	0
23	982	16	1	0
5300	736	15	1	1

Table 5: Biclusters obtained by using hamming similarity measure

Page query ids	Similarity score of a bicluster	Bicluster			
		No. of users	No. of pages	ACV	MSR ($1*e^{04}$)
4	15	290	17	0.4064	6.3875
5	15	310	16	0.4064	6.3875
8	15	350	17	0.4064	6.3875
11	15	159	16	0.2678	0.9589
15	15	297	16	0.4191	6.0453
16	15	396	16	0.4064	6.3875

Table 3 shows the characteristics of the bicluster extracted using Hamming similarity measure. In Table 3, most of the MSR values are 0 where as ACV is 1 which means that all identified biclusters are of constant row/constant column bicluster.

Table 4 shows the characteristics of the bicluster extracted using Liu similarity measure. Table 4 most of the MSR values are 0 where as ACV is 1 which means that all identified biclusters are of constant row constant column bicluster. But, the number of users in the biclusters is less than biclusters in Table 3. From Table 3 and 4, it is concluded that hamming similarity measure extracts the larger volume bicluster than Liu similarity.

The characteristics of the biclusters with reference to the page query using Hamming and Liu similarity measures are shown in Table 5 and 6, respectively. It is inferred that Hamming similarity measure based QBB is good in extract maximum similarity bicluster with high volume from the binary dataset than Liu similarity measure.

The results of the comparative study based on the number of iterations for the user query user id in Table 3 on MSBE and QBB algorithms are shown in

Table 6: Biclusters obtained by using Liu similarity measure

Page query ids	Similarity score of a bicluster	Bicluster		ACV	MSR ($1*e^{04}$)
		No. of users	No. of pages		
4	22.5	301	14	0.4064	6.3875
5	22.5	301	13	0.4064	6.3875
8	22.5	301	16	0.4064	6.3875
11	22.5	110	15	0.2678	0.9589
15	22.5	297	14	0.4191	6.0453
16	22.5	301	16	0.4064	6.3875

Table 7: Comparison of MSBE and QBB based on No. of iterations

Hamming similarity measure		Liu similarity measure	
QBB	MSBE	QBB	MSBE
8	5916	8	5916
10	5916	30	5916
11	5916	21	5916
8	5916	18	5916
10	5916	17	5916

Table 7. QBB algorithm using multiple node deletion used less iteration than MSBE without degrading the quality of the biclusters.

CONCLUSION

The main objective of the proposed biclustering is to recruit the similar set of users/pages based on the user/page reference pattern or query. Each bicluster defines the browsing pattern of the user across the subset of pages or vice versa. The main contribution in this algorithm is the usage of hamming similar measure to recruit the most similar and constant bicluster based on the query. For target marketing application, it identified the subgroups of customers with similar preferences or behaviors towards a subset of pages. Analyzing the results, it can be used for any other focalized marketing strategy. The quality of the biclusters is evaluated by the different measures like MSR and ACV for constant user bicluster, constant page bicluster and Additive bicluster. Thus, the query driven biclustering approach assures the targeted search and provides computational feasibility and extensibility.

REFERENCES

Cheng, Y. and G.M. Church, 2000. Biclustering of expression data. Pcoc. Int. Conf. Intell. Syst. Mol. Biol., 8: 93-103.

Cooley, R., J. Srivastava and M. Deshpande, 1999. Data preparation for mining world wide web browsing patterns. Knowledge Infor. Syst., 1: 5-32.

Hartigan, J.A., 1972. Direct clustering of a data matrix. J. Am. Stat. Assoc. 67: 123-129.

Koutsonikola, V.A. and A.I. Vakali, 2009. A fuzzy bi-clustering approach to correlate web users and pages. Int. J. Know. Web Intell., 1: 3-23.

Koyuturk, M., W. Szpankowski and A. Grama, 2004. Biclustering gene-feature matrices for statistically significant patterns. Proceedings of the IEEE Computer System Bioinformation Conference, Volume 16, August 16-19, 2004, IEEE Computer Society, USA., pp: 480-484.

Liu, X. and L. Wang, 2007. Computing the maximum similarity biclusters of gene expression data. Bioinformatics, 23: 50-56.

Madeira, S.C. and A.L. Oliveira, 2004. Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Trans. Comput. Biol. Bioinform., 1: 24-45.

Mobasher, B., 2004. Web Usage Mining and Personalization. In: Practical Handbook of Internet Computing, Singh, M.P. (Ed.). CRC Press, USA.

Prelic, A., S. Bleuler, P. Zimmermann, A. Wille and P. Buhlmann *et al*., 2006. A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics, 22: 1122-1129.

Qin, R.X., Y.J. Tian, J. Chen and N.Y. Deng, 2008. Biclustering based on self-multiplication of matrix. Proceedings of the 2nd International Symposium on Optimization and Systems Biology, October 31-November 3, 2008, Lijiang, China, pp: 304-310.

Rathipriya, R., K. Thangavel and J. Bagyamani, 2011. Evolutionary biclustering of clickstream data. Int. J. Comput. Sci. Issues, 8: 32-38.

Srivastava, J., R. Cooley, M. Deshpande and P.N. Tan, 2000. Web usage mining: Discovery and applications of usage patterns from web data. SIGKDD Explorat., 1: 12-23.

Teng, L. and L. Chan, 2007. Discovering biclusters by alternatively sorting with weighted correlation coefficient in gene expression data. J. Signal Process. Syst., 50: 267-280.

Van Uitert, M., W. Meuleman and L. Wessels, 2008. Biclustering sparse binary genomic data. J. Comput. Biol., 15: 1329-1345.

Zhang, Y., G. Xu and X. Zhou, 2005. A latent usage approach for clustering Web transaction and building user profile. Proceedings of the 1st International Conference on Advanced Data Mining and Applications ADMA, July 22-24, 2005, Wuhan, China, pp: 31-42.