A Model for Clustering Clickstream Data

Christos Makris
Department of Computer Engineering & Informatics
School of Engineering, University of Patras Rio Campus, 26500 Patras, Greece
+302610996968
makri@ceid.upatras.gr

Nikos Tsirakis
Department of Computer Engineering & Informatics
School of Engineering, University of Patras Rio Campus, 26500 Patras, Greece
+302610997529
tsirakis@ceid.upatras.gr

ABSTRACT
The extremely large number of data sets that can be drawn from internet has bootstrapped in a way the data mining techniques from extracting real time results. For this reason clustering and other mining techniques in the data stream model have grasped the interest of Data Mining community [6]. Recently have been proposed many algorithms for the basic clustering problem for massive data sets [7] that produce an approximation using efficiently the memory, which is the most critical resource for streaming computation. Also have been studied some approximation algorithms [8] for the k-median clustering using corsets in R^d with only polynomial dependency on d.

In this paper, we present a new model for clustering click stream data. We adopt the idea of division the clustering process into an online component which periodically stores detailed summary statistics and an offline component which uses only this summary statistics [3]. In this concept we use k-means in order to perform clustering over data streams which is a very efficient and powerful algorithm for this type of problems. For the purposes of our experiments we have modified k-means in the input process. In other words we have built a mechanism that provides k-means with real time data feeds from web server.

The main originality of our model is based on the fact that it uses compressed data of click streams in memory providing on-line clustering with real time results. These results allow further cluster analysis and have security and usability perspectives.

General Terms
Algorithms, Measurement, Performance, Design, Reliability, Experimentation, Security, Theory.

Keywords
Clustering, Web Mining, Click Streams, Security, Usability.

1. INTRODUCTION
The evolution of internet has lead to an explosion of available information. Modern hardware and database technology has made it possible to store all this information in big and distributed databases. However, this rapid digitalization has pointed out an important need for techniques, tools and algorithms to delve and efficiently discover valuable, non-obvious information from large databases and more recently from data streams. Data mining techniques have been proposed and studied to help users better understand and analyze the information. Clustering and other mining techniques have grasped the interest of the Data Mining community.

Clustering is a useful and ubiquitous tool in data analysis [9]. In broad strokes, is the problem of finding a partition of a data set so that, under some definition of "similarity," similar items are in the same part of the partition and different items are in different parts. With the rapid increase in web-traffic and e-commerce, understanding user behavior based on their interaction with a website is becoming more and more important for website owners and clustering in correlation with personalization techniques of this information space has become a necessity [10]. The knowledge obtained by learning the users preferences can help improve web content, find usability issues related to this content and its structure, ensure the security of provided data, analyze the different groups of users that can be derived from the web access logs and extract patterns, profiles and trends.

Data clustering problem is not so contemporary in computer science society. Recently a new class of data came up to augment the difficulties of data clustering. Data streams are large volumes of data arriving continuously. This renders most traditional algorithms too inefficient.

Our results. We propose a model for clustering click streams using a framework consisting of an online component and an offline component. In [3], the authors were interested in clustering evolving Data Streams via a framework. According to their design the framework is separated out into an online micro-clustering component and an offline macro-clustering component. We adopt this general concept and we use k-means for clustering click stream data. In addition we form a generalized model that works on the data many times providing clustering results for certain time periods. These results show the differences between clusters and easily can be extracted trends and usability and security conclusions.

The basic attribute of clustering data streams is one-pass algorithms [1]. Current methods don't address the following issues: 1)The quality of clusters is poor when data evolves considerably over time. 2)A data stream clustering algorithm requires much greater functionality in discovering and exploring clusters over different portions of the stream.

The paper is organized as follows. In Section 2 we describe general the field of clustering data streams and we focus on click stream data. In section 3 we define the problem of clustering giving the motivation of our work. Next in Section 4 we present

the implemented model to perform clustering over click streams using a framework of two components. The presented architecture helps illustrate our main idea. Cluster analysis and possible extensions are described in section 5. Sections 6 and 7 discuss security and usability issues related with the clustering process of click stream data. In Section 8 we present the experimental methodology. The paper wraps up with a conclusion in Section 9.

