Algorithms for Clustering ClickStream Data

Panagiotis Antonellis, Christos Makris, Nikos Tsirakis *

Dept. of Computer Engineering & Informatics,
University of Patras, 26500 Rio-Patras, Greece
E-mails: {adonel,makri,tsirakis}@ceid.upatras.gr
Tel: +302610997529, Fax: +302610969002

Abstract

Clustering is a classic problem in the machine learning and pattern recognition area, however a few complications arise when we try to transfer proposed solutions in the data stream model. Recently there have been proposed new algorithms for the basic clustering problem for massive data sets that produce an approximate solution using efficiently the memory, which is the most critical resource for streaming computation. In this paper, based on these solutions, we present a new model for clustering clickstream data which applies three different phases in the data processing, and is validated through a set of experiments.

Key words: Data Mining, Clustering Algorithms, Data Bases.

1 Introduction

Even if clustering has been widely studied in several disciplines accompanied with many models and algorithms [1,3–6,9,11,12], little has been done in processing web clickstreams and providing real time analytics. Many tools tend to ignore detailed sequence and time information in order to save space and time of processing data but according to [2] these ignored data make a substantial difference in cluster quality for certain web sites. Although the above research focuses in user behaviour and how this can be structured, it is still difficult to

* Corresponding author: *Nikos Tsirakis*, *E-mail: tsirakis@ceid.upatras.gr*

process clickstreams by using single access methods, to process them online in order to provide direct results to the analysts and finally to adapt to the changes of user behaviour and handle these random alterations.

We propose a model for clustering clickstreams, which extends the design proposed in [1], by using a framework consisting of three different phases in the data processing. In [1], the authors were interested in clustering evolving data streams via a proposed framework. They introduce a different philosophy for data stream clustering which is guided by application-centered requirements. According to their design, the framework is divided into an online micro-clustering component which periodically stores detailed summary statistics and an offline component which uses a pyramidal time frame. We adopt this general concept and we extend it appropriately by using a three phase process instead of a two phase in [1]. Our approach can support the clustering tasks efficiently by processing data in an online way providing both short-term and long-term results about user's clickstreams.

2 Clustering ClickStream Data

Our proposed model extends the work in [1] and specializes in the use of access log data in the form of clickstreams. In our case the online component stores the summarized data in the temporary memory where the web log data is merged instead of using a storage system. In addition this component does not create micro-clusters but creates an matrix with fixed size that summarizes the web log data. The offline component can be comprised by many different clustering algorithms that work in parallel and independently in order to provide a variety of groups of data clusters depending on the current given attributes. For our experimental study, we apply only clustering and after a pre-processing of the resulted clusters we apply a meta-clustering procedure. In other words we don't propose only a three-phased approach but an integrated system that can be extended even to a web service that could provide combinations of clustering methods in order to give as an output powerful insights in the web click streams. The novelty in our work is that we handle the problem in an online way by the first phase of the framework process and we

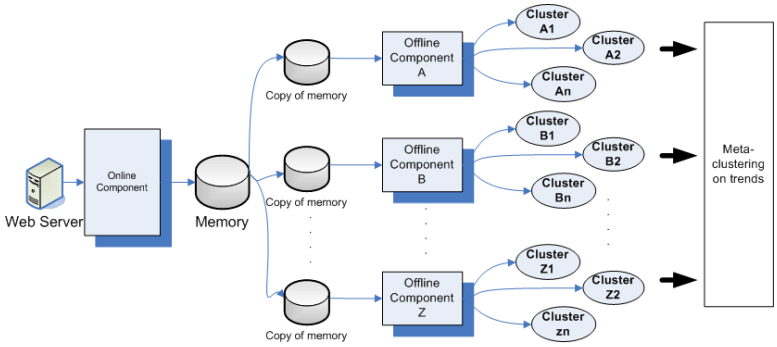

Fig. 1. Model architecture.

provide different views of users in the following two phases, which is valuable for many web community systems and web analytics tools.

3 Framework

3.1 Architecture

The main idea is to divide the clustering process into three separate phases (Figure 1): i) an on-line component that automatically stores continuously data in a compressed way; ii) a set of offline components that use these compressed data in order to produce clusters; iii) a meta-clustering procedure that detects trends in the produced clusters in order to provide a long-term clustering. The proposed model supports not only clustering but various data mining techniques, in the offline component, depending on the current needs of the problem. The input stream that is being handled by our model is the web-clickstream of a web server that hosts websites.

