Web Usage Mining for Automatic Link Generation

Olatz Arbelaitz, Ibai Gurrutxaga, Aizea Lojo, Javier Muguerza,
Jesús M. Pérez and Iñigo Perona

Dept. of Computer Architecture and Technology, University of the Basque Country UPV-EHU,
M. Lardizabal, 1, 20018 Donostia, Spain

Abstract. During the last decades, the information in the web has increased drastically but larger quantities of data do not provide perse added value for web visitors; there is a need for more efficient access to the required information and adaptation to user preferences or needs. The use of machine learning techniques to build user profiles allows to take into account users' real preferences. We present in this work a preliminary system, based on the collaborative filtering approach, to identify and generate interesting links for the users while they are navigating. The system uses only web navigation logs stored in any web server (according to the Common Log Format) and extracts information from them combining unsupervised and supervised classification techniques and frequent pattern mining techniques. It also includes a generalization procedure in the data preprocessing phase and in this work we analyze its effect on the final performance of the whole system. We also analyze the effect of the cold start (0 day problem) in the proposed system. The experiments show that the proposed generalization option improves the results of the designed system, which performs efficiently w.r.t. a web-accessible database and is even able to deal with the cold start problem.

1 Introduction

During the last decades, the information on the web has increased drastically and this often makes the amount of information intractable for users. As a consequence, the need for web sites to be useful and efficient for users has become specially important; there is a need for faster access to the required information and adaptation to user preferences or needs. That is, Web Personalization becomes essential. Web Personalization [20] can be defined as the dynamic adaptation of the presentation, the navigation schema and the contents of the Web, based on the preferences, abilities or requirements of the user. Nowadays, as Brusilovsky et al. describe in [2], many research projects focus on this area, mostly in the context of e-Commerce [2] and e-learning [9].

Within this context, the current paper presents the design of a complete and generic system that can adapt the web pages to new users' navigation preferences proposing automatically generated links that they will probably be using in a short future.

Our research is contextualized in Web Usage Mining [17]: the application of machine learning techniques to the web usage data. This process requires a data acquisition and preprocessing phase which is not straightforward because it requires different steps such as fusion of data from multiple log files, cleaning, user identification, session

Arbelaitz O., Gurrutxaga I., Lojo A., Muguerza J., M. Pérez J. and Perona I..
Web Usage Mining for Automatic Link Generation.
DOI: 10.5220/0004090400710080
In *Proceedings of the 10th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems and 1st International Workshop on Web Intelligence* (WEBI-2012), pages 71-80
ISBN: 978-989-8565-14-3

identification, path completion processes, etc. Machine learning techniques are mainly applied in the pattern discovery and analysis phase to find sets of web users with common web-related characteristics and the corresponding patterns or user profiles. And finally, the patterns discovered during the previous steps are used in the exploitation phase to adapt the system and make the navigation more comfortable for new users.

The most widely explored application of web access patterns' prediction in the web research community has been web page prefetching [3, 1, 16]. Common characteristics of many of the systems are the use of clustering and/or Markov models to predict the next link to be accessed, but, the URL access order can also be taken into account using sequence alignment methods [4] for the clustering process or sequential pattern mining algorithms [22] when generating user profiles.

We built a system based on the *collaborative filtering approach* that takes as input the minimum information stored in a web server (web server log files stored in Common Log Format –CLF [5]) and combines unsupervised and supervised machine learning techniques and frequent pattern mining techniques to build user profiles and propose links drawn from those profiles to new users. That is, the profiles are built without any effort from the user. We specifically analyze in this work how a generalization process applied to the URLs influences the performance of the system. To evaluate our system we performed experiments in a web-accessible database composed of server log information captured in the NASA [18, 19], with 30K examples, but the system could be integrated in any web environment and applied to larger databases.

We developed the described system and performed experiments to try to answer the following research question: is it possible to automatically generate and propose to users links that they will be using in the future? Are the machine learning techniques we use and their combination adequate for our aim? Is it always interesting to work with generalized URLs or is it worth maintaining the specificity of the URLs in some stages? And finally, is the designed system able to deal with the cold start problem?

The paper proceeds describing in Section 2 the intuition of the system from the user point of view. In Section 3 we describe the database we used in the process and Section 4 is devoted to describing the system we designed. The paper continues in Section 5 where we describe some results of the performed experiments. Finally, we summarize in Section 6 the conclusions and further work.

