USER BEHAVIOUR ANALYSIS IN STRUCTURED E-COMMERCE WEBSITE

[1]Shashank Kodate, [2]V.Vamshi Goud, [3]Taran Patel, [4]B.Nithin Reddy, [5]J.Himabindu Priyanka

[1,2,3,4]Student, [5]Associate Professor
Department of Computer Science and Engineering
St.Martin's Engineering College, Hyderabad, Telangana, India

***ABSTRACT*: We are living in the world of growing technology. Everything is accessible at a touch of fingertip. E-commerce is already a huge thing. Many people prefer to buy things online. It is important to understanding users' behaviour in order to adapt e-commerce system to meet the customer's requirements. The user behaviour data is registered in web server logs. Data mining techniques are applied on such information for the analysis of user's behaviour. During the session, to identify more complex behavioural patterns, we incorporate a view of the actions performed by the user. We propose linear-temporal logic model checking method for the analysis of web logs in this paper. Here, we apply different predefined queries on the data during the session for identifying behavioural patterns performed by the user.**

IndexTerms: Data Mining, E-commerce, Data Analysis, User Behaviour

1. INTRODUCTION

In today's ever connected world, the way people shop has changed. People are shopping more over the Internet instead of going traditional shopping. E-commerce provides customers with the opportunity of browsing endless product catalogues, comparing prices, being continuously informed, creating wish list and enjoying a better service based on their individual interests. The e-commerce market is highly competitive; the possibility of a customer to move from one e-commerce to another is high when their requirements are not satisfied. Therefore, it is required to understand consumer's behaviour, the way they navigate through the website and the reasons that motivate them to purchase a product or not. By analyzing the behavioural patterns it will help the e-commerce websites to deliver services to customers with a personal touch and increasing benefits.

However, discovering customer' behaviour and the reasons that guide their buying process is a complex task. E-commerce websites provide customers with a wide variety of navigational options and actions: users can freely move through different product categories, follow multiple navigational paths to visit a specific product, or use different mec0hanisms to buy products. The events generated by each user commonly known as *click-streams* are stored in sequence in the web server logs. The users' behaviour is stored in these logs which must be analyzed to define a pattern. A correct analysis is used to improve the website's contents and structure to understand the interest of user in specific products.

Data mining techniques are used to discover the patterns; its main goal is to discover the user behaviour and interest. Different techniques are used by the e-commerce websites, such as classification techniques, clustering,association rules or sequential patterns. In many application domains these techniques are used in conjunction with process mining techniques. Such techniques are part of the business intelligence domain and apply specific algorithms to discover hidden patterns and relationships in large data sets.

We propose Linear-Temporal Logic model checking techniques as an alternative to data mining techniques. The goal is to discover and analyze customers' behavioural patterns by using temporal logic formulas to describe such behaviours. Events can be user or system actions performed when a client visits a product or product category page, when he or she adds a product to the wish list, when the search engine is used, etc. The business analyst can use a set of (predefined) temporal logic patterns to formulate queries that could help him to discover and understand the way clients use the website. Considering the website structure and contents as well as the different types of user's actions, these queries can check the existence of complex causality relationships between events contained in the client sessions. From the tool point of view, the necessity of having control on the way the checking algorithms are applied, as well as the disappointing performance results we obtained when using some model checking tools at our disposal, mainly when used against big models, drove us towards the interest of developing a specific model checking tool. We did it using the SPOT libraries for LTL model checking.

2. LITERATURE SURVEY

Mining E-Commerce Data:

Web server logs are the source for mining data. Moreover, many events are never logged in web server logs, limiting the source of data. To eradicate the problems related to server logs the e-commerce website must be architecture properly. Even with a good architecture, still there are problems that are hard to resolve.

E-commerce sites can generate great data for mining, containing all the right ingredients. However, the naive approach of using web logs is insufficient for many business questions and additional data must be collected and conflated. There are several challenges that make data mining hard, and we hope to see them addressed by the community.

Web Mining in E-Commerce:

E-commerce has provided a cost efficient and effective way of doing business. Web mining is the application of data mining technique to discover useful information from World Wide Web. Web mining is applied to ecommerce to know the browsing behaviour of customers, to determine the success of marketing efforts, to improve the design of e-commerce web site and to provide personalized services. Here we discuss web mining in e-commerce, the categories of web mining, pattern discovery techniques to find out interesting patterns, issues of web mining in e-commerce and application of web mining in e-commerce.

