Cluster Comput
DOI 10.1007/s10586-017-0988-z

A four-gram unified event model for web mining

Xinyao Zou[1]

Received: 13 March 2017 / Revised: 11 June 2017 / Accepted: 14 June 2017

Abstract In order to improve the quality of web data mining algorithm, this paper summarizes the advantages and disadvantages of several web data source models, including web log, application server log, Client-side log, Packet sniffer, and 5-gram united events model. Based on this analysis, a new 4-gram united events model (UEM4) is proposed in this paper. Simulation experiments were conducted to verify the performance of UEM4, compared with web log and 5-gram united events model. The experiment results show that web log has the worst session identification performance; UEM5 has high accuracy, best online and offline performance, but it needs the application system support the ability to identify the session; UEM4 does not require the application system to support session identification, and also has a good accuracy and performance of session identification. Therefore, this model can be used in e-commerce, which can provide high quality data sources for web mining algorithms and improve the quality of intelligent services.

Keywords 4-Gram unified events model · Session identification · User session

1 Introduction

Web mining is a hot topic nowadays [1–12]. Web mining algorithm can be used to find hidden information from a large number of web data, which has been widely used in e-commerce decision support applications, such as personalized recommendation. Personalized recommendation can improve e-commerce sales from three aspects: browser into buyers, increase cross selling, and build loyalty. No matter what kind of algorithm is used, the data model influences mining results greatly [13–18]. The quality of a data model can be evaluated from three parts, including collecting, storing and preparing. Data collecting can deploy in client-side, proxy-side and serve-side. Client-side collecting can be forbidden by users for privacy reasons. Proxy-side collecting can only collect the behavior of clients that send requests to the proxy. Serve-side collecting can collect all server activities of all accessing clients. Files (structured or unstructured) and databases are the main mediums for web data storing. Data preparing, including cleaning, integration, selection and transformation, is to pre-process stored data and transform them according to the requirement of web mining algorithms.

At present, there are several kinds of common web data source model, including web log [19], application server log [20], Client-side log, Packet sniffer [21], and 5-gram united events model [22]. Web log is the most frequently used data source in web mining system. This data is common in a variety of web servers, such as IIS, Apache, etc. Kohavi [23] points out the shortcomings of the web log as a data source for the web mining algorithm: (1) poor data collection. First, due to the impact of local cache, proxy servers and firewalls, part of the data can't reach the server, which causes it is not complete in web log data collection. Secondly, there are many problems in the clock synchronization of multi file system. Thirdly, the web log contains a lot of redundant information, such as the JPG, BMP and other pictures downloaded due to the need of Http protocol. (2) It is not designed for analysis. First, the URL string lacks the semantics, which is logged by web log. Especially in today's most dynamic web site, URL analysis is almost meaningless because the page content and site structure can be dynamically generated. The second aspect is the lack of storage form information,

✉ Xinyao Zou
xinixy@163.com

[1] Mechanical and electrical department, Guangdong AIB Polytechnic College, Guangzhou 510507, China

Published online: 21 June 2017

which is very useful for the analysis of decision-making. Thirdly, there is no other way to collect useful information, such as the user's local time, the user's screen size, etc. (3) Pre-processing is difficult and inaccurate. First of all, the preprocessing could be affected by the downloaded images and sounds. Secondly, due to the stateless Http protocol, user session identification usually depends on cookie, timestamp, IP, and browser type, but some problems will occur due to the proxy server cache, IP redirection and the cookie refused by user browser. Thirdly, the dynamic IP allocation makes the same IP for many different users, while the same user has different IP in different time periods. User session identification is a heuristic approach to refactoring, and the results are imprecise. (4) the data type is unitary, and there is no effective way to integrate with other electronic commerce event data.