2. CLUSTERING CLICK STREAM DATA

We study clustering under the click stream model of computation where given a sequence of points, the objective is to maintain a consistently good clustering of sequence observed so far, using a small amount of memory and time [11]. A click stream is defined as the ordered sequence of pages visited or requested by a user in a session [2]. The process of categorizing visitors from their clickstream is a difficult problem since typically the number of possible sequences of page-visits is huge. Moreover the time spent at each page may differ significantly from user to user. Clustering Clickstream data has become a crucial need for scalable and online observation of user actions.

The algorithms that can be applied in this kind of data are relevant to the algorithms used generally for data streams. There are many approximation algorithms for clustering with a single pass. The most common algorithm with high quality results is k-means and its variants.

There are also many techniques that have been proposed for this problem last years, such as [15], [16], [17], [18], but an autonomous procedure that gives a solution is yet to emerge. The main source of data is web access logs and in recent works like [13], [14] authors have proposed algorithms that take into account both the trajectory taken through a website and the time spent at each page even using functions of longest common subsequence of click streams.

The whole implementation is based on [3]. Besides there are some crucial differences and moreover in our implementation we focus in web log data. In our case the online component doesn't uses a storage system to save summary statistics. The whole process takes place in the temporary memory where the web log data are merged. In addition the online component doesn't create micro-clusters but creates an array with fixed size that merges the web log data in the memory. The offline component can be comprised by many different clustering algorithms that work parallel and independently in order to provide a variety of groups of data clusters depending on the current given attributes. In other words we don't propose only a two-phased approach but an intergraded system that can be extended even to a web service that provide combinations of clustering methods in order to give as an output powerful insights of data streams.

3. PROBLEM MOTIVATION

The clustering problem is defined as follows: for a given set of data points, we wish to partition them into one or more groups of similar objects. With the term similar objects we mean similarity between objects that derives from some distance measures or objective functions. Last years this problem is being widely researched in the database, data mining and statistics communities. The motivation here is the large number of relevant applications that are needed to find some alternative ways to handle this data.

4. THE MODEL

In this section we present a new approach on clustering data streams via a data stream clustering model which is guided by some server-centered requirements. The stream data of the proposed implementation are click stream data. The input are these data and the output are a number of clusters.

4.1 Architecture

The main idea is to divide the clustering process in two fundamental phases. In the first phase runs an online component that automatically stores continuously data in a compressed way and in the second phase run some offline components that use these compressed data in order to produce clusters.

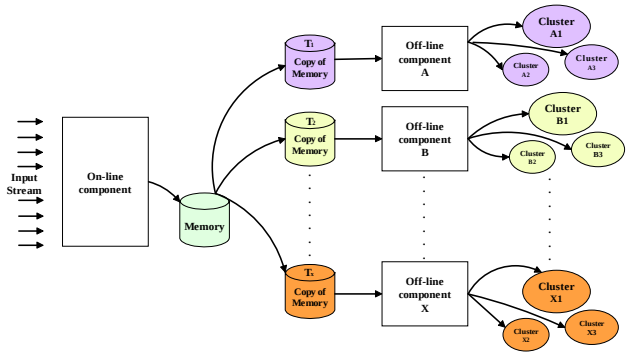

Figure 1. Model architecture.

We propose an integrated solution for performing click stream analysis. For the purposes of our model the input stream that is being handled by the model is the web-click streams of a web server that hosts websites. A click stream is defined as the ordered sequence of pages visited (or requested) by a customer/user in a session. In a web log data preprocessing is to define a session and a customer. Click stream analysis plays an important role in the decision-making process.

4.2 Using weblog data

When a user access a file in a web server a corresponding row of data is available and is being recorded in log files. Log recorded by a web server provide the data for different types of analysis. Different web-servers generate different types of web logs and some are capable of keeping more than one type of log [2].

In our model we don't use these log files because in this case we have to use hard disks of big capacity and also this will aggravate the total running time of our model. The proposed model cancels this process of the creation of log files and uses directly the data of log files in the on-line component.