.The growth of World Wide Web and technologies has made business functions to be executed fast and easier. As large amount of transactions are performed through ecommerce sites and the huge amount of data is stored, valuable knowledge can be obtained by applying the Web Mining techniques. Using Web Mining, companies can understand customer behaviour, improve customer services and relationship and measure the success of marketing efforts. In this paper, we have discussed web mining in e-commerce, categories of web mining, pattern discovery, issues and application of web mining in e-commerce. The extension in web mining research will lead to success of e-commerce sites and also it will improve the services for customers.

3. OVERVIEW OF THE SYSTEM

I. MODEL CHECKING TO ANALYSE EVENT LOGS

A. Basics on Linear Temporal Logic and Model Checking:

Usually, a program execution is viewed as the sequence of state transformations moving from a given initial state to a final state. We are considering a program state in terms of Boolean formulas over a set of atomic propositions A. Each atomic proposition is assumed to mean the truth of some property. The execution of a program sentence means that the values of some atomic propositions can change, moving from a Boolean formula to another one. Therefore, talking about the program behaviour requires being able to talk about program states and also state evolutions. Besides using propositional logic to talk about states ((v1^v2) _:v3)), temporal logics add temporal (causality) operators, such as next (l) and until ([). As a way of simplifying the writing of temporal logic formula, additional operators, such as eventually (_) and always (_), are used (which are defined in terms of the former ones). A program execution can be seen as the ordered sequence of the Boolean formulas satisfied by the successive states the program reaches. This execution order is considered as the temporal structure. Having the finite set of possible program executions allows the analysis of the program behaviour. To carry out the analysis of data, model checking techniques have been developed.

We are going to use Linear-Temporal Logic, which defines logic for (infinite) traces to program executions, with the approach followed in. Let A be a given set of atomic propositions. The formal syntax of the set of correct LTL formulas is recursively defined as follows: 1) every a 2 A is a LTL formula 2) if f and g are LTL formulas.

From a semantic point of view a LTL formula must be interpreted over runs of a program. A finite state program is a tuple PA = (S;!; s0) where S is the finite set of program states, s0 2 S is the initial state and !_ S _ 2A _ S is the transition relation, which describes the actions available at a given state and the state transitions corresponding to the execution of such actions. A run of PA is an infinite sequence _ = $\rho = s \longrightarrow s \longrightarrow s \longrightarrow s \longrightarrow \ldots$ where $(s, x, s) \in \rightarrow$for any j _ 0. Since we are interested in talking about log traces, for a run _, let us define tr(_) = x0 _ x1 _ x2 : : : as its trace. Notice that the trace is an infinite word over the alphabet 2A, of the possible subsets of A. In the following, we will denote as _i the suffix of _ starting a i (notice that_ = _0).

B. Applying model checking to event log analysis

Let us now move from this conceptual framework to the case of event log analysis. A trace can be considered as the run of a program, where the set of atomic propositions corresponds to the set of events or event attributes

Each row corresponds to an event, where columns correspond to event attributes (the elements of a column can be considered as instances of the same attribute class). Events are ordered according to the time each one took place. We are considering the set of ordered events corresponding to the same session, those with the same Id, as a program run (*trace*). We associate an atomic proposition to each attribute value appearing in events, calling to the whole set. The sequence of events of a trace can be seen as a sequence of elements belonging to 2A.

In order to enable the use of model checking techniques, traces, which are finite, must be transformed into infinite ones. To achieve this, there are different proposals in the literature. The most commonly used is the addition of a final loop with a dummy *End* event to every terminal state. Doing so, traces are now infinite and model checking can be applied. In fact, for each trace we have added a transition to a dummy final state, as well as a self-loop for this state, both labelled with the End event and the conjunction of all the atomic variables, negated. The model checker must take into account that transformation in order to avoid interpretation mistakes.

B. Model checker implementation

In order to enable the application of LTL-based model checking on event logs, we have developed a log analysis system composed of two main components offered as REST Web Services. More details about the model checking analysis architecture can be found in. First, the *Model Generator* uploads and transforms the input log file, specified as a Comma Separated Values (CSV) file, so that it can feed the checker. Second, the *Model Checker*, which loads and analyses the previous file. The model checker has been implemented using the SPOT libraries for LTL model checking.

Besides usual temporal logic formulas, the tool provides with the possibility of defining sets of variables and macros to make easier the writing of LTL formulas. Subsets of variables can be defined in multiple ways: by enumeration, as a range of identifiers or by means of regular expressions. Once a set VAR is defined, the appearance of "?VAR" in a formula means that the formula must be evaluated for each element belonging to "VAR", Thereby, as many formulas as elements in the set VAR are automatically checked by the tool. Similarly, macros can be defined on these sets as a logical OR or AND between all the elements on the set. For example the macro " OR or var VAR" indicates that the appearance of "?or var" is replaced in the formula by the logical "OR" of all the elements in the set "VAR". Also, some formulas can be defined with a given name, avoiding the same formula to have to be written more than once.

