Fast recognition and application of Web users' behavioral patterns

Tomáš CHOVAŇÁK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xchovanak@fiit.stuba.sk

Abstract. Understanding of website user behavior is a crucial assumption for improving the website and user experience with it. Typical and repeating features of behavior during user's visit of website can be represented through behavioral patterns. In this work we represent behavioral patterns as frequent itemsets of actions frequently performed by users in their browsing sessions. We propose a method for behavioral patterns recognition combining global patterns with patterns specific to groups of similar users. Proposed method was evaluated indirectly through recommendation task. We performed several experiments over data from e-learning and news domains. Our results clearly show that combination of common global patterns and specific group patterns reaches higher prediction precision than its components used individually. Inclusion of group patterns also brings only constant computational load, which supports its maintenance in production usage.

1 Introduction and related work

Understanding of behavior of website users is crucial for the site personalization and adaptation of its content and structure. Every user is unique and his actions subject to his actual aim, context etc. But when we look at behavior of multiple users together, we would be able to observe some regularities and actions typical for specific situations. These regularities are in general known as the behavioral patterns. The behavioral patterns may be modelled in various ways as frequent itemsets, frequent sequences of actions or association rules [2].

These patterns can be applied to user groups of various sizes. Smart detecting of groups of users with similar behavior and their typical patterns may lead to better understanding of users' intentions.

We believe that identification of communities of similarly behaving users will help to discover specific behavioral patterns, which cannot be recognized on the global level. Using knowledge about both types of behavioral patterns and even their combination will enhance their quality and usability in different applications. For this reason, we propose an innovative method for identification of global and also group behavioral patterns and their mutual combination specialized for recommendation of user's future visited pages.

At first, our method uses data stream clustering algorithm for segmenting active users to several groups according their actual behavior. Several algorithms were proposed for this task [1,5,8]. All these algorithms consist of two (online and offline) components. Online component is fast and manages compact representation of current data (data grid [5], microclusters [1], other [8]). Offline component is computationally more expensive. On demand, it applies batch clustering approach (e.g. k-means) to obtain final clusters from the compact representation.

Next, our method uses algorithm for mining global and group behavioral patterns. We focus on algorithms mining only frequent closed itemsets, which are complete and not redundant representations of all frequent itemsets. Existing algorithms can be classified according to window model they use [4]. Landmark window contains all items from start of the stream [9]. It cannot handle concept changes in a data stream as good as sliding window does. Sliding window contains only most recent elements [6,7]. Algorithms could be mining exact set of frequent itemsets [7] or approximate set of frequent itemsets [6, 9]. Approximate mining is much more effective because it doesn't have to track all itemsets (frequent and not frequent) in history (compared to exact frequent itemset mining) and is able to respond well to conceptual drift.

As a method application, we use identified patterns for recommending interesting pages to website users.

---

* Master degree study programme in field: Information Systems
Supervisor: Ondrej Kaššák, Institute of Informatics, Information Systems and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2017, Bratislava, April 27, 2017, pp. 1–4.

2 Method for mining personalized behavioral patterns over a data stream

In this section, we describe method for online mining of behavioral patterns from the user activity within the website. The method combines global patterns, identified from behavior of all website users, with the group patterns determined for dynamically identified groups of similar users. Based on identified behavioral patterns, proposed method is able to recommend to individual users the pages they are probably going to visit in current session, cache these pages in advance etc. We implemented the method into MOA framework [4], which contains implementations of data stream frequent patterns mining algorithm *IncMine* [6] and data stream clustering algorithm *Clustream* [1]. We use these algorithms to quick prove of concepts proposed in this paper. Later on, we plan to experiment with other algorithms and compare them with actual method results.

2.1 User session processing

The process is designed as a framework, where individual components are independent and can be replaced by another implementation (e.g., different clustering or frequent patterns mining algorithm) (Figure 1). Every user session, represented as set of actions, is continuously loaded from the data stream. User actions could vary from webpage visits to product purchases, shopping basket manages (adding or removing items), etc.