3.2 Algorithms

3.2.1 Data Summary

In this component we maintain compressed data of server requests in a continuous way. These summary statistics provide sufficient information about users that access the server. The advantage of this part is that the process is running without breaks and continuously updates efficiently the summary statistics in the memory.

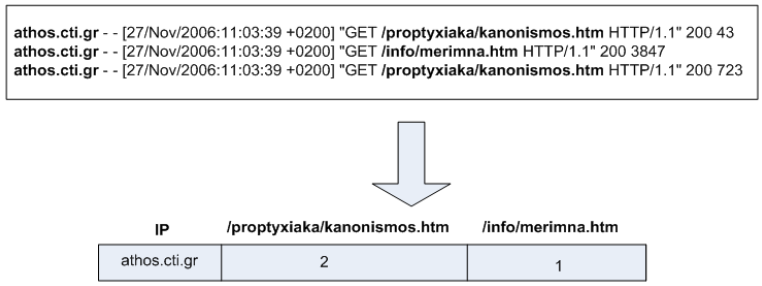

Fig. 2. Summary Matrix.

In our experimental model we use a summary visitor data matrix. Figure 2 shows part of it. In the columns we store the number of times that a user visit a specific page and in the rows the *IPs* that correspond to the current user. We have chosen these two specific attributes of log entries as they maintain the most useful information about a user and at the same time facilitate the merge of data. In this way we save in memory an array that only holds counters about user accesses in the predefined urls. It is assumed that the array is maintained at any moment by the algorithm.

Figure 2 describes how the log records are transformed to data item records in order to apply the clustering process. For every distinct *IP address* in the log file, we create a corresponding data item record. This record summarizes all the distinct web pages that the corresponding user (*IP address*) has visited. For each such web page, the data item record stores the total number of user hits on that page.

3.2.2 Data Clustering

In this component we can apply several clustering algorithms, as separate components, that use the array of the on-line component to create clusters. The main idea here is to create snapshots of the array in the memory and apply the offline components in these snapshots in order to create clusters. The processes of the offline clustering components run periodically with a time period equal to T_i, and in each of their execution they deal with different clickstream snapshots. The length T_i of the time period depends on the runtime of the clustering algorithms that are used and on the frequency with which the clusters are speculated to change. For each time period we store the current results from the on-line component accompanied with the corresponding clusters. The

produced clusters can be stored in a database or in memory depending on how detailed clusters we want to have every time. The clustering algorithm that we have chosen to apply is the well known k-means clustering algorithm [8]. While it does not work well with categorical attributes, it has the good geometric and statistical sense for numerical attributes. K-means takes as input the array of counters and applies clustering to them. Every time instant the k-means algorithm runs, it produces some clustering results and we store these clusters and the current snapshot of the array of data. In this way we have every time a collection of clusters stored in secondary memory, accompanied with the corresponding array of data.

3.2.3 Data Meta-Clustering

The final phase of our framework is the application of a long-term clustering procedure called meta-clustering, in which we try to cluster the web users based on their behaviour during the whole time period under consideration. In the previous phase we have created a number of clusters for different time periods. In this phase, we transform each user, which is identified by a distinct *IP address*, into a string sequence of length equal to the total number of micro-clustering procedures applied previously. The *i-th* position in the string sequence of a user is the cluster's number where the user belonged during the *i-th* microclustering procedure. For example, a user that belonged to *Cluster 1*, *Cluster 3*, *Cluster 3*, *Cluster 2*, *Cluster 6* at the *5* corresponding microclustering procedures, is encoded as *13326*.

Concerning now the processing of the attained user representations, there exist many challenges on how common or not are sequences of data and how we can measure them. Clustering based on sequential structural characteristics can serve as a powerful tool to discriminate sequences belonging to different functional categories. An approach in this problem is the use of the edit distance [7] in order to measure the distance between each pair of sequences. The Levenshtein distance or edit distance between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character. In the meta-clustering procedure the unconstrained edit distance measure is needed to transform the information about a user into a vector as a distance

measure [6]. In other words every user here is being described by a vector with values of the distances that he has with the other users.

4 Experiments

The proposed framework was illustrated in the main server of the Computer Engineering & Informatics Department, University of Patras and we analyzed the data from the web portal of the department (http://www.ceid.upatras.gr) and its sub-domains. We have selected this web portal for three major reasons: we could better validate our results, it is a great example with many users and hits every day and the majority of users were allocated with unique *IP* number. Over the course of the study, which lasted *1* month, *3500* interactions were appended to the web server and *670* users used the web site. We have performed the framework on a P4 at 2.3 GHz with 512 Mb of main memory under Windows XP. The online component run continuously for this time period of *4* weeks (as a service) and every *6* days the clustering procedure was formed. At the end of this period the experiment finalized with the meta-clustering procedure of the sum of resulted clusters. This experimental study was setup using the following steps:

- The requests from the apache web server were collected in a special structure.
- The k-means clustering algorithm was employed for the analysis of user behaviour.
- The collected clusters were pre-processed in order to be the input for a meta-clustering procedure.