The log files for the Apache Web server are found in the /var/log/httpd directory (all file names and locations are based on Fedora 4). You'll find a number of logs, the most current one being access_log. The default format of the file is:

Table 1. Default format of access log file

```
192.168.1.4 - - [16/Mar/2006:01:54:44 +0200] "GET
/demo/page1.html HTTP/1.1" 200 313 "-" "Mozilla/5.0
(Windows; U; Windows NT 5.1; en-US; rv:1.8.0.1)
Gecko/20060111 Firefox/1.5.0.1"

192.168.1.4 - - [16/Mar/2006:01:54:50 +0200] "GET
/demo/page2.html HTTP/1.1" 200 313 "-" "Mozilla/5.0
(Windows; U; Windows NT 5.1; en-US; rv:1.8.0.1)
Gecko/20060111 Firefox/1.5.0.1"

192.168.1.2 - - [16/Mar/2006:01:54:52 +0200] "GET
/demo/page3.html HTTP/1.1" 200 313 "-" "Mozilla/5.0
(Windows; U; Windows NT 5.1; en-US; rv:1.8.0.1)
Gecko/20060111 Firefox/1.5.0.1"

192.168.1.2 - - [16/Mar/2006:01:54:54 +0200] "GET
/demo/page5.html HTTP/1.1" 200 313 "-" "Mozilla/5.0
(Windows; U; Windows NT 5.1; en-US; rv:1.8.0.1)
Gecko/20060111 Firefox/1.5.0.1"

192.168.1.4 - - [16/Mar/2006:01:54:58 +0200] "GET
/demo/page5.html HTTP/1.1" 200 313 "-" "Mozilla/5.0
(Windows; U; Windows NT 5.1; en-US; rv:1.8.0.1)
Gecko/20060111 Firefox/1.5.0.1"

192.168.1.3 - - [16/Mar/2006:01:55:20 +0200] "GET
/demo/page1.html HTTP/1.1" 200 313 "-" "Mozilla/5.0
(Windows; U; Windows NT 5.1; en-US; rv:1.8.0.1)
Gecko/20060111 Firefox/1.5.0.1"

192.168.1.3 - - [16/Mar/2006:01:55:24 +0200] "GET
/demo/page8.html HTTP/1.1" 200 313 "-" "Mozilla/5.0
(Windows; U; Windows NT 5.1; en-US; rv:1.8.0.1)
Gecko/20060111 Firefox/1.5.0.1"

192.168.1.5 - - [16/Mar/2006:01:55:27 +0200] "GET
/demo/page3.html HTTP/1.1" 200 313 "-" "Mozilla/5.0
(Windows; U; Windows NT 5.1; en-US; rv:1.8.0.1)
Gecko/20060111 Firefox/1.5.0.1"
```

4.3 Framework

Authors in [3], study the clustering problem for the data stream domain. They introduce a fundamentally different philosophy for data stream clustering which is guided by application-centered requirements. According to their design the framework is separated out into an online micro-clustering component and an offline macro-clustering component. In our framework we adopt this general concept in order to solve a specific problem of clustering click stream data.

To the best of our knowledge there is no previous work about this problem of mining Clickstream data with an on-line framework. Some algorithms on clustering data streams [12] assume that the clusters are to be computed over the entire stream. Such methods simply view data stream clustering problem as a variant of one-pass clustering algorithms. In our problem of clustering click stream data we have to view data as an infinite sequence of data which continuously evolves with time. In the following two subsections we will discuss the framework components of our Clickstream clustering model in detail.

4.3.1 The On-line component

In this component we maintain compressed data of server requests at a continuous way. These summary statistics provide sufficient information about users that access the server. The information that we save in memory with the on-line component are some of the fields that an access log provide. The advantage of this part is that the process is running without breaks and continuously updates efficiently the summary statistics in the memory.

In our first experiment model we use a two-dimensional array and in its rows we have the requested urls and in the rows Ip's that correspond to the current user. In this way we save in the 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. An example of the structure of this array is the array of Table 2 below.