Application layer log uses the application layer system to record data of various key events in e-commerce, including browsing, buying, adding shopping basket and so on. Application layer log data is collected at the server application layer, which can well identify users and sessions, integrate browsing records and purchase records, and analyze key events in E-commerce. It has three advantages: (1) high quality data collection. Because data logging is on the application layer, application layer log does not contain redundant information. In addition, application layer logs also can get a lot of information that is not available in the Web logs, such as user connection speed, form information, page content, etc. (2) Convenient user identification and session identification. Since the application layer directly records sessions, users register, log in and log off, there is no need to use heuristic rules to reconstruct sessions, thus reducing the time overhead caused by preprocessing and improving the accuracy of the analysis. (3) Easy integration of multiple data. In general, the application layer server logs can well identify the user and the session, integrate the browsing records and purchase records, and analyze the key events of e-commerce. At the same time, there are also some problems in the implementation. In detail, the application layer log for the new site is very convenient, which only needs to add relevant logical control records. but for the existing e-commerce sites, the cost of rebuilding the system is very large. In addition, the application layer log has not formed a good standard, and there is no relevant implementation model and its user session recognition algorithm.

Client-side log relies on client data collection, and the client's proxy can return the page and time requested by the user for the server. There are usually two ways to collect data from clients, one is to install software or plug-ins on the client side, and the other is to implement the client script or cookie through embedded pages. Considering most users usually don't like to install many unnecessary software and plug-ins on the computer, the first way is limited to use. The second way is limited to use in the same way because client scripts and cookie can be banned in browsers.

Packet Sniffer can decode protocols, report statistics, automatically identify many common problems in the network, and generate management reports by observing network traffic and network configuration. Data records of Packet Sniffer can be used in a Web data model to obtain access information by listening to Web packets. Packet Sniffer data collection can be deployed on the server side, client side or proxy server side. Packet sniffer can get more information than the web log, such as the user's click stop button event. It has some advantages, such as supporting for any web server, capturing data in real time, unified processing of multi-server, and reducing the load on the web server. Except these advantages, Packet sniffer model has three disadvantages: Firstly, it is difficult to identify users and sessions. Secondly, encrypted traffic can't be handled. Thirdly, the information in URL string can't be captured. UEM5 can identify the session with high accuracy, but it needs more support from the application system, which increases the burden of the application system and has higher requirements on the performance of the application system.

To summarize, the advantages and disadvantages of Web data models are shown in Table 1.

In the rest of this paper, we discuss the 4-gram united events model in detail in Sect. 2. Simulation experiments are conducted in Sect. 3 to verify the performance of UEM4. The last part is the conclusion.

2 UEM4

Based on application layer, we propose a data model UEM4, which has the advantage of application layer record, including high quality of data collection, convenient user identification and session identification. UEM4 can be used to describe all kinds of events in the electronic commerce, and record the user's access. In an e-commerce site, the user has gone through different stages from the beginning of visiting the site to leave the site, the entire cycle becomes a web user life cycle [24]. Key events occur in the Web user life cycle of each stage, which include Login, Navigate, Depart, Search, Buy, Abandon, AddToCart, DeleteItem, and ViewCart. Each event is represented by an event type identifier. Depending on the type of event, the N tuple is used to give a different description.

2.1 UEM4 model representation

Usually, network users can be divided into registered users and anonymous users. UEM4 uses different methods to identify the two user, and the access records of the two kinds of user can be consolidated and processed. In detail, for reg-

Table 1 Comparison of advantages and disadvantages of Web data models

Name	Advantages	Disadvantages
Server log	Every one has got one	Poor quality
	Large volume	Not designed for analysis
	Not interfere with users	Error-prone pre-processing
		Hard to integrate with E-commerce data
Packet sniffer	Additional information available	Not handle encrypted traffic (SSL)
	Handle multiple Web servers as one	Not capture sub URL information
Application server log	High quality and semantic data	Must design an application server properly
	Accurate session identification	Difficult to revise codes for existing websites
	Integrate with E-commerce data	
5-gram	High accuracy	Need the application system support the ability to identify the session
United event model	Best online and offline performance	Increases the burden of the application system
Client		User privacy

istered users, the user login mechanism is used to identify the user; for anonymous users, IP and user browser type are used to identify the user, which is similar to the web log method. The difference is that the level of web logging is the page URL, and the level of UEM4 can be selected by the user themselves, which can be a page URL, commodity ID, models, brands, and also be mixed in several ways.

$$userID = \begin{cases} Session[''userID''] & Session[''userID''] \neq null \\ IP + BrowserAgent & otherwise \end{cases} \tag{1}$$

$$time = System.DateTime.Now \tag{2}$$

UEM4 is a two-dimensional table model. Each line has a record of access information, including four columns: user ID, access time, event type, and event parameter information. Either event type must include a user identifier (userID) and timestamp, as two of the elements of the N tuple. Unified event (UE) is a four tuple, which has four elements, including userID, time, type, and param. UserID means user identification, as defined in the formula 1. As we can see from formula 1, useID has two definitions, for registered users, userID is the database user identifier name for the logged in user; for anonymous users, userID is IP+ user browser type. Time means the access time, defined in the formula 2. Type and param are defined in Table 2, type means the event type, and param means the event parameter information.