As far as concerning data clustering, the k-Means clustering algorithm was employed in this step of research work. The algorithm took as an input the data that have been extracted, the number of clusters k and the centroids. Finding the right number of clusters for a data set is a difficult problem, because even in the same data set, there can be several answers depending on the scale or granularity one is interested in. The value of k was selected after several experiments in order to efficiently execute the algorithm. More specifically, the selection of a large number of clusters may result in one or

Table 1
Micro-Clustering Results

Cluster	**Number of objects**	**Page categories**
1	7	Forums, Main pages
2	26	Forums, Admin pages
3	105	Mailbox, Admin pages
4	34	Mailbox, Main pages
5	16	Forums, Mailbox
6	13	Mailbox

more empty clusters. As far as no empty clusters are created, the criterion of our decision was the granularity of the clustering. We defined as optimal granularity, the one that offers the most detailed representation of the users. Hence, we gradually increased the number of clusters observing the granularity of the clustering while preserving the condition that no empty clusters were created. As far as concerning the centroids of the clusters, we have initialized them using the results from a pre-processing procedure. Some techniques that try to avoid random initialization of centroids can be found in [14].

K-means was used in both micro and macro clustering phases. Concerning the micro-clustering phase the analysis focuses to sub groups and to small number of users. According to [10] micro-mining analysis gives better results for understanding the site's use and users current behavior.

Table 1 presents the microclusters formed in one of the microclustering phases. It is clear from Table 1 that the microclusters group users based on the type of visited pages during the experiment's period. For example, the first cluster groups all the users who have visited the forum pages along with the main pages of the website. On the other hand, the second cluster groups all the users who have visited both the forum pages and the administrator pages. Hence, because of this property of microclusters, the formed microclusters can be utilized in order to provide recommendation pages to the users, based on the microcluster they belong. Moreover, we can use our results as follows:

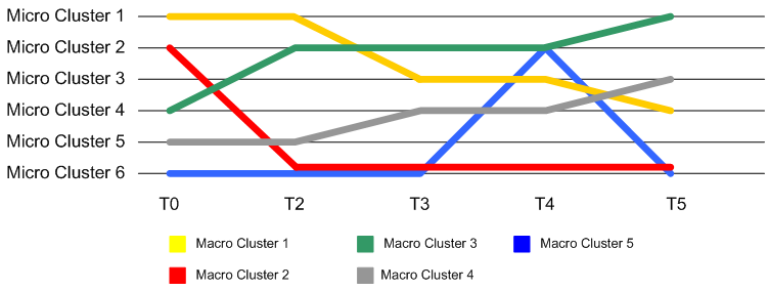

Fig. 3. User's clustering transition.

- We can use the top visited pages of the big clusters in order to cache them for future use.
- We can find out the current user interests and recommend them with pages that other users of the same cluster have visited.

The final step of the proposed model was the meta-clustering procedure where we performed the long-term clustering. We used the data from the *5* previous clustering procedures and after the transformation of user transitions from cluster to cluster we created a sequence for each user of length 5 that corresponds to the total period of our experiment. Then we used the Levenshtein distance between the sum of sequences and we built an array of size *670*670*. In this array every row describes the edit distance between the current user and the others including him. The base assumption for this data set was that significant changes were observed over time between users and clusters, due to user behaviour, increasing the need for further mining over the data. Figure 3 bellow illustrates these changes between clusters for a certain user.

According to [13] it is desirable to have long term analysis in order to make any usability analysis about a web site. Moreover log files and microclusters usually fall short for this type of analysis. On the opposite long-term clustering results can provide groups of users that share common interests or common behavior and thus can be used to improve usability problems in a web site. This leads us to identify typical visiting profiles.

- We have found that clusters were mostly consisting of specific types of users. For example the majority of postgraduate students belonged in the same cluster. The same stand for Professors or Department staff.
- We have clusters with monothematic behaviors, people who visit monthly

a specific area of the website, illustrated by *cluster 3*.

- A second type of behavior is polythematics behavior, illustrated by *cluster 4*. Here visitors exploit all areas of the site.
- Another type of behaviour is visitors who belong between the two previous types but this is also another profiling category of visitors. Example of this is *cluster 5*.

