A new approach to session identification by applying fuzzy c-means clustering on web logs

Georgios Alexandridis

A new approach to session identification by applying fuzzy c-means clustering on web logs

Dimitrios Koutsoukos*, Georgios Alexandridis†, Georgios Siolas‡, Andreas Stafylopatis§
School of Electrical and Computer Engineering
National Technical University of Athens
Zografou, 157 80, Greece
Email: *koutsoukos.dimitr@gmail.com, †gealexandri@islab.ntua.gr, ‡gsiolas@islab.ntua.gr, §andreas@cs.ntua.gr

Abstract—**In this paper a new algorithm for session identification in web logs is outlined, based on the fuzzy c-means clustering of the available data. The novelty of the proposed methodology lies in the initialization of the partition matrix using subtractive clustering, the examination of the effect a variety of distance metrics have on the clustering process (in addition to the widely-used Euclidean distance), the determination of the number of user sessions based on candidate sessions and the representation of the session data. The experimental results show that the proposed methodology is effective in the reconstruction of user sessions and can distinguish individual sessions more accurately than baseline time-heuristic methods proposed in literature.**

I. Introduction

Examining web data for the extraction of useful access patterns has been a very active research area over the past two decades [1]. More specifically, session identification is the process of locating and extracting from web server logs the resources requested by a specific user, during the different visits he or she performs on the website in the course of time. Discovering these access patterns is useful in a number of ways; i.e. for personalizing the website, for the prefetching of links and for the improvement of the web server performance.

Access logs may be gathered at two different locations; either on the client-side (the web browser) or the web server side. In the first case, a browser plugin (or a JavaScript application) actively monitors the pages visited by a specific user and the time spent on each page and periodically feeds the collected information to a monitoring server. This methodology exhibits certain advantages, such as making the user and session identification tasks straightforward and pushing the computational load on the client side, therefore saving on computing resources. However, it is easy for the end user to "mask" his or her presence, by blocking the monitoring procedure (e.g. through the installation of 3rd party software, such as Ad-Block) as recent research indicates [2].

Server-side collection of web usage information, on the other hand, is performed by the web server application itself. Every request received is stored either into a database or in predefined log files at the web server. This procedure has the advantage of being able to record all information exchanged between a website and its visitors, regardless of the use of blocking software by the latter. However, this comes at the cost of rendering the user and session identification process a difficult task, for a number of reasons (no user-identifiable information in the request, multiple users accessing the website through a proxy etc).

In this work, we propose a novel algorithm for session identification, based on the fuzzy c-means clustering of the available web logs. The novelty of our approach lies in the specific choices made concerning the initialization of the fuzzy partition matrix (using subtractive clustering) as well as the examination of the effect a variety of distance metrics have on the clustering process, in addition to the widely-used Euclidean distance. On top of that, the number of user sessions in the underlying data is approximated with candidate sessions, using a most appropriate representation for the underlying data. The following sections describe the proposed methodology in more detail, along with its ability to uncover user sessions on a privately collected dataset, especially when compared to other time-heuristic methods proposed in literature.

II. Related work

Session identification based on web log data has been extensively studied in literature. The very first approaches to this task resorted to time heuristics, by defining a maximum timeout threshold between two consecutive user requests in order to place them in the same session. This threshold may be set to several values, based on empirical data. For example, it has been determined that a 25.5 minute period exhibits the most accurate results [3]. Other choices range from 10 minutes to 24 hours [4], but a widely accepted timeout value used both in literature and commercial websites is 30 minutes [5].

However, a fixed timeout threshold does not take into account that various pages may have completely different content in terms of volume as well as interest for each user. Additionally, other parameters that may affect the time that a user spends in a web page, such as each individual's reading speed, the site topology, other distractions, etc. In order to account for the aforementioned cases, dynamic timeout thresholds have been proposed in literature, based on traditional session identification algorithms [6].

In addition to the time-heuristics, another approach to session identification are the navigation-oriented heuristics [7], which construct a graph of the website, based on its structure. In this setting, individual web pages are represented as nodes and links in between them as edges among the respective nodes. The sessions are split according to whether a user's

TABLE I
EXAMPLE EXCERPT FROM A WEB LOG, IN ACCORDANCE TO THE EXTENDED COMMON LOG FORMAT (CLF)

123.456.789.123 [29/Mar/2016:12:44:03] "GET / HTTP/1.1" 200 612 "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)" http://www.google.com/
123.456.789.123 [29/Mar/2016:12:44:03] "GET /image1.jpg HTTP/1.1" 200 854 "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)" /index.html
123.456.789.123 [29/Mar/2016:12:44:03] "GET /style.css HTTP/1.1" 200 143 "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)" /index.html
456.789.123.456 [29/Mar/2016:12:48:29] "GET /robots.txt HTTP/1.1" 200 98 "Mozilla/5.0 (compatible; bot)" -
789.123.456.789 [29/Mar/2016:12:55:08] "POST /form.php HTTP/1.1" 200 558 "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:45.0)" http://www.s2.com/

consecutive requests are connected by a direct edge. If there is not a connection between the currently visited web page and the previous one, then a new session is considered to have started.