Proposed method uses evaluation approach interleaved test-then-train [4] where each individual example can be used to test the model before it is used for training. This way method is tested on the whole dataset. Therefore, at first, user session is used for recommendation. Next, the session is used for update of user model *u*. Actions from current session are added to queue in user model and *u.nsc* (new sessions count) counter is incremented. If *u.nsc* is greater than input parameter *tcu* (threshold number of changes in user model) then *u.nsc* is nulled and microclusters (compact statistical representations of input data [1]) are updated with instance generated from *u.aq* (user model's actions queue) and *u.muc* (microclusters updates counter) is incremented by 1. If number of updates in microclusters (*muc*) is greater than given threshold *tcm* (threshold number of changes in microclusters) then *muc* is nulled and macroclustering (clustering of microclusters [1]) is performed.

With macroclustering *lmid* (last macroclustering id) is incremented. And every old user model *u*, meeting condition that $lmid - u.lmid > tcdiff$, (where *tcdiff* parameter is threshold of clustering identifiers difference) is deleted from memory. Next if user model *u* has *u.lmid* attribute value other than current global *lmid* then *u.gid* is assigned identifier of group he belongs to according to last macroclustering performed. Lastly, user session becomes an input for algorithm mining frequent patterns (both global and group).

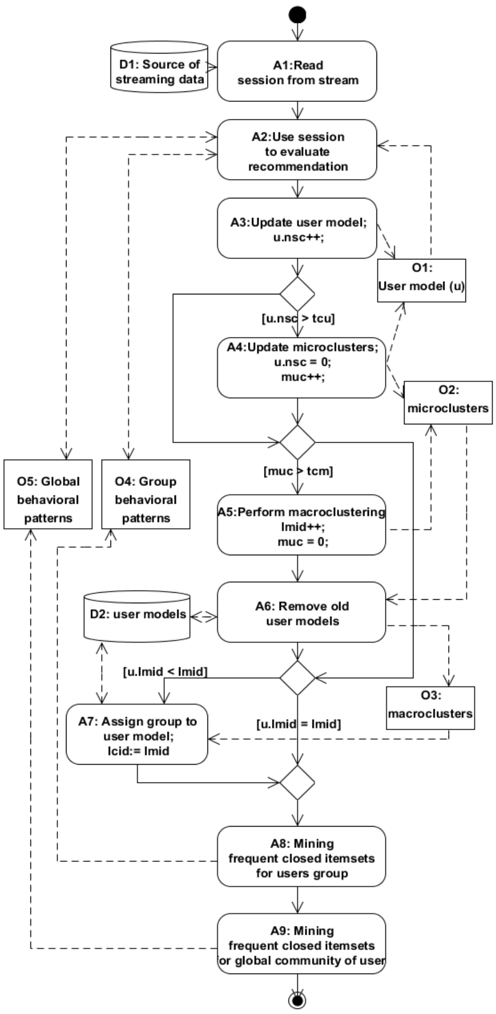

Figure 1: Activity diagram displaying processing of user sessions within method. Diagram elements are tagged with prefixes describing their types: actions as A, data objects as O, data sources as D.

2.2 Application of behavioral patterns

As we mentioned before, behavioral patterns can be applied in wide scale of tasks. In this paper we used them to recommend pages to users based on their previous behavior in the actual session. Let the session be represented as vector of user actions $S = < a_1, a_2, \ldots, a_n >$. Found behavioral patterns are $P = \{P_1, P_2, \ldots P_m\}$. Each pattern P_i is represented as set of actions $P_i = \{a_1, a_2, \ldots, a_l\}$. Let first *k* actions of *S* be dedicated as evaluation window $W_e = < a_1, a_2, \ldots, a_k >$ and all other actions from *S* as testing part $T_e = < a_{k+1}, \ldots, a_n >$. Let *r* be number of recommended items. We use following strategy to choose behavioral patterns according to actual evaluation window W_e:

1. For each P_i in *P* (containing global patterns and patterns from group user belongs to) approximate support value normalized to *<0,1>* interval is computed. Let's mark it as $support(P_i)$. Size of intersection of P_i with W_e is determined with *LCS* algorithm (least common subset). It is also normalized to *<0,1>* interval. Let's mark it as $lcs(P_i, W_e)$.
2. All patterns P_i (global and group) are sorted. First by size of intersection with W_e descending. Second by descending support value.
3. Let *M* be map of items and their *"votes"*. By iterating over all patterns votes values of items are updated. Votes value of item *i* that is contained in pattern P_i and not in W_e is incremented by $support(P_i) * lcs(P_i, W_e)$.
4. Finally, *M* is sorted descending by votes values and best *r* items are picked to be recommended to user.