Table 2 summarizes the types and parameters of all of the events and their meanings. Param means the parameter string, which has different interpretation according to the different type. For Search type, param is the search string; for Navigate type, param is a commercial ID; for other types of events, such as AddToCart, DeleteItem, ChangeQuantity, and Buy, param is a complex list; for Compound types, param is a composite event composed of multiple events. Therefore, according to the complexity of the param string, the unified event can be divided into four types, namely, the basic event, the simple event, the complex event and the combined event. The basic events are those events that have no parameters, such as Login, Depart, Abandon, ViewCart and other events; simple events are those events that have a single parameter string, such as Navigate, Search, param; parameter of complex events is a complex list, such as Buy, AddToCart, DeleteItem, ChangeQuantity. In the three types of basic, sim-

Table 2 Event types and parameters of four different events

Event	Event type	Event parameters
Basic events	Login	–
	Depart	
	Abandon	
	ViewCart	
	Customized	
Simple events	Search	The Searched term
	Navigate	The navigated item
	Customized	Customized
Complex events	AddToCart	Newly added items and their quantity
	DeleteItem	The item to be deleted
	ChangeQuantity	The item to be changed its quantity
	Buy	Purchase items in the shopping cart
	Customized	Customized
Compound events	Compound	Compound

ple, and complex, the user can customize the type of event required, that is to say, the model is extensible.

It is easy to implement UEM4 as a relational database table called united event table (UET), which is a collection of unified events, as defined in the formula 3. UE is a tuple in UET. The UE manager is responsible for capturing various events and storing event information in UET. UEM4 is a good integration of a variety of key events in e-commerce, and the model is scalable, it's easy to add custom unified events to the model.

$$\mathrm{UET} = \{\mathrm{UE}\} = \{\mathrm{UE}_1, \mathrm{UE}_2, \ldots, \mathrm{UE}_n\}, n \in N, UET \subseteq UED \tag{3}$$

2.2 User session identification

UET is just a user access that stored a variety of events, which can't be used directly in web mining algorithms. In order to be able to be used directly by the Web mining algorithm, it is necessary to carry out user identification and session identification operation, and convert the UET into the transaction database.

2.2.1 User identification

The purpose of user identification is to identify each access user, and to classify all access by user. The classification method is to sort UET by userID and time to get the record set RS. Different classification methods are used for registered users and anonymous users, respectively. For these two types of user, the UET tuples are divided into two different parts, UET_r and UET_a, as shown in formula 4. UET_r is the event set containing events of all registered users, which is defined in the formula 4. UET_a is the event set containing events of all anonymous users, which is defined in the formula 5. UseID in formula 4 is a registered user account.

$$UET = UET_a \cup \mathrm{UET_r} \tag{4}$$

$$UET_r = \{UE | UE \in UET \wedge UE.user\,ID\} \tag{5}$$

$$UET_a = UET - UET_r \tag{6}$$

For registered users, the method is to classify the events belong to the same useID into the same user, and it's easy. For anonymous users, the method is to classify the events belong to the same IP and browser agent into the same user, and it's complicated.

Suppose there are two events UE1 and UE2, the condition that can classify UE1 and UE2 into the same user is that the value of formula 7 is true.