The methodology above may be easier implemented when the referrer attribute exists in the logs [5] (Sec. III). For example, consecutive requests with the same referrer attribute may be assigned to the same session. Otherwise, a new session is thought to have been initiated. Of course, time heuristics and referrer-based methods may be combined together for the better identification of user sessions [8].

Recently, fuzzy clustering approaches have been introduced to the session identification problem [9], where user sessions are being extracted according to time heuristics and then weights are assigned using fuzzy membership functions of a number of characteristics, including the frequency of each URL, the number of bytes downloaded and the time elapsed between two requests. The preprocessed requests are subsequently being clustered using both the *fuzzy c-means* and the *fuzzy c-medoids* algorithms, employing *mountain density functions* in the initialization process (in order to estimate the number of clusters). Finally, the quality of the produced clustering is assessed by certain validity indices.

Our approach is inspired from the methodology described above, especially in the use of fuzzy c-means clustering. However, a number of fundamental design choices are completely different. Firstly, the frequency of each URL is not taken into account. Secondly, the chosen representation tries to simplify the requests into time data for the clustering algorithm. Subsequently, the fuzzy partition matrix is initialized using subtractive clustering which is expected to be faster than mountain clustering because it considers fewer candidate points as cluster centers. Finally, different distance metrics between the cluster centroids and the data points are used, thereby affecting the results of the clustering as well as the convergence of the algorithm.

III. DATA PREPARATION

A. Data cleaning

Even though web server logs usually contain a variety of information about each specific request, the majority of modern web servers' logging functionality adheres to the *Common Log Format* (CLF) [10], a de facto standard imposed by the World Wide Web Consortium (W3C). Most commonly, two extra fields are added to the aforementioned standard, that of the *User Agent* and the *Referrer*, yielding to the *Extended Common Log Format* (ECLF).

Table I provides an excerpt from a sample log file, in accordance to the ECLF. Each line of the table corresponds to single request for a specific resource (html file, image, style sheet, etc) of the web server. The log entry consists of several fields, such as the remote host making the request (IP address or domain name), user authentication information (if available), a timestamp, the HTTP method (GET, POST, etc), the actual resource requested, the web server's response status code (200, 301, 404, etc), the total number of bytes exchanged in the request, the *User Agent* field that characterizes the operating system and the browser used by the client and finally the *Referrer* field, which contains the URL that this specific request originated from.

In general, web log files contain more data than it is actually required for session identification. For this reason, a data cleaning step is necessary. Initially, all unsuccessful requests (those that have HTTP status codes other than 2xx and 3xx) are removed. Following, all activity originating from bots/spiders/crawlers etc is excluded as well. This is achieved by examining the *User Agent* field (e.g. 4th entry in Table I).

The final stage of the data cleaning operation involves the removal of all those resources that the user does not directly request, but are eventually asked for by the browser itself (in order to properly render the page the user wanted to view). These resources include image files, style sheets, javascript programs, etc. As a result, only the log entries that contain requests to either HTML pages or to server-side scripts (ASP, JSP, PHP, etc) are retained. For example, the application of the data cleaning process in the excerpt of Table I would result in the removal of the 2nd and 3rd entries (because they are indirectly retrieved by the browser) and the 4th entry (because it is a request made by a bot).

B. User identification

After the data cleaning phase is over, we proceed to identify individual users. In some cases, this is a straightforward task, as the ECLF standard permits the storage of user identification data. However, when the aforementioned information is absent from the logs, other fields of the ECLF have to be used (e.g. *IP Adddress/Host*) in order to identify the user behind each request.

The simpler approach would be to assign each IP address/Host to a different user [11]. This choice is sound for short periods of time (e.g. in the vicinity of minutes or few hours) and yields satisfactory results. For longer periods of time, thought, the user identification process has to be extended by incorporating the information included in the *User Agent* field as well. Proper analysis of the said field permits the

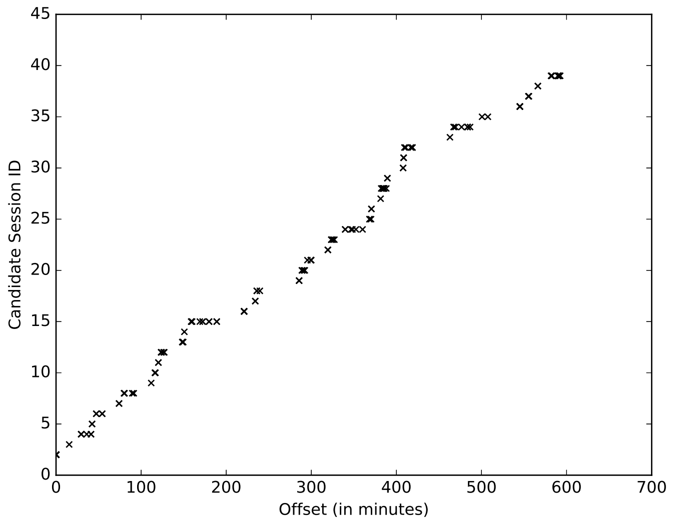

Fig. 1. Candidate session assignment of the requests made by a sample user

extraction of the operating system and the browser name and version each request originated from, leading to the formation of the triplet {`Host, Operating System, Browser`}. This triplet contains sufficient information to identify each and every user for longer time frames (that may exceed 24 hours) given that IP addresses/domains belonging to known proxy servers have been filtered out.