Table 2. Structure of merged data

IP Addresses	Page1	Page 2	Page 3	…...	Page N
192.168.1.1	6	12	1	……	34
192.168.1.2	1	34	21	……	9
192.168.1.7	23	0	1	……	0
192.168.1.3	0	77	9	……	8
192.168.1.22	34	112	87	……	67
192.168.1.121	2	5	48	……	12
192.168.1.4	41	8	11	……	4
……..	…	…	…	……	…

4.3.2 The Off-line component

In this component we use several clustering algorithms, as separate components, that use the array of on-line component to create clusters. The main idea here is to create snapshots of the array in the memory and run in these snapshots the offline components in order to create clusters. This process for every off-line component can take time Ti and after this time we take some new snapshots and we run again every off-line component. The time Ti depends on the runtime of the clustering algorithms that we are going to use and on who often our clusters we speculate to change. At each such time, we store away the snapshot of the array. We also delete the least recent snapshot. After that we have updated clusters even some new. The produced clusters can be stored in a data-base or in the memory. The number of clusters is something that is based on how detailed clusters we want to have every time.

As a first clustering algorithm we used the well known K-means clustering algorithm. K-means take as input the array of counters and make clustering to them. Every Ti that K-means runs produces some cluster 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 with the corresponding array of data.

4.4 Storing Clusters

Clusters refer every time in the Ti we are running the clustering algorithm. Every Ti time we produce new clusters and we store these clusters and the array of data that we have used in this round. In order to have better results we don't have to store the array every time but we can have from the begging an array with some column more that refer to the cluster that the current entry belongs the given time Ti.

Table 3. Structure of merged data

IP Addresses	Page1	Page 2	Page 3	…	Page N	Clu 1	Clu 2	…	Clu M
192.168.1.1	6	12	1	…	34	2	3	…	3
192.168.1.2	1	34	21	…	9	1	3	…	4
192.168.1.7	23	0	1	…	0	4	4	…	55
192.168.1.3	0	77	9	…	8	75	22	…	6
192.168.1.22	34	112	87	…	67	99	4	…	1
192.168.1.121	2	5	48	…	12	8	1	…	2
192.168.1.4	41	8	11	…	4	8	1	…	4
……..	…	…	…	…	…	…	…	…	…

5. CLUSTER ANALYSIS

In this type of clusters an analyst can extract many different conclusions. In correlation with the clustering problem, such an analysis can play an important role. The main idea of cluster analysis is the knowledge by monitoring the changes over clusters during a given time horizon.

In our implementation there are many perspectives and possible extensions that can be applied in order to have better and dipper cluster analysis. In the offline component where k-means perform clustering in our data we can use classification techniques. In other words we can cluster web log data with k-means for one initial time and after that where we will have a number of clusters we can categorize next web log data in order to associate them in the produced initial clusters. In this way data are divided into regions based on class and formulas are generated to predict the output class value. Moreover we can transform k-means clustering algorithm in order to prevent the reproduction of clusters after every run step and keep only the first initial clusters. In this concept we can have the first initial clusters and in next steps of running the input segments of data can be associated to these clusters.

According to these analysis techniques we can have, for a time horizon, the different variations of the initial clusters. Time series analysis can play an important role here as may be viewed as finding patterns in the data and predicting future values. Detected patterns may include trends that can be viewed as systematic nonrepetitive changes to the attribute values over time. As a consequence many straightforward techniques can be used to detect trends in these time series. For example smoothing and correlation are two approaches for finding moving averages of attribute values and to determine the correlations between data values at different lag intervals respectively.

Some final 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.

6. SECURITY ISSUES

In comparison with this model there are some security issues that can be mentioned. First of all there is a clear segregation about security between *clusters* and *segments* of *clusters*. In other words we can meet some open security issues concerning both the clusters and the segments of clusters. The most important of them are the security issues concerning the behavior of the segments that a cluster has.

A suggested security model for data streams could use some rules concerning the behavior of clusters and of the segments in the clusters. For example some rules could be the number of clusters, the size of them, and the anticipated target-cluster of a concrete segment or even the fact of a strange alteration of one segment from a stable cluster to another one. In this example if one of the rules contravenes then we face security problems.