2 Proposed System: Intuition

We have designed a system able to dynamically propose links to the user who is navigating in the web. The system uses navigation logs to automatically generate user profiles in a batch process, and use them later to automatically propose links to new users. In this kind of systems the proposal of a large amount of links to the user would probably distract her/him and wouldn't be very helpful. As a consequence, our system proposes a small amount of useful links so that the user is not confused. This means, in machine learning terms, that high precision values will be preferable to high recall values.

The evaluation of this kind of systems is complex. We are adapting the presentation of the web data so that after a new user started navigating, proposals of links that she/he will likely be using are done. The best validation strategy would be to perform an user study but that was impossible at this point. As a consequence, we measured the efficiency of the system as the percentage of the proposed links over the proposed ones.

The system has been evaluated using part of the data (2/3) to build it (training) and the rest (1/3) for testing it. The results showed that we were able to successfully predict links. For example in our context where the median of the length of the user navigation sequences is 8 clicks, once the user started navigating (after he/she has done for example 4 clicks) we propose his/her in average 4 new links. More or less one of them will not appear in the user navigation sequence, in average 1.5 of them will belong to the set of clicks already performed by the user, and, the rest 1.5 will be used by the user in the future.

The system would make the user navigation more efficient since it would be proposing the links that he/she will probably use and he/she could faster and more comfortably reach his/her objectives.

3 Database

In this work we have used a database from *The Internet Traffic Archive* [11] concretely NASA-HTTP (National Aeronautics and Space Administration) database [18, 19]. The data contained in this database belongs to web server logs of user requests. The server was located at NASA Kennedy Space Center in Florida and logs were collected during two periods of time. The first set of logs was collected from 00:00:00 July 1, 1995 until 23:59:59 July 31, 1995, a total of 31 days. The second log was collected from 00:00:00 August 1, 1995 until 23:59:59 August 31, 1995, a total of other 31 days. The complete database contains 3,461,612 requests. The contained information is similar to the standardized text file format, i.e. Common Log Format [5] which is the minimum information saved on a web server. Therefore, the system proposed in this work will use the minimal possible amount of information and, as a consequence, it will be applicable to the information collected in any web server.

4 Proposed System: Description

The work presented in this paper is a Web Usage Mining [21] application and as every web usage mining process it can be divided into three main steps: data acquisition and preprocessing [6], pattern discovery and analysis, and, exploitation. The data acquisition phase has not been part of our work. We have designed the system starting from the data preprocessing step up to the exploitation phase.

4.1 Data Preprocessing

We preprocessed the log files filtering out erroneous requests, image requests, etc. to take into account for experimentation only the requests related to user clicks. We performed the user identification based on IP addresses and as an heuristic to identify sessions within a users' activity, we fixed the expire time of each session to 30 minutes of inactivity [14]. Among the obtained sessions, we selected the most relevant ones, i.e., the ones with higher level of activity (6 or more clicks). After applying the whole data pre-processing stages to NASA-HTTP database, the size of the database was reduced to 346,715 HTML requests and 31,778 sessions composed of at least 6 clicks where a total of 1,591 different URLs are visited.

We represented the information corresponding to each of the sessions as a sequence of clicks preformed in different URLs. Other data representation, where other kinds of characteristics such as navigation times, number of clicks per session, length of the longest path form the home page, average times per link, etc. of the users are extracted, could have been used but, in this case, we focus on the navigation patterns: on the visited URLs and the order of these visits.

4.2 Generalization of the Structure

Our aim in this work is to model users based on the identification of general navigation patterns. In this context, having too specific paths in the used data (it is very probable that navigation paths of different users, or the same user in different moments, won't be exactly the same) will make complicated to draw conclusions from the output of machine learning algorithms. As an example, NASA database has 1,591 different URLs and each one is accessed, in average, in 41.3 different sessions out of the 31,778 appearing in the database (we removed the outliers, the ones out of 90% percentile). This means that it will be difficult to find similar click sequences in different sessions. For that reason, we added a generalization procedure to the URL representation whose aim is to represent them a higher level of abstraction. Depending on the parameters of the procedure the average amount of different sessions a URL appears is increased up to a 60%, making it easier to find common patterns.