The last step of the data preparation process involves the transformation of the available data to a suitable form for the session identification algorithm itself (to be outlined in the following section). For this reason, the timestamps of the requests made by each user (as identified by the triplet referenced above) have to be transformed. A timestamp of zero is assigned to the first request and an offset in time from the first request is assigned to all subsequent requests belonging to the same user.

After fixing the timestamps, each request is appointed to a candidate session. Two consecutive requests may either belong to the same session or to adjacent sessions. The latter is the case when the time difference between the aforementioned requests is more than 10 minutes or there is a change in the *Referrer* field [7]. Fig. 1 illustrates this procedure for a sample user.

IV. Fuzzy c-means clustering

The identification of the number sessions and their boundaries (Fig. 1) is a cumbersome task, where hard partitioning approaches (e.g. *k-means clustering*) are not expected to perform well. Soft partitioning approaches on the other hand, like the the *fuzzy c-means clustering* [12], allow each data point to belong to one or more classes and therefore seem to be a more natural fit for this type of problems.

A. Fuzzy partition matrix

Let us assume that our objective is to cluster a set $\mathbf{X}$ of n data points in c fuzzy classes. Each data point $\mathbf{x_k}$ is represented as a vector of l elements. An obvious constraint on the number of possible classes those points might belong to, is the following

$$2 \leq c < n \tag{1}$$

Furthermore, let A denote a family of sets, where $\mathbf{A_i}$ is the set of points belonging to the i-th class. Initially, a membership value in each class is assigned for every data point. That is, the membership value (μ_{ik}) of the k-th data point to the i-th class is defined as follows

$$\mu_{ik} = \mu_{A_i}(\mathbf{x_k}) \in [0,1] \tag{2}$$

Any given data point may be a member, to some degree, of any given class; however the sum of its membership values must be exactly one

$$\sum_{i=1}^{c} \mu_{ik} = 1, \quad \forall k = 1,2,\ldots,n \tag{3}$$

Another limitation is that there can be no empty classes and corollary there can be no class that contains all the data points

$$0 < \sum_{k=1}^{n} \mu_{ik} < n, \quad \forall i = 1,2,\ldots,c \tag{4}$$

A family of *fuzzy partition matrices* M_{fc} is defined over $\mathbf{X}$ for the classification in c classes, abiding to the constrains of Equations 3 and 4

$$M_{fc} = \left\{ U | \mu_{ik} \in [0,1]; \sum_{i=1}^{c} \mu_{ik} = 1; 0 < \sum_{k=1}^{n} \mu_{ik} < n \right\} \tag{5}$$

Any matrix U that is a member of the family of matrices defined in Eq. 5 above ($U \in M_{fc}$) constitutes a fuzzy c-partition of the set $\mathbf{X}$ of data points. The rows of U represent the classes, the columns represent the data points and the elements represent the membership of the specific data point to the given class.

B. Fuzzy c-means clustering algorithm

1) Euclidean distance: The *fuzzy c-means* (FCM) clustering algorithm computes a valid fuzzy partitioning of the input data in c clusters. This is achieved by assigning every data point to one or more clusters, based on its distance from each cluster center. The most widely used distance metric in literature is the Euclidean distance, yielding to the objective function of Eq. 6

$$J_m(U,V) = \sum_{k=1}^{n}\sum_{i=1}^{c}(\mu_{ik})^m d^2(\mathbf{x_k},\mathbf{v_i}) \tag{6}$$

where V is the set of cluster centers and $d(\mathbf{x_k},\mathbf{v_i})$ is the Euclidean distance between the k-th data point and the center of the i-th cluster ($\mathbf{v_i}$)

$$d_{\text{euclidean}}(\mathbf{x_k},\mathbf{v_i}) = ||\mathbf{x_k}-\mathbf{v_i}||_2^2 = \sqrt{\sum_{j=1}^{l}(x_{kj}-v_{ij})^2} \tag{7}$$

The exponent m of Eq. 6 is called the *weighting parameter* [12] (or the *fuzzifier*) and it controls the amount of fuzziness in the clustering process, taking values in $[1,+\infty)$. Smaller values of the objective function correspond to a better partitioning of the input data; therefore computing the optimal partitioning may be expressed as the following minimization problem

$$\arg\min_{M_{fc}} J_m(U,V) \tag{8}$$

Constrained optimization problems, such as that of Eq. 8 above, cannot be solved deterministically. They may be approximated, however, by using Lagrange multipliers and then by computing the partial derivatives of the Lagrange function with respect to the membership values and the cluster centers [13]. Equating the partial derivatives to zero and keeping fixed one set of the variables, the values of the other set may be computed. This proccess is continued iteratively until a given level of accuracy ϵ has been reached (Alg. 1).