Generally our aim is to have a security model that runs in collaboration with the clustering model of Fig. 1 so as to find automatically differences from the defined rules. According to the model of Fig. 1 if the resulted clusters and their segments at a current time period T1 satisfy the rules related to this off-line component then if in time period T2 the updated clusters and their segments don't satisfy the rules then the security model stores the necessary information about the current problem.

Other issues are the protection of privacy of information sources and the exercise of control over potential data mining results [19]. According to the first issue we mean the ability to stop data mining projects to have access into individual data. This is a very common problem especially resent years that has been adapted by many companies and states. In this issue are included and other problems like data mining techniques that can give the ability to predict individual data values. According to the second issue we discuss problems that derive from the bad use of the available results. There is no assurance that data mining techniques will not misuse data and will not reveal private information.

As a conclusion data mining techniques in general and in our case the proposed model for clustering click stream data have to consider issues about safely releasing data.

7. USABILITY ISSUES

The data clustering method can plays an important role in the usability factor of the Data Stream Applications. The ISO 9241-11 [4] describes usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". A usable system is one that is designed to consider all the aforementioned features.

Since usability is a term too abstract to be studied directly, it is usually divided into the following terms, identified by Nielsen [5]:

1. Learnability: Is a measure of how much training is required before a specified level of proficiency is reached.
2. Efficiency: The attribute of a system that allows for the user to maximize productivity.
3. Memorability: This shows how easily a user recalls how to reuse a system after a period of time and measures how robust the learning and performance are.
4. Errors: It's a measure of the accuracy of the work carried out to complete tasks.

5. Satisfaction: Is the feeling that a user must have after the use of a system.

Until today both user testing with real users and questionnaires were the most common usability evaluation methods. In this work we introduce an alternative method for measuring usability in data stream applications. By applying some clustering analysis in the output clusters we can find some metrics in order to find out if the data stream application is usable in a high level.

In our implementation:

- Assume that we have a number of clusters of users that correspond to the web pages that the web server hosts. If a cluster is very big and the path from the index page to the page of this cluster is not direct then we can conclude that the web site faces usability problems (Efficiency, Memorability, Satisfaction).
- Assume that we have clusters that correspond to pages that don't belong to our page set. This is another usability problem (Errors).

Finally usable data stream applications can lead the quality of provided services to a high level.

8. EXPERIMENTAL METHOLOGY

Our model can be divided into two parts. The first part contains the online component that takes the data from the web server and creates an array in the memory with the compressed information. The second part contains the offline component where we employ k-means clustering in this array.

8.1 Data extraction

As soon as the web server is running, the next essential step is to design and implement a methodology for extracting requested data from this server. For this purpose an engine which operates on the web server at a very low level was built in order to read every request on the server and create an array in the memory with summary information for every request. This engine runs continuously in parallel with the server.

The basic requirements for this engine were:

- To operate without a need to pre-process or compile files.
- To handle all expected requests for all type of files. The engine should identify patterns in these requests in order to handle only requests for page files and then use them to fill the array.
- To execute swiftly and handle arbitrary amounts of data requests.

Based on these requirements we employed regular expressions in order to implement the engine, as these can efficiently find for patterns of file types and names among the requests. The output array that represent the input for clustering is presented bellow.

Table 4. Input for clustering

Name	Description
User's IP	Its values are identification of the corresponding input record to the fitting cluster
Page counter	The maximum number of page visits of every user (ip) for every page

8.2 Data clustering

We propose here an offline framework for using data mining techniques in order to facilitate comprehension of user's actions over a web site, as depicted in Fig. 1. First, we employ K-means clustering on data extracted from the web server. Clustering is more suitable for this purpose because it produces overviews of user's behavior by creating mutually exclusive groups of ip's according to their similarities, thus reducing the time required to understand the overall actions of users. Clustering has also the potential to discover user patterns and trends even "unusual" or outlier cases which may require further attention for security and usability issues. As soon as clustering is completed, the following type of information is created as its output, which is in turn used as input for further mining techniques and cluster analysis:

- Cluster ID.
- User's ip (ID) as it is the identification for each user.