The first approach to the generalization step consists on erasing a fraction of the URL segments (parts separated by '/' appearing in the URL) from the right end of the path to diminish the URL's specificity. For each one of the visited URLs, we obtained the length of the generalized URL based on next expression:

$$\max \{MinNSegment, (1-\alpha) * NSegments\} \quad (1)$$

Where $NSegments$ represents the number of segments appearing in the URL and α and $MinNSegment$ are parameters that can vary depending on the structure of the site. $MinNSegment$ represents the minimum number of segments, starting from the root, an URL can have after the generalization step, whereas, α represents the fraction of the URL that will be erased in the generalized version. This generalization process will allow us to work with a more general structure of the site avoiding the confusion that too specific zones could generate. A study of the generalized structure of NASA database, instantiating $MinNSegment = 3$, showed that the URL structure saturates with values greater than 0.5 for α, that is, the structure has 367 URLS when $\alpha = 0.5$ and it is only reduced to 330 when $\alpha = 0.9$. As a consequence, we used 3 values of α to evaluate the system: 0 (not generalization), 0.25 and 0.5.

4.3 Pattern Discovery and Analysis

This is the stage in charge of modeling users and producing user profiles taking as input the user click sequences. In this context unsupervised machine learning techniques have shown to be adequate to discover user profiles [20].

We used the *PAM (Partitioning Around Medoids)* [15] clustering algorithm in order to group into the same cluster users that show similar navigation patterns according to click sequences and a sequence Alignment Method (Edit Distance [10, 4]) as a metric to compare sequences. *PAM* requires as input an estimate on the maximum number of clusters to provide as output (*k* parameter). Since we didn't have prior knowledge of the structure of the data, in NASA database, in this preliminary approach we fixed *k* to 100. As a consequence, we might be somehow obliging the clustering algorithm to assign patterns to a cluster even if their distance to the rest of the patterns in the same cluster is big. We have avoided the noise this could produce by removing in each cluster the patterns with distance 1 (maximum distance) to every other pattern.

The outcome of the clustering process is a set of groups of user sessions that show similar behavior. But we intend to model those users or to discover the associated navigation patterns or profiles for each one of the discovered groups, that is, to find the common click sequences appearing across the sessions in a cluster. In this step we used SPADE (Sequential PAttern Discovery using Equivalence classes) [22], an efficient algorithm for mining frequent sequences, to extract the most common click sequences of the cluster. The application of SPADE provides for each cluster, a set of URLs that are likely to be visited for the sessions belonging to it. The number of the proposed URLs depends on parameters related to SPADE algorithm such as minimum support or minimum number of sessions in the cluster containing that URL, and maximum allowed number of sequences per cluster. We performed several experiments, and, finally, a fixed value for the minimum support, 0.5, showed to be a good option.

Although for the clustering and exploitation stages we experimented with generalized and not generalized URLs, if we would use generalized URLs at this stage, the system wouldn't be able to propose real URLs to the user, and, as a consequence, the it would require an extra stage in order to be useful for the final user. Thereby, we applied the SPADE algorithm using the original URLs appearing in the user click sequence.

The reader could probably easily conclude from the previous paragraphs that the pattern discovery and analysis phase is a batch process and its outcome will be used in the exploitation phase in real time. As a consequence, increasing the size of the input database wouldn't increase the cost of the process in the exploitation phase.

4.4 Exploitation

This is the part that needs to be done in real time. Up to now, we have identified groups of users with similar navigation patterns and the URLs that are most likely to be visited, or most common paths, for each of the groups. At this point we need to use that information to automatically propose links to new users navigating in the web. As a first approach, we propose the use of k-NN (1-NN) [7] learning approach to calculate the distance (average linkage distance based on Edit distance [10]) of the click sequence of the new users to the clusters generated in the previous phase. This distance can be calculated at any stage of the navigation process, that is, from the first click of the new user to more advanced navigation points. Our hypothesis is that the navigation pattern of that user will be similar to the user profile of its nearest cluster. As a consequence the system will propose to the new user the outcome of SPADE, the set of links that models the users in that cluster.

5 Experiments: Results and Analysis

In order to evaluate the performance of the whole process, we applied the Hold-out method dividing the NASA database into two parts. One for training or generating the model, that is, for generating the clusters and extracting profiles. And another one for testing or using it in exploitation, that is, for evaluating to what extent the links generated by the system would come along with the navigation performed in those sessions. To simulate a real situation we based the division of the database on temporal criteria: we used the oldest examples (66% of the database, 21,185 user sessions) for training and the latest ones (33%, 10,595 user sessions), for testing. In the training database, the total number of requests is 235,155 (1,419 different URLs accessed) and the median of the number of clicks per session 8. The test database seems to have similar characteristics. Although it is smaller, the total number of requests is 111,605, the median of the number of clicks per session is still 8.