In the special case of the Euclidean distance discussed here, fixing the membership values leads to the following equation for computing the cluster centers (it should be noted that each cluster center is itself an l-dimensional vector as well)

$$v_{ij} = \frac{\sum_{k=1}^{n}(\mu_{ik})^m x_{ki}}{\sum_{k=1}^{n}(\mu_{ik})^m} \tag{9}$$

Algorithm 1 Iterative Optimization Algorithm for Euclidean FCM

```
Require: Number of classes $c$, fuzzifier $m$, accuracy level $\epsilon$
Input: $c \times n$ empty fuzzy partition matrix $U = \{\mu_{ik}\}$
Output: Fuzzy partition matrix $U$
1: Initialize $U^{(0)}$
2: repeat
3:     Update the $c$ cluster centers $\mathbf{v}^{(t)}$ according to Eq. 9
4:     Update the membership values $m_{ik}$ according to Eq. 10
5: until $||U^{(t)} - U^{(t-1)}|| \leq \epsilon$
```

After updating the cluster centers, the values of the membership functions are renewed according to Eq. 10 below

$$\mu_{ik} = \left[\sum_{j=1}^{c}\left(\frac{d(\mathbf{x_k},\mathbf{v_i})}{d(\mathbf{x_k},\mathbf{v_j})}\right)^{\frac{2}{m-1}}\right]^{-1} \tag{10}$$

2) L_p norms: The Euclidean distance discussed before is a special case of a more general distance metric, the *Minkowski* distance, defined below

$$d(\mathbf{x_k},\mathbf{v_i}) = ||\mathbf{x_k}-\mathbf{v_i}||_p^p = \sqrt[p]{\sum_{j=1}^{l}(x_{kj}-v_{ij})^p}, \quad p \in [1,+\infty) \tag{11}$$

Eq. 11 is also referred to as the L_p-norm and even though p may reach positive infinity, it most usually lies in $[1,2]$.

Other interesting cases of the Minkowski distance include the *Manhattan* distance ($p=1$)

$$d_{\text{manhattan}}(\mathbf{x_k},\mathbf{v_i}) = ||\mathbf{x_k}-\mathbf{v_i}||_1^1 = \sum_{j=1}^{l}(x_{kj}-v_{ij}) \tag{12}$$

and the *Chebyshev* distance ($p=+\infty$)

$$d_{\text{chebyshev}}(\mathbf{x_k},\mathbf{v_i}) = ||\mathbf{x_k}-\mathbf{v_i}||_{+\infty}^{+\infty} = \lim_{p\to+\infty}\sum_{j=1}^{l}(x_{kj}-v_{ij})^p \tag{13}$$

In order to be able to use the Minkowski distance in the objective function of Eq. 6 and therefore in the minimization problem of Eq. 8, it must be re-written in quadratic form [12]

$$d_{\text{minkowski}}(\mathbf{x_k},\mathbf{v_i}) = \sqrt{(\mathbf{x_k}-\mathbf{v_i})^\top A(\mathbf{x_k}-\mathbf{v_i})} \tag{14}$$

where A is a positive definite $l \times l$ matrix. Setting the matrix A to a suitable form yields the desired distance function. For example, if $A = I$, Eq. 14 computes the Euclidean distance; if $A = E, \quad (e_{ij} = 1, \forall i,j)$, it computes the Manhattan distance and so forth. As a result, Eqs. 9-10 along with Alg. 1 may be used with the distance functions presented above as well.

C. Partition matrix initialization

The iterative algorithm for FCM outlined in Sec. IV-B is very sensitive to the initialization of the fuzzy partition matrix (first step of Alg. 1). A poor choice on the initialization parameters may lead to a local optimum of the objective function (Eq. 6) instead of the global one. This issue may be addressed by the techniques presented next.

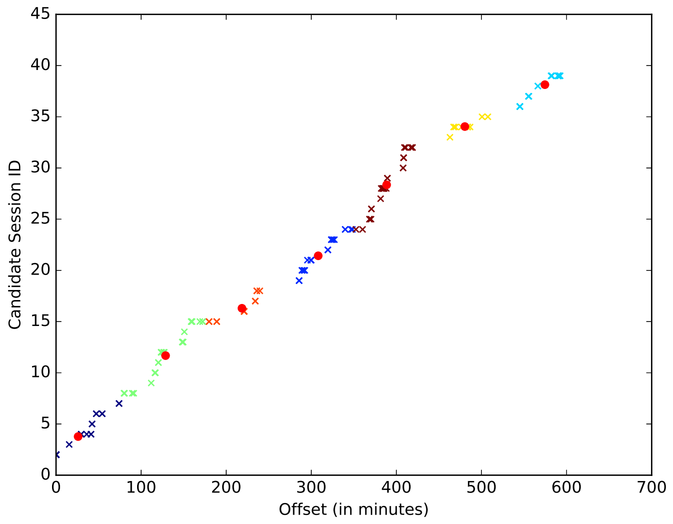

Fig. 2. Cluster assignment of the requests of the same sample user. Circles represent the cluster centers (color online)

1) Mountain clustering: The *mountain clustering* method [14] is used to estimate the number and initial locations of the cluster centers. Firstly, a grid is formed on the data space and the potential cluster centers are positioned inside the grid (e.g. the central points). Secondly, a *mountain function* is computed for those potential cluster centers, that represents a data density measure (Eq. 15 below)

$$m(\mathbf{v}) = \sum_{k=1}^{n} \mathrm{e}^{-\alpha \mathrm{d}(\mathbf{v},\mathbf{x}_k)} \tag{15}$$

where $\mathrm{d}(\mathbf{v}, \mathbf{x}_k)$ is a distance metric between the potential cluster center $\mathbf{v}$ and $\mathbf{x}_k$, the k-th data point (α is a constant that determines the smoothness of the mountain function).