$$(UE_1, UE_2 \in UET) \wedge UE_1.userID = UE_2.userID \tag{7}$$

2.2.2 Session identification

Session identification is to regard a continuous access of the same user as a session, accordingly, to find out all the sessions of all users. In UEM4, the session is divided into two different types: anonymous sessions and registered sessions. A sequence of united events that satisfy the following four conditions is called anonymous session:Firstly, $< \mathrm{UE}_1, \mathrm{UE}_2, \ldots, \mathrm{UE}_m >$ is unified event sequence; Secondly, for the same user, $(\forall i,j \in N)((i,j \in [1,m]) \rightarrow \mathrm{UE}_i.\mathrm{userID} = \mathrm{UE}_j.\mathrm{userID})$; Thirdly, events' interval is less than the threshold value, that's to say, $(\forall i \in N)(i \in [1, m) \rightarrow UE_{i+1}.time - UE_i.time \leq \varepsilon)$; Fourthly, the sequence is the largest. Anonymous session is similar to the Web log session identification, because RS has been sorted by userID and time, anonymous session can be obtained by timeout method [25,26] according to the time interval between two adjacent events. For a registered user, the session identification method is divided into two cases according to whether the system using UEM4 allows the same user to have multiple sessions at the same time. Boolean parameter AllowMultipleAccess is provided to indicate whether the UEM4 system allows multiple sessions at the same time as the same user. If the value of AllowMultipleAccess is true, representing the system allows the same user to have multiple sessions at the same time, the UEM4 uses the same session identification method as an anonymous user; if the value of AllowMultipleAccess is false, which means that the system does not allow more than 1 session at the same time as the same user, UEM4 uses a registration session identification method. The registration session identification method scans the RS to identify the UE of all logged events to its NextLogin event during the session identification. For the same user, since RS has been sorted by time, the NextLogin is easily identified by sequentially scanning rs. There are two modes for users to leave the site: the first mode is to click the "left" link from the site, which can record the event of Depart; the second mode is directly close the browser, which led to loss of the Depart event record. Either way, the UE before next Login is the last unified event of a user in a continuous access to the registration session.

2.2.3 User session identification algorithm

The user session list is a nested two tuple, which is divided into an internal two tuple and an external two tuple. The internal two tuple $<\mathrm{type, param})_i >$ is the sequence of events of a user's session, which (type, param) represents the type and parameter of a unified event UE. And the external two tuple is a collection of event sequences for multiple sessions. In order to explain the meaning of internal two tuples, let's suppose that there are two nested two tuples, one of which is (U1, {< (Login), (Navigate, 001), (Buy, (001100) & (002,20)),

Table 3 Timeout splitting method and Login-tag splitting method

	AllowMultipleAccess=false		AllowMultipleAccess=true	
	Anonymous user	Registered user	Anonymous user	Registered user
Timeout segmentation	√	–	√	√
Login mark segmentation	–	√	–	–

(Depart) >}), and the other one is (U2, {< (Login, >)}). The first one means that the user U1 logs in first and then browse the 001 item, then buys the 100 items of the 001 and 20 items of the 001, and then leaves the site in a session. The other user U2 has only logged events in a session.

when identifying user sessions, the GenSessions algorithm first divides the user into a set of records, and then splits the same user event. There are two kinds of event segmentation methods for the same user: timeout segmentation session identification method and Login mark segmentation method, which are shown in Table 3. The timeout segmentation session identification method checks whether the time interval between the two adjacent events of a user is timed out, and if it is out of time, it is split between the two events. Login mark segmentation checks whether the type of an event is Login, if it is Login, then The split point is located between the event and the previous event.

When the user session is identified by the GenSessions algorithm, the input is UET, and the output is a nested two tuples. The whole GenSessions algorithm is as follows.

```
Input: UET, AllowMultipleAccess
Timeout threshold:  ε ( ε >0)
Output: User session lists{(userID,<(type,param)i>)j}, i, j  ∈N
Read events from UET. Sort events by userID, time and obtain a dataset RS
Devide RS into event sets {rs} by userID
For each rs      {                              // for each event set rs
Initialize sequenceSet and sequence
rs.Read();                    // Read an event
if (AllowMultipleAccess | | rs.UserID is anonymous user)
      while (true) {     // for AllowMultipleAccess=true
//or for anonymous users, use timeout segmentation session identification
            lastRecordTime = rs.time;
                        Add (rs.type, rs.param) to sequence
            if (rs.Read()){
if (rs.time – lastRecordTime > ε )
    Add sequence to sequenceSet, and Clean sequence;
}
else{
   Add sequence to sequenceSet
        Add(rs.userID, sequenceSet) to UserSessions;
break;
} //end if
} //end while
      else
            if (rs.UserID is registered user)
while (true) {   // for registered users and AllowMultipleAccess=false, use Login mark segmentation method
Add(rs.type, rs.param) to sequence
            if (rs.Read()){
  if (rs.type= “Login”)
Add sequence to sequenceSet，and clean sequence;
}
else{
      Add sequence to sequenceSet
                  Add (rs.userID, sequenceSet) to UserSessions;
break;
} //end if
}//end while
} //end for
return UserSessions
```

Logging in the application layer can solve a series of pre-processing problems of Web log. User and session identification are more accurate and can be used as a good data source for analysis. Browse, purchase and other types of event records are well integrated in the same model. The user session set obtained by the user session identification can be saved to the database or used directly by the mining algorithm.