D. Applying model checking to the analysis of e-commerce websites

Users of any e-commerce site navigate through the different web pages executing two types of interactions: either a GET operation to retrieve some information or a POST operation, usually requesting the website to execute some action, such as adding some product to the cart, buying some product, logging in, etc. The website log records such actions together with some associated information, such as the IP the user is connected from or the time at which the interaction occur, for instance. Some of these actions correspond to events that are common to any e-commerce website such as the ones related to visiting the sections containing products. Therefore, a general way of classifying the events in the web logs according to the product categorization can be proposed. From now on, we are going to describe the proposed approach to relate the website structure and the events in the log, to identify meaningful set of events, and to ask for behavioral usage patterns using model checking based on the previous classification.

To apply model checking techniques, we are going to associate temporal logic formulas to events, which will allow us to see the log as a Kripke structure representing the model to be analyzed. For that, we are first going to define the set of atomic propositions, and transform events into conjunctions of such variables. This will be done during the pre-processing phase, whose output will be the model representing the log. Figure 1 show the typical structure used in e-commerce websites to organize and categorize products. Similar tax-anomies have been proposed by different authors but including only main sections. From the homepage (level 0) different sections can be accessed (level 1). Two different types of sections can be distinguished.

1) *Main sections*, which correspond to the main product categorization. These sections allow access to all products. In general, the product categorization is usually disjoint, but this is not mandatory: in some e-commerce websites the same product could belong to different sections.

2) *Secondary sections*, which provide a secondary categorization of the website products, whose objective is to allow the access to a subset of products with some common and specific features. Unlike the previous case, not all products must be accessible from these secondary sections. Furthermore, we can distinguish two different types of secondary sections depending on whether products in such sections are permanently or temporarily added to the section. An example of sections with temporary products would be offers or sections with new products that are periodically renewed. An example of secondary sections with permanent links would be sections where you can access products by manufacturer, theme, etc.

Independently of its type, each section is usually split into several subsections to refine product classification. Each e-commerce website establishes its own organization (categories, levels, etc.).

Since each section corresponds to a different web page, each section has a unique URL that enables to identify the visiting event. The website designer knows the relation between the URL and the associated sections in the structure shown in Figure1.

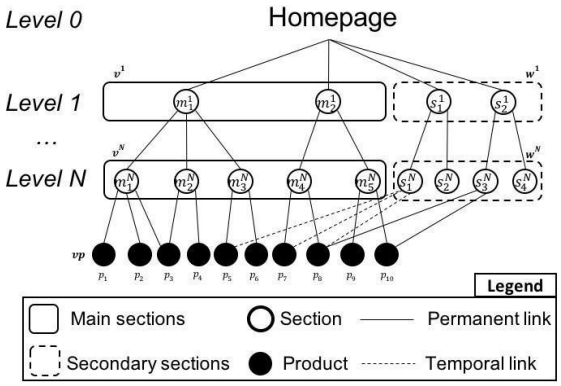

Fig. 1. Typical structure used to categorize products in an e-commerce website.

In many cases this relation is clearly reflected in the URL itself. For instance, in the case we are studying in this paper, the second event is related to visiting relative URL /, belonging to the main level-2 category which is a subcategory of main level-1 category. When not, the system manager must establish the mapping between web-pages and atomic propositions. Similarly, for POST requests the same approach can be used by assigning to each POST request a specific event type.

MODULES

Data Pre-processing

Data pre-processing is the initial step of web usage mining analysis. Therefore, in order to enable the analysis, raw logs must be pre-processed to discard uninteresting requests, to identify user sessions and to prepare the log to enable its analysis.

The pre-processing step can be split into three main phases. The first two are common to any web usage mining project. The third one is introduced to prepare the log contents for applying the used model checking techniques. Let us describe that phases in more detail.

A. Log cleaning

The objective of this phase is to remove undesired records that may distort the results of the analysis. For that, the following steps are carried out:

- Removing automatic requests such as the ones performed by robots, spiders and crawlers. To do that, the IPs requesting the robots.txt file and the requests with a user agent belonging to automatic requests are deleted. Furthermore, requests corresponding to IPs demanding for a large number of pages in a short period of time are also removed since it can be assumed that they are performed by automatic tools.