The form of Eq. 15 designates that the data density measure at a point $\mathbf{v}$ is affected by the amount of data points that are near it. The more data points that are near the tested point, the higher the value of the mountain function is.

Cluster center selection is based on the value of the mountain function. More specifically, the first cluster center $\mathbf{c}_1$ is defined to be the point with the highest value. After it has been selected, the form of the mountain function has to be revised, in order to eliminate its effect. Therefore, Eq. 15 becomes

$$m_{new}(\mathbf{v}) = m(\mathbf{v}) - m(\mathbf{c}_1)\mathrm{e}^{-\beta \mathrm{d}(\mathbf{v},\mathbf{c}_1)} \tag{16}$$

where β is a new smoothing factor.

The second cluster center $\mathbf{c}_2$ is selected to be the point where $m_{new}(\mathbf{v})$ reaches its maximum. Again, after fixing $\mathbf{c}_2$, Eq. 16 has to be revised in order to eliminate the effect of the newly appointed cluster center. This iterative process continues until the desired number c of clusters has been reached.

2) Subtractive clustering: The main drawback of the mountain clustering method described above is that it is computationally expensive, as the number of calculations grows exponentially with the dimensions of the problem. In order to amend this disadvantage, the *subtractive clustering* method has been proposed [15]. Its main difference to mountain clustering is that it uses data points as candidate cluster centers instead of grid points.

Under this modification, the required computations now become proportional to the problem size. Even though in many problem domains it is not necessary for the cluster centers to coincide with the data points, this approximation is good enough, when the reduced computations are considered.

Since every data point is a potential cluster center, the density measure is computed as

$$D_k = \sum_{j=1}^{n} \mathrm{e}^{-\alpha ||\mathbf{x}_k - \mathbf{x}_j||^2}, \quad \alpha = \frac{4}{r_a^2} \tag{17}$$

where r_a is a neighborhood radius.

As in the case of mountain clustering, high density values mean that a data point has a lot of neighboring points. Again, the first cluster center $\mathbf{x}_{c_1}$ is set to the data point with the highest density value. After $\mathbf{x}_{c_1}$ has been fixed, the density measure has to be revised, in order to eliminate its effect in the selection of the second cluster center

$$D_{k,\mathrm{new}} = D_k - D_{c_1}\mathrm{e}^{-\beta ||\mathbf{x}_k - \mathbf{x}_{c_1}||^2}, \quad \beta = \frac{4}{r_b^2} \tag{18}$$

where r_b is a positive constant that defines a neighborhood in which the density measure will be reduced to. This procedure is iteratively repeated, until the desired number c has been reached, selecting a new cluster center at each step and amending the density function accordingly. Fig. 2 illustrates the result of the session identification for the sample user of Fig. 1; the red circles denote the uncovered cluster centers (different sessions of the same user).

V. Validity indices

The FCM algorithm analyzed in the previous section (Algorithm 1) requires that the number of clusters c is specified beforehand. However, in many problem domains, including the one addressed in this work, this number of clusters is not known and it is in fact part of the desiderata themselves. After all, different number of initial clusters would result in different partitioning of the input data.

It is therefore evident that a way of assessing the quality of the produced clusterings is more than necessary. To this end, a number of *validity indices* have been proposed in literature [16]. In general, they fall in two categories; those based on the membership values only and those examining both the membership values and the input data.

The first validity index to be examined in this work is the Fukuyama and Sugeno (FS) index [17], that combines both the membership values and the data points

$$\begin{aligned} V_{\mathrm{FS}} =& J_m(U,V) - K_m(U,V) \\ =& \sum_{i=1}^{c}\sum_{k=1}^{n} \mu_{ik}^m \mathrm{d}^2(\mathbf{x}_k,\mathbf{v}_i) - \sum_{i=1}^{c}\sum_{k=1}^{n} \mu_{ik}^m \mathrm{d}^2(\mathbf{v}_i,\bar{\mathbf{v}}) \end{aligned} \tag{19}$$

where $\bar{\mathbf{v}}$ is the mean of the cluster centers and $\mathrm{d}(\mathbf{x}_k,\mathbf{v}_i)$ is a distance metric. The term $J_m(U,V)$ quantifies the compactness of the representation with respect to the c cluster centers while the term $K_m(U,V)$ measures the distance of each cluster center to their mean. The optimal number of clusters c^* is found by solving the following minimization problem

$$\underset{2\le c\le n-1}{\arg\min}\, V_{\mathrm{FS}} \tag{20}$$

A similar validity index is the Xie and Beni (XB) index [18]

$$V_{\mathrm{XB}} = \frac{J_m(U,V)}{n\cdot \min\limits_{i,j} \mathrm{d}^2(\mathbf{v}_i,\mathbf{v}_j)} \tag{21}$$

which tries to combine two desired properties of a good clustering; compactness and separation. The numerator computes how compact the fuzzy partitions are and the denominator how well distinguished the cluster centers are. The optimal number of clusters c^* is once again expressed as a minimization problem

$$\underset{2\le c\le n-1}{\arg\min}\, V_{\mathrm{XB}} \tag{22}$$

The last validity index to be examined is the partition coefficient and exponential separation (PCAES) index [19]

$$\begin{aligned} V_{\mathrm{PCAES}} &= \sum_{i=1}^{c} \mathrm{PCAES}_i, \\ \mathrm{PCAES}_i &= \frac{1}{\mu_M}\sum_{k=1}^{n}\mu_{ij}^2 - \exp\left\{-\min_{k\ne i}\frac{d^2(\mathbf{v}_i,\mathbf{v}_k)}{\beta_T}\right\} \end{aligned} \tag{23}$$

where

$$\mu_M = \min_{1\le i\le c}\sum_{k=1}^{n}\mu_{ij}^2,\quad \beta_T = \frac{1}{c}\sum_{i=1}^{c} d^2(\mathbf{v},\bar{\mathbf{v}}),\quad \bar{\mathbf{v}} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_k \tag{24}$$