2.2.4 Web mining analysis

UEM4 can achieve multi-dimensional multi-level mining algorithm by integrating browsing records and purchase records. Taking the FP algorithm [27] as an example, we demonstrate how to use UEM4 to change the existing algorithm to the multidimensional algorithm UEMFP. The algorithm is as follows.

```
Input UserSessions
Output patterns
The interpreter replaces the complex events of UserSessions (such as Buy) as a simple event set;
Scan UserSessions and count the number of items, get the list head_tbl by the number of order from large to small
order;
Create root node;
for each (session in UserSessions.sequenceSet)   // Session for all users
Suppose session = s1s2 ··· sn , here si = (type, param)i, i ∈ N, i ∈ [1, n] is two tuple
    insert_tree(session);                  // Insert the session into the tree by FP
next
for each header_item in head_table
 //s is two tuple (type,param),  the item in the header of the FP tree
    Construct subSessions for header_item;
    tmps=UEMFP(subSessions);   // Recursive call UEMFP, Return pattern set tmps
    if tmps!=null then
        for each tmp in tmps
        tmp.Add(header_item);     // Adds a header item to each mode of the TMP set
        patterns.Add(tmp);          // Add tmp mode to patterns mode
    next
    end if
  patterns.Add({header_item});   // Header table entry as an independent mode
next
return patterns;
```

FP algorithm is a typical association rule algorithm without frequent item sets. There are two differences between UEMFP and FP: (1) the tree node of UEMFP stores two-dimensional information, which can be used to mine multidimensional frequent patterns. (2) the event parameters of UEMFP are multi-level, so it can be used to mine multi-level frequent patterns.

As we can see from the algorithm, the change from FP algorithm to UEMFP algorithm can be done just by modifying URL match of FP algorithm to the matching of two tuples because there is just a little change about the information node tree, and the process of UEMFP algorithm is the same as FP algorithm. Therefore, UEMFP can correctly mine all frequent patterns. The UEM4 model is not only suitable for association rules algorithm, but also can be used to transform the sequence pattern algorithm and the clustering algorithm, which can also produce the multi-dimension and multi-level level Web mining algorithm.

3 Simulation experiments

In order to verify the performance of insertion events and user session identification of UEM4, simulation experiments of Web log data and UEM data in the intelligent e-commerce network developed by our laboratory is carried out. All programs were written in C#, and all experiments were performed on a PC machine with i7-4500U CPU, 8 MB memory, and a Win8 operating system.

3.1 Inserting events performance experiments

The experiments verify the performance of inserting uniform events of UEM4 model. Due to the time of inserting a record is very short, the experiments tested the insertion time of a large amount of data, and we take its average value as a record insertion time. The performance of the model is tested in two cases, which are the insertion of different events and the insertion of new records under the number of different events. Figure 1 shows the results of inserting 10,000–50,000 events into the UET table with UEM4 and UEM5 respectively when the UET table is empty. As we can