3 Evaluation

We evaluate proposed method indirectly by recommendation of items for user to visit within the actual session. We compare results of 3 different methods for behavioral patterns identification. First method generates recommendations using global patterns only (*GL*), second using patterns specific for groups of users with similar behavior (*GR*) while third, proposed method, combines previous two into new hybrid method (*GG*) as we proposed in last section.

Recommendation results are compared based on *precision* metric. The recommendation is generated and evaluated for every session *S* from used datasets. The session is divided into training window *A* and testing window *B*. Let *R* be a set of recommended items. We compute *precision* as:

$$precision = \frac{|R \cap B|}{|R|} \quad (1)$$

Processing speed results are compared based on speed metric we defined as average number of sessions processed in a second. It is computed as:

$$speed = \frac{num\ of\ processed\ sessions}{processing\ time\ [s]} \quad (2)$$

We evaluated proposed method on two datasets from domains with different characteristics. First used dataset comes from e-learning system ALEF (Adaptive Learning Framework) [3] (24k sessions, 870 users, 2072 pages - learning objects), second dataset belongs to news portal (NP) (334k sessions, 199k users, 85 categories of pages). Preprocessing of both datasets consisted mainly of users' sessions identification and omitting of too short sessions (1 action long).

Considering number of input parameters our method takes (12), we used grid search approach to find most promising configurations maximizing recommendation precision. To reduce space of possible parameter values combinations, we tuned method parts independently based on their category (clustering, pattern mining, recommendation, general).

Finally, according to performed search, for each dataset, we chose the best configuration maximizing precision metric and processing speed. For these best settings, we compared proposed method (*GG*) to the baseline methods using global (*GL*) and group only patterns (*GR*) (Table 1).

Table 1. Differences in precision of methods for best configuration in ALEF and NP. BEST marks best theoretical combination of patterns, when the better result of GL and GR is chosen for every session.

	P@1	P@2	P@3	P@4	P@5	P@10
			ALEF			
BEST	55.14%	57.02%	58.40%	59.06%	59.88%	63.91%
GG	50.71%	51.26%	51.73%	51.96%	52.41%	52.62%
GL	49.27%	49.85%	50.44%	50.73%	51.06%	51.10%
GR	32.41%	32.93%	33.21%	33.24%	33.57%	34.09%
GG - GL	1.44%	1.41%	1.29%	1.23%	1.35%	1.52%
GG -GR	18.30%	18.33%	18.52%	18.72%	18.84%	18.53%
			NEWSPAPERS PORTAL			
BEST	66.79%	55.92%	59.33%	60.65%	61.00%	54.65%
GG	64.88%	53.68%	55.68%	55.66%	54.91%	51.23%
GL	63.69%	52.62%	54.58%	54.19%	52.76%	49.03%
GR	13.06%	10.86%	15.43%	19.88%	23.07%	11.89%
GG - GL	1.19%	1.06%	1.10%	1.47%	2.15%	2.20%
GG - GR	51.82%	42.82%	40.25%	35.78%	31.84%	39.34%

We performed statistical t-test to compare precisions of baseline methods. For both datasets *GG* reached significant increase of precision ($p < 0.0001$) when compared to *GL* (1.7% in ALEF, 0.9% in NP) and *GR* (21.4% in ALEF, 41.4% in NP).

To be able to compare results of combined *GG* method not only to its parts (*GL*, *GR*), we present also the best theoretical combination of both individual methods (Table 1). In this case, the better result of *GL* and *GR* is chosen for every session. Based on this information, it is possible to evaluate the quality of used approaches combination. In addition to overall precision, we observed recommendation precision inside each user group during the stream processing (Figure 2).