We applied to the training data the combination of *PAM* with different values for α generalization parameter so that the sessions with similar navigation characteristics are clustered into the same group. Then we generated with SPADE the navigation profile of each group of users or set of links that they will probably use. These profiles will be compared to the click sequence of the users in the test examples to evaluate the performance of the system.

To validate the system, we used the test examples as described in the exploitation phase and then we compared the automatically generated links with the real click sequences of the users. We performed this comparison taking into account the 0-day problem. That is, although in the used database we have the complete navigation sequence of the test sessions, in real executions, when a user starts navigating, only her first few clicks will be available to be used for deciding the corresponding profile and proposing new links according to it. We have simulated the real situation using 10% (just one click out of 8), 25% and 50% of the user navigation sequence in the test examples to select the nearest cluster or profile. We also performed the experiments for the complete test sequences (100%) as an upper bound.

We computed statistics based on results for each one of the new users. We compared the number of proposed links that are really used in the test examples (hits) and the number of proposals that are not used (misses) and calculated precision (*precision*). Note that this could be seen as a lower bound because, although not appearing in the user navigation sequence, the proposed links could be useful for her/him. Unluckily their usefulness could only be evaluated in a experiment using the user feedback.

An ideal system would maintain precision and recall as high as possible. But we focus on precision because, as we mentioned before, in this kind of systems, proposing a large amount of links would distract the user and proposing a small amount of links, we can not expect to guess the whole navigation path of new users.

We calculated the precision (Pr) taking into account only the clicks in the test sequence that have not been used to select the nearest profile; that is, taking into account the remaining 90%, 75% or 50%. Moreover, we calculated an upper bound for the precision (PrUp) taking into account the whole test sequence.

As a summary, we evaluated the system's performance for the next configurations:

- PAM clustering algorithm applied to click sequences with different generalization options for the URLs: 0 (CL000), 0.25 (CL025), 0.5 (CL050).
- 1-NN with different generalization options for the URLs: 0 (NN000), 0.25 (NN025), 0.5 (NN050).
- Prediction at different stages of the navigation: 10%, 25% and 50% and 100% (as an upper bound).
- Real precision (Pr) and upper bound for precision (PrUp).

Table 1 and Figure 1 summarize the mentioned results. In Table 1 precision values are shown for the different percentages of the test sequences we used to identify the profile of the new user (columns) and the different configurations we evaluated (rows). In Figure 1 we can observe the trends of the precision (Pr and PrUp) for the different configurations we evaluated, while the percentages of the test sequences used to identify the profile of the new user increase (X Axe).

Table 1. Summary of precision values.

Option	**10%**	**25%**	**50%**	**100%**
CL000-NN000-Pr	39.7	42.0	35.1	
CL025-NN000-Pr	40.0	42.2	35.4	
CL025-NN025-Pr	39.3	42.4	35.6	
CL050-NN000-Pr	47.2	46.7	39.1	
CL050-NN050-Pr	**56.2**	**51.7**	**43.8**	
CL000-NN000-PrUp	52.9	63.0	69.6	73.5
CL025-NN000-PrUp	53.4	64.3	70.0	73.8
CL025-NN025-PrUp	52.9	64.4	70.1	73.7
CL050-NN000-PrUp	60.9	68.5	72.3	73.5
CL050-NN050-PrUp	**69.5**	**72.2**	**75.5**	**76.1**

The first conclusion we can draw form the results is that even if the values of the measured parameters vary depending on the selected option, all configurations are able to predict a certain percentage of the links a new user will be visiting. That is, we designed a general system that, without any effort from the users, is able to automatically generate link proposals that will be helpful to make more comfortable and efficient the navigation of new users in the web. And as a consequence, we can claim that the machine learning techniques we use and their combination seem to be adequate.