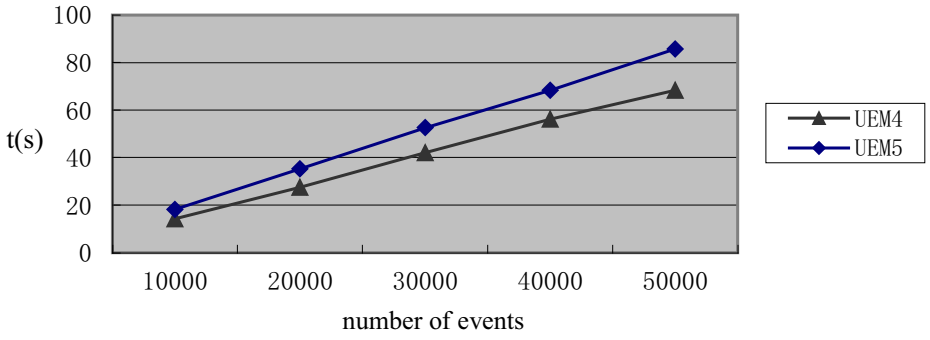

Fig. 1 Performance test of inserting events

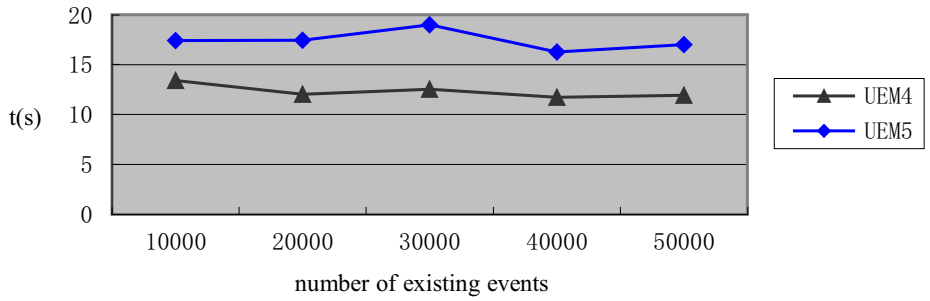

Fig. 2 Performance test of inserting events under different number of current events

see from the Fig. 1, the UEM4 and UEM5 model inserts the record in a linear relationship with the number of inserted records. The average time for UEM4 to insert a record is 1.43 milliseconds, while UEM5 is in the order of 1.82 milliseconds. Like UEM5, UEM4 has very good performance in inserting records, and is suitable for adding uniform events online without affecting the performance of the application.

Figure 2 shows the test results that 10,000 events are inserted into the UET table in the case of a different number of existing records in the UET table. As can be seen from the Fig. 2, UEM4 and UEM5 are not affected by the amount of data recorded by the existing UET. It can be seen that both UEM4 and UEM5 are scalable, and even if a large amount of data has been stored, it does not affect the performance of the online insertion record. By using the data warehouse technology, the real-time UET table is stored as a ETL in the data warehouse, which can not only accelerate the performance of online insert record, but also realize the goal of mining analysis and transaction independence.

3.2 Precision rate test of user session identification algorithm

This experiment tests the accuracy rate of the GenSessions algorithm in the session recognition, that's to say, to identify the degree of agreement between the session obtained by the GenSessions algorithm and server session. We construct a set of 5000 session sets, and then use the web log model, UEM4 (AllowMultipleAccess) model and UEM5 model to record respectively, the precision of session identification of these three models under different timeout threshold can be measured.

As shown in Fig. 3, UEM5 has the highest accuracy rate, the accuracy rate of UEM4 is centered, and the accuracy rate of web log is the lowest. Due to the use of ex-ante identification method, the accuracy of UEM5 is 100%, but it has a drawback that the application system needs to support the ability to identify the session, which will increase the burden on the application system. The accuracy of UEM4 session identification is discussed in two cases: when AllowMultipleAccess is false, identification of UEM4 registered users is correct, but the users need to use the timeout segmentation method, and the timeout segmentation method treats two cross access sessions as the same session, or a session split as the two session, which affects the accuracy of session identification; when AllowMultipleAccess is true, UEM4 registered users also need to adopt the timeout segmentation method, accordingly, the session identification accuracy of UEM4 is reduced, but there is still more than 80% accuracy. The lowest accuracy of web Log is because it contains a lot of redundant access, such as taking the Html file embedded as a user access.

3.3 Performance test of user session identification algorithm

In this experiment, the performance of GenSessions algorithm is tested, and the performance of session identification algorithm is compared with that of web log and UEM5. Based on the session set constructed in 3.2, assuming a timeout threshold of 10 seconds, we tested the session recognition