- Deleting requests with erroneous status codes (4*xx* and 5*xx* codes). Since we are interested in navigational pat- terns, erroneous requests are not interesting in this regard.

- Discarding requests of irrelevant HTTP methods. Only GET and POST requests have been considered since they are the unique directly requested by users.

- Deleting requests asking for multimedia contents, since these requests are automatically requested by the browser. After this first phase the log has been reduced to 5, 875, 479 records, a 68.26% of the original size.

B. User identification and sessionization The aim of this phase is to group the events belonging to the same session (in terms of process mining, we are establishing the traces of processes). For that, we have used a heuristic where a user session is composed of those events generated from the same IP address in a period of 25.5 minutes. This concrete value has been typically used in web usage mining approaches and it is consistent with the behaviour we have observed in the log. As a result of this phase, 144, 330 user sessions have been identified.

C. Log preparation

The aim is to prepare the log file to feed the model checker. For that two types of actions are performed. On the one hand, in the *categorization* sub-phase each record is analyzed to identify high-level events and to extract meaningful information. On the other hand, in the *simplification* sub-phase, log contents are reduced to increase the effectiveness of the model checking techniques.

II. Identifying Users' Behavioural Patterns

Next we are going to detail the process carried out to analyze and identify behavioural patterns from the logs. Prior to the analysis we have defined a set of variables and macros based on the sets identified. They are enumerated below, according to their equivalence with the sets proposed.

Variables. Let ?M1 denote M^1, ?M2 denote M^2, ?M denote $M^1 \cup M^2$, ?S1 denote S^1, ?S2 denote S^2, ?S denote $S^1 \cup S^2$, ?V1 denote V^1, ?V2 denote V^2, ?V denote $V^1 \cup V^2$, ?V_BAR denote V, ?W1 denote W^1,?W2 denote W^2, ?W denote $W^1 \cup W^2$, ?W_BAR denote W, ?VW1 denote $V^1 \cup W^1$, ?VW2 denote $V^2 \cup W^2$, ?VW denote $V \cup W$ and ?VW_BAR denote $V \cup W$.

4. METHODOLOGY.

A. Usage patterns related to main sections

Main sections are the main resources used to navigate through the website. That means that main sections are preferred to secondary ones. Let us analyze if users prefer to access to level-1 or level-2 sections.

Query 1 *What are the most visited main sections when taking into account both level-1 and level-2?*

The query "♦(?OR V BAR ?M1)" counts, for each main section *m M*, the number of sessions where *m* and any element of *V* is visited. *Are level-1 main sections preferred to level-2 main sections?* The query "♦ (Visit main section L1 ? M1)" counts for each main section *m M*1 the number of sessions visiting it, while "♦ (Visit main section L2? M1)" counts, for each *m M* 1, the number of sessions visiting its subsections.

Query 2 *Is the access to level-2 main sections homogeneous?* The query "♦ ?V2" counts, for each *m* M^1 and each m^j M^2 the number of visits to each level-2 section.

B. Usage and navigational patterns

Next, we are going to explore some navigational patterns that illustrate users' preferences when browsing the website. Specifically, we analyse if sections are visited as the first option and if they are visited in an exclusive way.

Query 3 What main or secondary sections are preferred by users as the first step to look for products?
The query "(: ?OR VW BAR) [(?OR V BAR ^ ?M1)" identifies, for each m 2 M1, the sessions where the own m or any of its subcategories is visited, while no other section has been visited previously. For secondary sections an analogous query is used replacing M1 by S1 and OR V BAR by OR W BAR.

Query 4 Are sections visited exclusively?
The query "_(?OR V BAR ^ ?M1) ^ _(?OR VW BAR ! ?M1)" is executed to answer the previous question. For each m 2 M1, the first part of the query checks that m or any of its subsections is eventually visited, while the second part imposes that always that a section is visited, the visited section is m (since m ^ m0, being m0 2 M1, is not possible). As in the previous case, an analogous query is executed to analyse secondary sections replacing M1 by S1 and OR V BAR by OR W BAR.

C. Behavioural patterns related to the buying process

Regarding the buying process there are two specific actions that we are interested in. First, user sessions showing interest in acquiring a specific product. That corresponds to the events of adding *a product to the wish list and to the cart*. In this regard, it is important to identify the sections visited just before such events happen. This way we could identify those sections that help users to find interesting products allowing correlating such information with different access patterns. Second, sessions that buys some products, that is, sessions where the event *Buy products in the cart* happen. In this regard, it is important to analyze the relation between showing interest in a product and purchasing it.

First, we are interested in knowing from which sections the products are added to the cart or to the wish list.