Furthermore, from the comparison of the achieved values we can conclude that, as expected, independently of the approach used to calculate precision, generalization helps in the first step of the process and moreover the generated user profiles seem to be

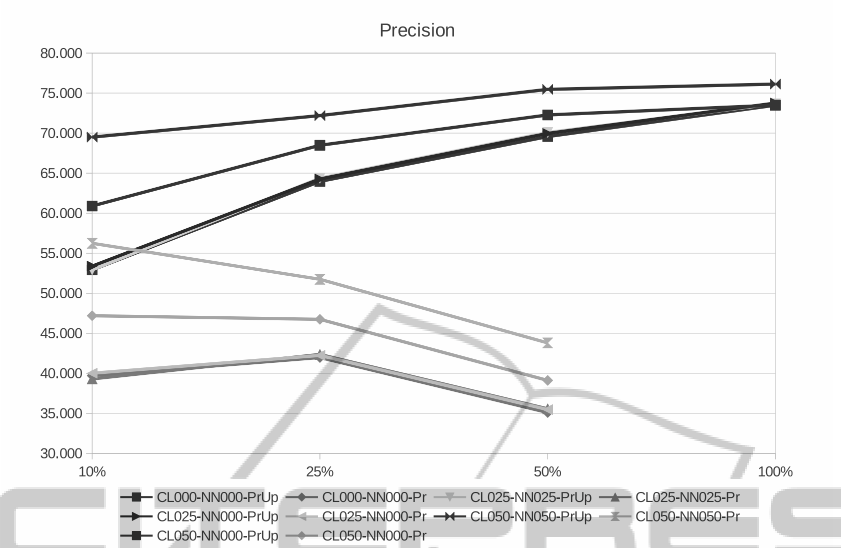

Fig. 1. Precision values achieved with different configurations of the system.

more adequate for bigger values of α parameter (0.5). Furthermore, although its effect is smaller, the use of generalization also seems to be important in exploitation, when the most similar profile to a new user is being selected. It is again the $\alpha = 0.5$ option the one obtaining the best results. Independently of the percentages of the user navigation known in exploitation (10%, 25%, 50% or 100%) the best precision values are the ones obtained for CL050-NN050-Pr and CL050-NN050-PrUp.

Those are the options obtaining the highest precision values and their average difference with the rest of the options was in a range of about 12%. We calculated recall values for the different options and the differences were smaller than for precision; the difference of the values obtained for CL050-NN050-Pr and CL050-NN050-PrUp were only around 5.3% smaller if compared to the best recall value obtained.

It can also be observed that precision values for Pr option are more than half of the precision values achieved for PrUp option. From this data we can conclude that more than half of the hits belong to URLs our system has proposed and come along with the clicks the user will be doing in the future.

If we center the analysis in the 0-day problem, we realize that as it could be expected, the trends of the curves change for the Pr and PrUp. In the case of PrUp the quality of results logically increases when longer has the user been navigating. In both cases the values are still acceptable at very early stages of the navigation. When just 10% (one click in average) of the user navigation sequence is known, good precision values (Pr = 56.2 and PrUp = 69,5) are obtained. That is, the system is able to deal with the 0-day problem. However, in the case of Pr the quality of results decreases when longer has the user been navigating. This could also be expected because when longer this sequence is, the amount of URLs the user did not select yet becomes smaller.

6 Conclusions and Further Work

We designed a system that, without disturbing the users, based just on server log information and machine learning techniques, identifies different groups of users, builds the corresponding profiles, and automatically generates useful link proposals for new users so that their navigation becomes more efficient. This work has been done for NASA-HTTP database [18, 19], but could be extended to any other environment since it has been built using the minimum information stored in any web server (in Common Log Format). We preprocessed the data to identify users and sessions on the one hand, and prepared it so that it could be used with machine learning algorithms. We then proposed a generalization step and applied *PAM* clustering algorithm to the training data to discover groups of users with similar interests or navigation patterns. Once the groups of users were identified, we used SPADE algorithm to discover the profiles associated to each of the clusters, or, the links that will be proposed to new users. We evaluated different configurations of the system and how it deals with the 0-day problem. To perform this evaluation, we used a Hold-out strategy: we divided the database into two parts one for training and the other one for testing.

We evaluated to what extent the system is able to predict the links a new user will be using. The validation results showed that the discovered patterns made sense and that they could be used to ease the navigation of future users by proposing or underlying links that they will probably use. This happens even if the prediction is made at very early stages of their navigation (10%). Results showed that the proposed generalization is appropriate for both, the clustering stage and finding the nearest profile of a new user during the exploitation phase. Moreover, greater levels of generalization seem to work better (the best results are achieved with CL050-NN050).