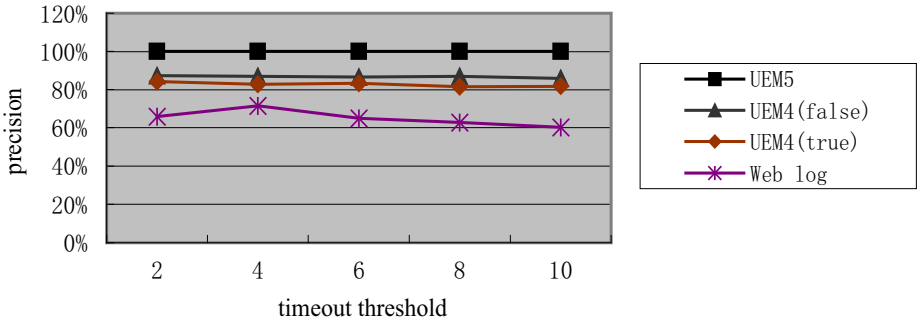

Fig. 3 Precision test on user session identification algorithm

Table 4 Performance test on user session identification algorithm

Session volume	UEM5	UEM4 (false)	UEM4 (true)	Web log
1000	0.156	0.462	0.458	16.06
2000	0.232	0.772	0.828	32.00
3000	0.344	0.964	1.014	48.30
4000	0.462	1.49	1.492	64.00
5000	0.676	1.706	1.712	80.80

time of web log, UEM4 (different AllowMultipleAccess) and UEM5 respectively under different data volumes. The results are shown in Table 4.

As shown in Table 4, both UEM4 and UEM5 have a high performance in session identification. UEM5 has the highest offline analytical performance due to the use of ex-ante identification sessions; UEM4 has a good performance and has a nearly linear time complexity. Due to the need for more complex pre-processing of log files, the web log session identification has poor performance.

In general, UEM5 has a high accuracy rate, the best online and offline performance, but the application system needs to be able to support session identification for UEM5. UEM4 does not require the application system to support session identification, and also has a good accuracy and performance of session identification. Which model to choose depends on the specific requirements of the user's UEM system. UEM4 and UEM5 have greatly improved the user access data provided by the web log source, which provides a reliable and convenient data source for intelligent recommendation.

4 Conclusion

In order to solve the problems of web data source, this paper proposes a web data source model UEM4 based on application layer record. The performance of the model is verified by simulation experiments, and the performance is compared with that of UEM5 and web log. Experimental results show that: (1) web log has the worst performance among the three model. (2) like UEM5, UEM4 has four advantages: firstly, it is more accurate and convenient user session identification than web log, and can solve the problem of a series of web log pre-processing; secondly, it is well integrated with the purchase, browsing and other types of events; thirdly, it is compatible with the existing web mining algorithm; fourthly, it supports multi-dimensional and multi-level web mining analysis. (3) UEM5 has a higher accuracy rate than UEM4, but for UEM5, the application system needs the ability to support session identification, which needs higher requirements on the performance of the application system; UEM4 does not require the application system to support session identification, and also has a good accuracy and performance of session identification. Which model to choose depends on the specific requirements of the user's UEM system.

In summary, UEM4 model provides a high quality data source for web mining algorithm, and has a good recognition accuracy and performance. The data records of various e-commerce can be easily added in the model. The new Web data source model is proposed, which provides a high quality data source for the intelligent e-commerce site, and thus improves the quality of intelligent service.

References

1. Tourassi, G., Yoon, H.J., Xu, S.H., Han, X.S.: The utility of web mining for epidemiological research: studying the association between parity and cancer risk. J. Am. Med. Inf. Assoc. **23**(3), 588–595 (2016)
2. Zhao, J.S., Zhao, S.Y.: Business analytics programs offered by AACSB-accredited U.S. colleges of business: a web mining study. J. Educ. Bus. **91**(6), 327–337 (2016)
3. Panda, B., Tripathy, S.N., Sethi, N., Samantray, O.P.: A comparative study on serial and parallel web content mining. Int. J. Adv. Netw. Appl. **7**(5), 2882–2886 (2016)
4. Patil, Swapnil S., Khandagale, Hridaynath P.: Enhancing web navigation usability using web usage mining techniques. Int. Res. J. Eng. Technol. **4**(6), 2828–2834 (2016)
5. Asha, K.N., Rajkumar, R.: Survey on web mining techniques and challenges of e-commerce in online social networks. Indian J. Sci. Technol. 9(13) (2016)
6. Siddiqui, A.T., Aljahdali, S.: Web mining techniques in e-commerce applications. Int. J. Comput. Appl. **69**(8), 39–43 (2013)
7. Xu, Z., Luo, X., Zhang, S., Wei, X., Mei, L., Hu, C.: Mining temporal explicit and implicit semantic relations between entities using