TABLE II
Dataset characteristics

No. of log entries (initial)	5,148,780
No. of log entries (after cleaning)	121,117
No. of unique URLs	36,281
No. of unique sessions	14,133

Higher values of V_{PCAES} designate the compactness and separation of each cluster from the others while lower values indicate that at least some of the clusters are not that compact or well separated from the others. The optimal number of clusters c^* is approached by solving the following maximization problem

$$\underset{2\le c\le n-1}{\arg\max} V_{\mathrm{PCAES}} \tag{25}$$

VI. Experiments

The proposed methodology has been tested on a private web log dataset, collected from the Online Community of the Students of the School of Electrical and Computer Engineering at NTUA [20], over a period spanning 14 days in November, 2015. The web logs were in the form of text files, adhering to the Extended Common Log Format and they also contained ground truth information about the user sessions (Table II). Initially, the dataset was cleaned, according to the reasoning of the Section III-A; requests with status codes not in the 200 and 300 ranges were removed, along with all bot activity (crawlers, spiders, feed readers etc). Following, all requests that did not involve directly an HTML resource (such as images, fonts, style sheets, javascript code) were removed as well. Lastly, the IP addresses of the remaining requests were checked against known lists of proxy servers. Only a small percentage of proxy servers were found in the remaining logs, which were removed as well.

After the completion of the data cleaning phase, we proceeded to user identification, as discussed in Section III-B. The browser and the operating system each request originated from were extracted from the *User Agent* field. Requests that matched all three fields, occurring within a 24 hour time frame (from the first appearance) were considered to come from the same user and therefore were assigned to the same user id. Users with too few requests were removed from our test set. Finally, on the remaining user ids we performed the candidate session id assignment outlined in Section III-B (Fig. 1) and then presented the input data to the session identification algorithms discussed below.

Two families of algorithms have been tested; a time-heuristics based one, serving as the baseline, and the FCM, outlined in Section IV-B. The default time out period of the former algorithm was set to be 25.5 minutes, because this value has been predominantly used in literature (Section II).

The FCM algorithm is dependent on a number of configuration parameters. Firstly, the initialization of the fuzzy partition matrix must be accounted for. This issue has been tackled by following both approaches presented in Section IV-C; that of mountain clustering (Section IV-C1) and that of subtractive

TABLE III
RESULTS

Algorithm	Effective Rate			Identification Rate			F1 Score		
Validity indices (for the FCM algorithm)	V_{FS}	V_{XB}	V_{PCAES}	V_{FS}	V_{XB}	V_{PCAES}	V_{FS}	V_{XB}	V_{PCAES}
Time-heuristic algorithm (Threshold 25.5 min)	53.88 %			40.37 %			46.16 %		
Fuzzy c-means - Mountain clustering									
Euclidean distance, $m = 2$	61.40 %	90.10 %	39.27 %	67.96 %	39.81 %	94.17 %	64.51 %	55.22 %	55.43 %
Euclidean distance, $m = 5$	41.73 %	75.19 %	24.53 %	51.46 %	**47.09 %**	69.41 %	46.09 %	**57.91 %**	36.25 %
Manhattan distance, $m = 2$	59.91 %	90.10 %	36.26 %	67.48 %	39.81 %	**96.11 %**	63.47 %	55.22 %	52.65 %
Manhattan distance, $m = 5$	41.29 %	75.59 %	27.52 %	52.91 %	46.60 %	72.82 %	46.38 %	57.66 %	39.94 %
Chebyshev distance, $m = 2$	61.59 %	89.23 %	26.52 %	45.15 %	28.16 %	69.90 %	52.10 %	42.81 %	38.45 %
Chebyhsev distance, $m = 5$	**82.61** %	75.59 %	21.14 %	36.89 %	46.60 %	57.77 %	51.00 %	57.66 %	30.95 %
Minkowski distance, $m = 2$, $p = 1.5$	61.30 %	**90.22 %**	**40.25 %**	**68.45 %**	40.29 %	95.14 %	**64.68 %**	55.70 %	**56.57 %**
Minkowski distance, $m = 5$, $p = 1.5$	41.90 %	75.00 %	25.91 %	51.46 %	46.60 %	68.93 %	46.19 %	57.48 %	37.66 %
Fuzzy c-means - Subtractive clustering									
Euclidean distance, $m = 2$	42.07 %	84.85 %	41.15 %	81.07 %	**40.78 %**	83.50 %	55.39 %	**55.09 %**	55.13 %
Euclidean distance, $m = 5$	46.38 %	68.32 %	27.94 %	46.60 %	33.50 %	70.39 %	46.49 %	44.96 %	40.00 %
Manhattan distance, $m = 2$	42.60 %	85.57 %	40.61 %	**81.17 %**	40.29 %	**83.98 %**	55.85 %	54.78 %	54.75 %
Manhattan distance, $m = 5$	46.60 %	62.71 %	27.54 %	46.60 %	35.92 %	68.45 %	46.60 %	45.68 %	39.28 %
Chebyshev distance, $m = 2$	43.34 %	84.04 %	**42.75 %**	80.58 %	38.35 %	80.10 %	**56.36 %**	52.67 %	**55.75 %**
Chebyshev distance, $m = 5$	45.25 %	67.77 %	28.32 %	48.54 %	39.81 %	69.42 %	46.84 %	50.16 %	40.23 %
Minkowski distance, $m = 2$, $p = 1.5$	41.23 %	**86.46 %**	39.86 %	81.07 %	40.29 %	83.01 %	54.66 %	54.97 %	53.86 %
Minkowski distance, $m = 5$, $p = 1.5$	**47.06 %**	62.50 %	29.61 %	46.60 %	33.98 %	70.87 %	46.83 %	44.02 %	41.77 %