Query 5 Which are the sections visited just before adding a product to the wishlist or to the cart?
The query "_(?OR V BAR ^ ?M1) ^ l((: ?OR VW BAR) [(Add product to the cart _ Add product to the wishlist)))" is performed for main sections and analogous one replacing (?OR V BAR ^ ?M1) by (?OR W BAR ^ ?S1) for secondary ones. Thereby, the query identifies for each main section (m 2 M1) and each secondary section (s 2 S1) whether products are added to the cart or to the wishlist from that section m or s or from some of their subsections.

Query 6 For that, we have repeated the previous query adding that in the future the event Buy products in the cart happens, that is, adding "^ l_ Buy products in the cart" to the query. This way sessions where the previous patterns happen and some products are later purchased are identified.

5. CONCLUSIONS AND FUTURE WORK

In the case of open systems, where the sequences of interactions (stored as system logs) are not constrained by a workflow, process mining techniques whose objective is to extract a process model will usually provide with either over fitting *spaghetti* models or under fitting *flower* models, from which little interesting information can be extracted.

A more flexible approach is required. We apply LTL-based model checking techniques to analyze e-commerce web logs. To enable this analysis, we have proposed a common way of representing event types and attributes considering the e-commerce web structure, the product categorization and the possibilities of users to navigate throw the website according to such organization.

From this structural point of view, the paper proposes a set of query patterns, translated into LTL formulas, which are of interest for the domain of electronic commerce The answers to the queries, in terms of the number (or percentage) of traces satisfying the corresponding formula, allows to extract interesting correlations among sequences of events, which can be interpreted in terms of users' behaviour. Among the wide set of possible behaviours, we have concentrated on finding how the different website sections are visited and which navigational patterns are related to buying actions.

The analysis carried out has allowed us to identify several issues and to propose improvements regarding the product categorization and the organization of some of the website sections, which have been transferred to the enterprise managers. Although the paper is strongly related to that website, the proposed approach is general and the methodology is applicable to structured e-commerce websites. The first phase of the methodology, the pre-processing phase, is the one which is specific for each e-commerce website, since it depends on the specific system log and, meanwhile the analysis technique and the queries can be completely reused.

On the other hand, the analysis in the paper has been made for a log corresponding to two months of use. However, the proposed method is directly applicable to much bigger logs, since both the method and the tool scale very well: it can be executed in parallel, deploying different parallel servers with different parts of the log and executing the queries in parallel. As a near future work we want to provide the analysis tool with a graphical interface, for both the input of properties to be analyzed and the output of results, with the aim of facilitating its use for non-technical staff, providing with an abstraction level hiding the LTL formalism. We also plan to extend the set of studied patterns in order to analyze more behavioural patterns and to facilitate their automatic discovery. For that, a side-by-side work with specialists of the problem domain is required in order to define a set of interesting queries as wide as possible. User's information would allow us to study patterns and correlate results with demographic information; while, online reviews would allow us to analyze customer's models are presented. Finally, we would like to consider the extension of the approach to consider time constraints between the events in the line shown.

BIBLIOGRAPHY

[1] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications." Hingham, MA, USA: Kluwer Academic Publishers, Jan. 2001, vol. 5, no. 1-2, pp. 115–153.

[2] N. Poggi, D. Carrera, R. Gavalda, J. Torres, and E. Ayguade´, "Char-acterization of workload and resource consumption for an online travel and booking site," in *Workload Characterization (IISWC), 2010 IEEE International Symposium on*. IEEE, 2010,

[3] R. Kohavi, "Mining e-commerce data: the good, the bad, and the ugly," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001,

[4] W. W. Moe and P. S. Fader, "Dynamic conversion behavior at e- commerce sites," *Management Science*, vol. 50, no. 3, pp. 326–335, 2004.

[5] G. Liu, T. T. Nguyen, G. Zhao, W. Zha, J. Yang, J. Cao, M. Wu, P. Zhao, and W. Chen, "Repeat buyer prediction for e-commerce," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 155–164.

[6] J. D. Xu, "Retaining customers by utilizing technology-facilitated chat: Mitigating website anxiety and task complexity,"*Information & Man- agement*, vol. 53, no. 5, pp. 554 – 569, 2016.

[7] Y. S. Kim and B.-J. Yum, "Recommender system based on click stream data using association rule mining," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13 320–13 327, 2011.

[8] R. Kosala and H. Blockeel, "Web mining research: A survey," *SIGKDD Explor. Newsl.*, vol. 2, no. 1, pp. 1–15, Jun. 2000.