web search engines. Future Gener. Comput. Syst. **37**, 468–477 (2014)
8. Satish, B., Sunil, P.: Study and evaluation of user's behavior in e-Commerce using data mining. Res. J. Recent Sci. **1**, 375–387 (2012)
9. Jafari, M., Sabzchi, F.S., Rani, A.J.: Applying web usage mining techniques to design effective web recommendation systems: a case study. ACSIJ Adv. Comput. Sci. Int. J. **3**(2), 78–90 (2014)
10. Kathirvel, P.: A survey on online shopping recommendation based on customer transactions. Int. J. Sci. Eng. Technol. Res. **4**(3), 564–566 (2015)
11. Asha, K.N., Rajkumar, R.: Survey on web mining techniques and challenges of e-commerce in online social networks. Indian J. Sci. Technol. **9**(13), 1–5 (2016)
12. Tesfaye, B., Atique, S., Elias, N., et al.: Determinants and development of a web-based child mortality prediction model in resource-limited settings: a data mining approach. Comput. Methods Progr. Biomed. **140**(3), 45–51 (2017)
13. Iyer, N., Dcunha, A., Desai, A., Jain, K.: Survey on online recommendation using web usage mining. Int. J. Comput. Sci. Inf. Technol. **6**(2), 1465–1467 (2015)
14. Xuan, J.Y., Luo, X.F., Zhang, G.Q., Liu, J., Xu, Z.: Uncertainty analysis for the keyword system of web events. IEEE Trans. Syst. Man Cybern. Syst. **46**(6), 829–842 (2016)
15. Ambili, P.S.: Varghese Paul. Enhanced user personalization by web log mining and link structure display. Middle-east. J. Sci. Res. **24**(3), 628–631 (2016)
16. Alessandra, M., Piercesare, S.: Statistical analysis of complex and spatially dependent data: a review of object oriented spatial statistics. Eur. J. Oper. Res. **258**(2), 401–410 (2017)
17. Zhang, W., Pan, X.F., Yan, Y.B., Pan, X.Y.: Convergence analysis of regional energy efficiency in china based on large-dimensional panel data model. J. Clean. Product. **142**(2), 801–808 (2017)
18. Jana, M., Jan-Philipp, M., Karsten, R., Fabian, E.: Retrieving chromatin patterns from deep sequencing data using correlation functions. Biophys. J. **112**(3), 473–490 (2017)
19. Mahajan, R., Sodhi, J.S., Mahajan, V.: Usage patterns discovery from a web log in an Indian e-learning site: a case study. Educ. Inf. Technol. **21**(1), 123–148 (2016)
20. Parthiban, P., Selvakumar, S.: Big data architecture for capturing, storing, analyzing and visualizing of web server logs. Indian J. Sci. Technol. **9**(4), 1–9 (2016)
21. Girdhar, Palak, Malik, Vikas: A study on detecting packet using sniffing method. J. Netw. Commun. Emerg. Technol. **6**(7), 45–46 (2016)
22. Zou, X.Y.: 5-gram united event model. Appl. Mech. Mater. 1319–1322 (2010)
23. Kohavi R.: Mining e-commerce data: the good, the bad, and the ugly. In: Provost, F., Srikant R. (Eds.) Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM Press: USA, pp. 8–13 (2001)
24. Ha, S.H.: Helping online customers decide through web personalization. IEEE Intell. Syst. **17**(6), 34–43 (2002)
25. More, A., Joshi, P.P.: Survey on inferring user image-search goals using click through logs. Int. Res. J. Eng. Technol. **3**(3), 149–152 (2016)
26. Liao, Z., Song, Y., Huang, Y.L., et al.: An effective segmentation of user search behavior. IEEE Trans. Knowl. Data Eng. **26**(12), 3090–3102 (2014)
27. Gaikwad, Pravin, Kulkarni, Jyoti: Inconsistency extraction using advanced FP-growth algorithm. Int. J. Comput. Appl. **105**(5), 6–10 (2014)

Xinyao Zou received the Ph.D. degree from South China University of Technology, China, in 2009. Since 2009, she has been with the Guangdong AIB Polytechnic College, where she is currently an associate professor. Her research interests include big data analysis, support vector machine, neural networks, pattern classification, web mining, and small sample electronic devices reliability assessment.