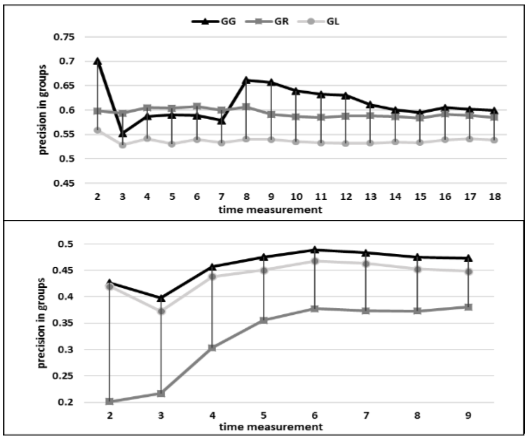

Figure 2: Average precision within groups for NP (upper) and ALEF datasets (lower). Precision is computed as average of precisions inside each group.

In NP dataset, *GR* method outperforms *GL* within groups thanks to high amount of users forming high quality groups and therefore quality patterns. In ALEF dataset, *GL* method outperforms *GR* because there are less users performing more specific sessions and thus it is unable to create such quality patterns. However, proposed method (*GG*) outperformed *GR* and *GL* in both datasets within groups.

We evaluated difference between speed of *GG* and *GL* (Figure 3). We observed that additional computational cost of GG compared to GL, caused by clustering and searching for group patterns, is constant and thus maintainable in production. Also that, abstraction of items (used in NP) from which are identified behavioral patterns results in significantly faster computation (GG is faster with NP than with ALEF).

4 Conclusion

We propose a method, which is able to perform multiple tasks over data stream: segmenting users to groups dynamically, searching for group and global behavioral patterns and applying these patterns in recommendation task. The method was evaluated on e-learning and news data, which have highly different characteristics (total size, number of users, different average length of session). *GG* reached significant overall increase of precision when compared to *GL* and *GR* with both datasets. Next, we plan to evaluate proposed method with multiple different algorithms for mining frequent itemsets and clustering and compare results. We will also try to parallelize individual tasks (clustering, recommendation, global patterns mining) to make method even faster.

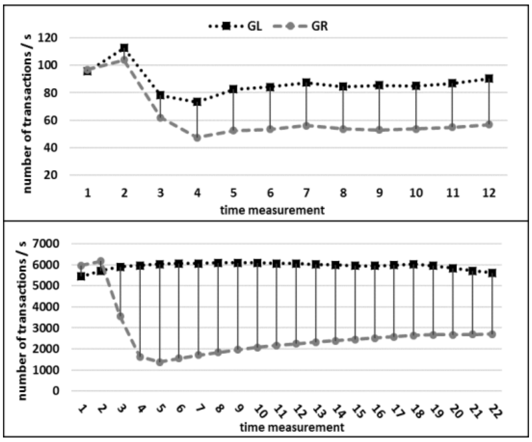

Figure 3: Average processing speed of ALEF (upper) and NP datasets (lower).

References

[1] Aggarwal, Charu C., et al. "A framework for clustering evolving data streams." Proceedings of the 29th international conference on Very large data bases-Volume 29. VLDB Endowment, 2003

[2] Agrawal, Rakesh, Tomasz Imieliński, and Arun Swami. "Mining association rules between sets of items in large databases." Acm sigmod record. Vol. 22. No. 2. ACM, 1993

[3] Bieliková, Mária, et al. "ALEF: from application to platform for adaptive collaborative learning." Recommender Systems for Technology Enhanced Learning. Springer New York, 2014. 195-225.

[4] Bifet, Albert, et al. "Moa: Massive online analysis." Journal of Machine Learning Research 11.May (2010): 1601-1604

[5] Chen, Yixin, and Li Tu. "Density-based clustering for real-time stream data." Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2007.

[6] Cheng, James, Yiping Ke, and Wilfred Ng. "Maintaining frequent closed itemsets over a sliding window." Journal of Intelligent Information Systems 31.3 (2008): 191-215

[7] Chi, Yun, et al. "Moment: Maintaining closed frequent itemsets over a stream sliding window." Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on. IEEE, 2004

[8] Kranen, Philipp, et al. "The ClusTree: indexing micro-clusters for anytime stream mining." Knowledge and information systems 29.2 (2011): 249-272

[9] Li, Hua-Fu, Suh-Yin Lee, and Man-Kwan Shan. "An efficient algorithm for mining frequent itemsets over the entire history of data streams." Proc. of First International Workshop on Knowledge Discovery in Data Streams. Vol. 39. 2004.