8.2.1 K-Means Clustering Parameters

As described *K-Means clustering* algorithm was employed in this step of research work. Some of its key features include the need for the user to define the number of derived clusters and the initial centroids as is presented in Table 5. The output parameters of k-means are presented in Table 6.

Table 5. Input parameters of k-means

Name	Description
Input Dataset	The given dataset
Number of iterations	The number of iterations to perform clustering.
Number of Clusters	The number of clusters the algorithm generates.
Values of centroids	The value of the initial centroids from where the algorithm will start to examines the data set.

Table 6. Output parameters of k-means

Name	Description
Cluster ID	The identifier of the fitting cluster for the corresponding input record.
User's IP	Its values are identification of the corresponding input record to the fitting cluster

Further experimental results will be available in the full version of the paper.

9. CONCLUSIONS

In this paper, we used a common clustering algorithm into a modified framework to extract a set of clusters in order to define

the behavior and the trends of users that access web pages in a web server.

10. REFERENCES

[1] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, Philip S. Yu, *A Framework for Diagnosing Changes in Evolving Data Streams*, Proceedings of the ACM SIGMOD Conference, 2003.

[2] Demiriz, A.. *On analyzing web log data: A parallel sequence mining algorithm*. In Kantardzic, M., Zurada, J., eds.: New Generation of Data Mining Applications (To be published), IEEE-Wiley (2003)

[3] C. C. Aggarwal, J. Han, J. Wang, P. Yu. *A Framework for Clustering Evolving Data Streams.* Proceedings of the VLDB Conference, 2003

[4] *ISO/IEC. 9241-14 Ergonomic requirements for office work with visual display terminals (VDT)s - Part 14 Menu dialogues, ISO/IEC 9241-14: 1998 (E),* 1998.

[5] Nielsen J (1993). *Usability Engineering J*, Academic Press, Inc., San Diego

[6] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan, *Clustering Data Streams: Theory and Practice* TKDE special issue on clustering, vol. 15, 2003.

[7] Kevin L. Chang and Ravi Kannan, *The space complexity of pass-efficient algorithms for clustering*, Proceedings of the SODA '06

[8] Ke Chen, *On k-Median clustering in high dimensions*, Proceedings of the SODA '06

[9] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. *Clustering data streams: Theory and practice. TKDE special issue on clustering*, 15, 2003.

[10] Arindam Banerjee and Joydeep Ghosh, "*Clickstream Clustering using Weighted Longest Common Subsequences*", Workshop on Web Mining of the 1st SIAM Intl conference on Data Mining

[11] Sudipto Guha and Nina Mishra and Rajeev Motwani and Liadan O'Callaghan, *Clustering Data Streams*, IEEE Symposium on Foundations of Computer Science

[12] L. O'Callaghan et al. *Streaming-Data Algorithms For High-Quality Clustering*. ICDE Conference, 2002

[13] A. Banerjee and J. Ghosh. *Concept-based clustering of clickstream data*. In Proc. 3rd Intl. Conf. on Information Technology, pages 145--150, Dec 2000

[14] A. Banerjee and J. Ghosh. *Clickstream clustering using weighted longest common subsequences.* In Proceedings of the Web Mining Workshop at the 1 st SIAM Conference on Data Mining, Chicago, April 2001.

[15] Fu, Y., Sandhu, K., Shih, M. *A Generalization-Based Approach to Clustering of Web Usage Sessions*, in Proc. of WEBKDD 1999 (San Diego CA, August 1999), 21-38.

[16] M.-S. Chen, J. S. Park, and P. S. Yu. *Efficient Data Mining for Path Traversal Patterns*. IEEE Trans. on Knowledge and Data Engineering, 10(2):209-221, March 1998

[17] C. Shahabi, A. M. Zarkesh, J. Adibi, and V. Shah. *Knowledge discovery from users web-page navigation.* Accepted for publication in IEEE RIDE'97

[18] R. Cooley and J. Srivastava and B. Mobasher, *Web Mining: Information and Pattern Discovery on the World Wide Web*, Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), November 1997.

[19] C. Clifton and D. Marks. *Security and privacy implications of data mining.* In Proc. 1996 SIGMOD '96 Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'96), pages 15--20, Montreal, Canada, June 1996.