clustering (Section IV-C2). The hyper-parameters for both methods were set according to [21]. Following, the value of the fuzzifier needs to be fixed; larger values of m result in more fuzzy clusters while lower values of m correspond to more crisp clusters. For this reason, the FCM configuration included both a high and a low value for the fuzzifier. Then, the distance metric has to be specified; all four distance metrics outlined in Sections IV-B1 and IV-B2 have been used (Euclidean, Manhattan, Chebyshev and Minkowski). Lastly, the validity index that would determine the optimal number of clusters should be decided upon; the Fukuyama-Sugeno, the Xie-Beni and the PCAES indices have been used for this task (Section V).

VII. RESULTS

The performance of the implemented algorithms is evaluated on two metrics; the *effective rate* and the *identification rate* [6]. The former is similar to the *Precision* metric of the Information Retrieval Theory while the latter is similar to the *Recall* metric of the same theory. More formally, let SI be the set of all sessions (for all users) identified by a given algorithm. Some of them match to existing sessions in the dataset, forming the set of the "real" sessions (RI), while others don't have a match, forming the set of "false" sessions FS. By definition $SI = RI \cup FS$. Finally, let TR be the set of all real sessions (for all users) existing in the dataset. Yet again, by definition, $RI \subseteq TR$.

The effective rate is defined to be the ratio of the "real" sessions identified over the overall sessions (identified)

$$ER = \frac{|RI|}{|SI|} \quad (\%) \tag{26}$$

while the identification rate is set to be the ratio of the "real" sessions identified over the overall sessions existing in the dataset

$$IR = \frac{|RI|}{|TR|} \quad (\%) \tag{27}$$

As with the precision and recall metrics of information theory, an algorithm is considered to be more efficient when it exhibits a better trade-off of both metrics defined beforehand. For this reason, they are combined together by calculating their harmonic mean, yielding to the F1 score

$$F1 = 2 \cdot \frac{ER \cdot IR}{ER + IR} \quad (\%) \tag{28}$$

Table III summarizes the results of the experiments of the algorithms discussed above, on the three validity indices outlined in the previous section. The worst outcome, in terms of the F1 score, is achieved by the time-heuristic algorithm; an indication that the baseline approach which considers a fixed amount of session duration is not always fit for every user.

The system which exhibits the best F1 score is the FCM algorithm, where the fuzzy partition matrix is initialized with the mountain clustering method. The distance metric used is the Minkowski distance with an exponent of $p = 1.5$; in the "middle" of the range between the Manhattan distance ($p = 1$) and the Euclidean distance ($p = 2$). Finally, the fuzzifier m is set to a low value.

An almost similar level of effectiveness is accomplished by some other configurations of the FCM algorithm as well; namely the *(Euclidean distance, m=2)* and *(Manhattan Distance, m=2)* when the fuzzy partition matrix is initialized with the mountain clustering method and the *(Chebyshev*

Distance, m=2), (Euclidean Distance, m=2), (Manhattan Distance, m=2), (Minkowski Distance, m=2) when the fuzzy partition matrix is initialized with the subtractive clustering method. What all those configurations have in common is a low fuzzifier value, resulting in a more crisp clustering of the session data. Indeed, higher fuzzifier values make the boundaries between the clusters less distinguishable, which in turn results in a greater overlap between successive sessions. It may also be deduced that the value of the fuzzifier (first term of the product of Eq. 6) influences the form of the objective function to a greater extend than the distance function (second term of the same product).

Another interesting observation is that the XB index (Eq. 21), when used in conjunction with the FCM algorithm for the determination of the optimal number of clusters, achieves by far the best results in the effective rate. This behaviour is attributed to the fact that this index penalizes more the compact representations than the other two. However, this comes at the cost of exhibiting a much lower identification rate; it clearly misses to uncover all the underlying sessions.

The opposite is true for the PCAES index (Eq. 23); it demonstrates a higher identification rate when compared to the other two, since it favours more compact representations at the largest possible distance between them. Yet, this tendency of trying to place clusters as far as possible from one another fails to properly model those sessions that occur in short (time) proximity and hence the lower effective rate.