Macro-clustering results give a different view of users than micro-clustering results. Hence we have different views of users with different interpretation and meanings. From our experiments we have collected some useful and remarkable conclusions. Initially according to the data clustering results we have discovered users that exhibit similar characteristics in the evolution of clickstreams (e.g. users that share common change patterns among clusters). In addition, we have verified that the largest cluster (every time) contained users who had visited the same group of pages which were eventually dynamic pages with important information in contrast with the other clusters that contained users who had visited the sum of pages. By combining the initial merged table with the data meta-clustering results we provide recommendations to users in order to visit other unvisited pages according to the moves of users that all belong to the same cluster. The meta-clustering procedure gives an overview of user's behaviour and this helps us recognize the useful clusters that have users who have stable preferences about page content. Finally, if we look into the outliers of macroclusters we find the pages that have to be re-structured by the characteristics of usability. The general conclusion is that the suggested framework shows promise providing practical results and working reasonably well.

5 Conclusions & Future Work

In this paper, we presented a model for clustering clickstream data which is based in the use of three different phases in the data processing, and is validated through a set of experiments. The proposed three-phase process provides more reliable and sophisticated measures and enhances static user information with dynamic information. The model was tested on a real web site and we compared the clustering results from the second and the third component of

the model. As mentioned before, many web mining models have been constructed based on web log data but our approach gives some new directions of data manipulation and cluster evaluation. Furthermore, the illustrated model seems adequate for applying other mining techniques in parallel with the current algorithms. Some other system extensions for augmentative and detailed analysis could be the use of queries in one-dimensional representations and persistent data structures and multidimensional data structures in order to apply queries based on time. Finally, we wish to point out that future works may be directed at studying issues such as data quality and privacy preservation [15].

References

[1] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, Philip S. Yu, "A Framework for Clustering Evolving Data Streams", Proceedings of the 29th VLDB conference, 81-92 (2003)

[2] A. Banerjee and J. Ghosh, "Concept-based clustering of clickstream data", Proceedings of 3rd International Conference on Information Technology, 145-150 (2000)

[3] Robert Cooley, Bamshad Mobasher, Jaideep Srivastava, "Web Mining: Information and Pattern Discovery on the World Wide Web", Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence, 558-567 (1997)

[4] Yongjian Fu, Kanwalpreet Sandhu, Ming-Yi Shih, "A Generalization-Based Approach to Clustering of Web Usage Sessions", International WEBKDD, 21-38 (1999)

[5] Minos N. Garofalakis, Rajeev Rastogi, S. Seshadri, Kyuseok Shim, "Data Mining and the Web: Past, Present and Future", Proceedings of the 2nd International Workshop on Web Information and Data Management, 43-47 (1999)

[6] S. Gunduz and M. Ozsu, "A Web Page Prediction Model Based on Click-Stream Tree Representation of User Behavior", Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 535-540 (2003)

[7] Dan Gusfield, "Algorithms on strings, trees, and sequences: computer science and computational biology", Cambridge University Press (1997)

[8] J. A. Hartigan, M. A. Wong, "Algorithm AS136: a k-means clustering algorithm", Applied Statistics, vol. 28, 100-108 (1979)

[9] Olfa Nasraoui, Cesar Cardona and Carlos Rojas, "Single Pass Mining of Evolving Trends in Web Data with Explicit Retrieval Similarity Measures", Proceedings of International Web Dynamics Workshop, New York, NY, (2004)

[10] Nicholas, D. and Huntington, P., "Micro-Mining and Segmented Log File Analysis: A Method for Enriching the Data Yield from Internet Log Files", Journal of information science, Vol. 29, No. 5, 391-404, (2003)

[11] Vir V. Phoha and S. Sitharama Iyengar and Rajgopal Kannan, "Faster Web Page Allocation with Neural Networks", IEEE Internet Computing 18-26, (2002)

[12] Santosh K. Rangarajan, Vir V. Phoha, Kiran S. Balagani, Rastko Selmic, S. Sitharama Iyengar, "Adaptive Neural Network Clustering of Web Users", IEEE Computer 37(4), 34-40, (2004)

[13] Tec-Ed, Inc, "Assessing web site usability from server log files", Ann Arbor, MI: Tec-Ed, 1999.

[14] M. Vazirgiannis, M. Halkidi, "Data & Web mining", (2006)

[15] V. Verykios and E. Bertino and I. Fovino and L. Provenza and Y. Saygin and Y. Theodoridis, "State-of-the-art in privacy preserving data mining", SIGMOD Record 33(1), 50-57, (2004)