So, we could conclude that, using adequate machine learning techniques, we have been able to design a generic system that based only on web server log information and without any effort from the users, is able to propose adaptations to make easier and more efficient the navigation of new users. Since at this point we haven't used any domain specific information, this system would be useful for any web site collecting server log information.

This is an ongoing work that needs to be improved in many senses. First, it should be applied to more recent data. Regarding to the evaluation, on the one hand, cross-validation should be used to assess precision more accurately, and, on the other hand, ideally an user study should be performed in a real web environment where links can be proposed to new users and the effect of those links in their navigation can be assessed. Moreover, regarding to the system architecture, on the one hand, a wider analysis of the parameters is required, and, on the other hand,the stage of the selection of the nearest profile for a new user needs more refinement. Finally, further improvements could be done using web structure information and content information of the selected web page for improving the results of the system.

Acknowledgements

This work was funded by the University of the Basque Country, general funding for research groups, ALDAPA (GIU10/02); by the Science and Education Department of the Spanish Government, ModelAccess (TIN2010-15549 project); by the Diputación Foral de Gipuzkoa, Zer4You (DG10/5); and by the Basque Government's SAIOTEK program, Datacc (S-PE11UN097).

References

1. Anitha, A. A new web usage mining approach for next page access prediction. International Journal of Computer Applications, 8(11):7–10, 2010.
2. Brusilovsky P., Kobsa A. and Nejdl W. The Adaptive Web: Methods and Strategies of Web Personalization LNCS 4321, Springer, 2007.
3. Chen X., Zhang X. A popularity-based prediction model for web prefetching. Computer, 36(3):63–70, 2003.
4. Chordia B.S., Adhiya K.P. Grouping Web Access Sequences Using Sequence Alignment Method. In Indian Journal of Computer Science and Engineering (IJCSE), Vol. 2(3), (2011)
5. The Common Log Format http://www.w3.org/Daemon/User/Config/Logging.html #common-logfile-format
6. Cooley R., Mobasher B. and Srivastava J. Data preparation for mining world wide web browsing patterns. Knowledge and Information Systems, 1(1), 1999.
7. Dasarathy B.V. Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques IEEE Computer Society Press, Silver Spring, MD, 1991.
8. Desikan P., Srivastava J., Kumar V. and Tan P.N. Hyperlink Analysis - Techniques and Applications. Army High Performance Computing Center Technical Report, 2002.
9. García E., Romero C., Ventura S. and De Castro C. An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. User Modeling User and Adapted Interaction, 19(1-2), pages 99–132, 2009.
10. Gusfield D. Algorithms on strings, trees, and sequences. Cambridge University Press, 1997.
11. The Internet Traffic Archive. http://ita.ee.lbl.gov/. ACM SIGCOMM.
12. Jain A.K., Dubes R.C. Algorithms for Clustering Data. Prentice-Hall, Upper Saddle River, NJ, USA, 1988.
13. Kosala R. and Blockeel H. Web Mining Research: A Survey. ACM SIGKDD Explorations Newsletter, 2(1), pages 1–15, 2000.
14. Liu B. Web Data Mining. Exploring Hyperlinks, Contents, and Usage Data. Springer, 2007.
15. Liu L. and Özsu M.T. Encyclopedia of Database Systems. In: PAM (Partitioning Around Medoids). Springer US, 2009.
16. Makkar P., Gulati P. and Sharma A. A novel approach for predicting user behavior for improving web performance. International Journal on Computer Science and Engineering (IJCSE), 2(4):1233–1236, 2010.
17. Mobasher B. Web Usage Mining. In: Web Data Mining: Exploring Hyperlinks, Contents and Usage Data. Springer, Berlin, 2006.
18. NASA-HTTP logs. HTTP requests to the NASA Kennedy Space Center WWW server. http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html, in Florida, 1995.
19. National Aeronautics and Space Administration. http://www.nasa.gov/, 2010.
20. Pierrakos D., Paliouras G., Papatheodorou C. and Spyropoulos C.D. Web Usage Mining as a Tool for Personalization: A Survey User Modeling and User Adapted Interaction, 13:311–372, 2003.
21. Srivastava J., Desikan P. and Kumar V. Web Mining - Concepts, Applications & Research Directions. In Foundations and Advances in Data Mining. Springer, Berlin, 2005.
22. Zaki M.J. SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning, 42:31–60, 2001.