VIII. Conclusion

In this paper, a new method for session identification in web log data has been proposed, based on the fuzzy c-means clustering algorithm. The novelty of the approach lies in the initialization of the fuzzy partition matrix using the subtractive clustering technique, the introduction in the FCM of more distance metrics (Manhattan, Chebyshev, Minkowski) as well as the use of candidate sessions in order to determine the real number of sessions in the underlying web log data. In order to accomplish this goal, an appropriate representation of the session data has been employed.

Preliminary results on a reference dataset justify the choices made, especially when compared to baseline approaches such as the time-heuristic methodologies. However, there is still room for improvement. A different mountain function could be tried, either to subtractive or to mountain clustering, that takes into account the temporal nature of the data at hand and the particularities of session identification. In addition, the FCM algorithm could be initialized with other methods such as Particle Swarm Optimization and Ant Colony Optimization and apart from these, other validity indices could be introduced in order to determine the quality of the clustering.

References

[1] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan, "Web usage mining: Discovery and applications of usage patterns from web data," *SIGKDD Explor. Newsl.*, vol. 1, no. 2, pp. 12–23, Jan. 2000. [Online]. Available: http://doi.acm.org/10.1145/846183.846188

[2] "The cost of ad blocking," PageFair and Adobe, Tech. Rep., 2015. [Online]. Available: http://downloads.pagefair.com/reports/2015_report-the_cost_of_ad_blocking.pdf

[3] L. D. Catledge and J. E. Pitkow, "Characterizing browsing strategies in the world-wide web," *Comput. Netw. ISDN Syst.*, vol. 27, no. 6, pp. 1065–1073, Apr. 1995. [Online]. Available: http://dx.doi.org/10.1016/0169-7552(95)00043-7

[4] N. Sharma and P. Makhija, "Web usage mining: A novel approach for web user session construction," *Global Journal of Computer Science and Technology: E Network, Web & Security*, vol. 15, no. 3, pp. 15–18, 2015. [Online]. Available: https://globaljournals.org/GJCST_Volume15/3-Web-usage-Mining.pdf

[5] M. Srivastava, R. Garg, and P. K. Mishra, "Preprocessing techniques in web usage mining: A survey," *International Journal of Computer Applications*, vol. 97, no. 18, pp. 1–9, July 2014, full text available.

[6] H. Xinhua and W. Qiong, "Dynamic timeout-based a session identification algorithm," in *Electric Information and Control Engineering (ICEICE), 2011 International Conference on*, April 2011, pp. 346–349.

[7] V. Chitraa and A. S. Davamani, "A survey on preprocessing methods for web usage data," *CoRR*, vol. abs/1004.1257, 2010. [Online]. Available: http://arxiv.org/abs/1004.1257

[8] M. S. Kamat, W. Dr. Bakal, J, and M. Nashipudi, "Improved data preparation technique in web usage mining," *International Journal of Computer Networks and Communications Security*, vol. 1, no. 7, pp. 284–291, Dec. 2013. [Online]. Available: http://www.ijcncs.org/published/volume1/issue7/p1_1-7.pdf

[9] Z. Ansari, S. A. Sattar, A. V. Babu, and M. F. Azeem, "Mountain density-based fuzzy approach for discovering web usage clusters from web log data," *Fuzzy Sets Syst.*, vol. 279, no. C, pp. 40–63, Nov. 2015. [Online]. Available: http://dx.doi.org/10.1016/j.fss.2015.01.021

[10] W3C, "Logging control in w3c httpd." [Online]. Available: https://www.w3.org/Daemon/User/Config/Logging.html\#common-logfile-format

[11] R. Iváncsy and S. Juhász, "Analysis of web user identification methods," *World Academy of Science, Engineering and Technology*, vol. 2, no. 3, pp. 212–219, 2007.

[12] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 1981.

[13] H. C. Huang, Y. Y. Chuang, and C. S. Chen, "Multiple kernel fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 120–134, Feb 2012.

[14] R. R. Yager and D. P. Filev, "Generation of fuzzy rules by mountain clustering," *J. Intell. Fuzzy Syst.*, vol. 2, no. 3, pp. 209–219, May 1994. [Online]. Available: http://dl.acm.org/citation.cfm?id=2656634.2656635

[15] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *J. Intell. Fuzzy Syst.*, vol. 2, no. 3, pp. 267–278, May 1994. [Online]. Available: http://dl.acm.org/citation.cfm?id=2656634.2656640

[16] W. Wang and Y. Zhang, "On fuzzy cluster validity indices," *Fuzzy Sets and Systems*, vol. 158, no. 19, pp. 2095 – 2117, 2007, theme: Data Analysis. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165011407001133

[17] Y. Fukuyama and M. Sugeno, "A new method of choosing the number of clusters for the fuzzy c-means method," in *Proceedings of the Fifth Fuzzy Systems Symposium*, 1989, pp. 247–250.

[18] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, Aug 1991.

[19] K.-L. Wu and M.-S. Yang, "A cluster validity index for fuzzy clustering," *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1275 – 1291, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865504003629

[20] "Online community of the students of the school of electrical and computer engineering - http://shmmy.ntua.gr/." [Online]. Available: http://shmmy.ntua.gr/

[21] K. Hammouda and F. Karray, "A comparative study of data clustering techniques."