Received 23 September 2008
Revised 21 May 2009
Accepted 22 May 2009

A new approach to web users clustering and validation: a divergence-based scheme

Vassiliki A. Koutsonikola, Sophia G. Petridou,
Athena I. Vakali and Georgios I. Papadimitriou
*Department of Informatics, Aristotle University of Thessaloniki,
Thessaloniki, Greece*

Abstract

Purpose – Web users' clustering is an important mining task since it contributes in identifying usage patterns, a beneficial task for a wide range of applications that rely on the web. The purpose of this paper is to examine the usage of Kullback-Leibler (KL) divergence, an information theoretic distance, as an alternative option for measuring distances in web users clustering.

Design/methodology/approach – KL-divergence is compared with other well-known distance measures and clustering results are evaluated using a criterion function, validity indices, and graphical representations. Furthermore, the impact of noise (i.e. occasional or mistaken page visits) is evaluated, since it is imperative to assess whether a clustering process exhibits tolerance in noisy environments such as the web.

Findings – The proposed KL clustering approach is of similar performance when compared with other distance measures under both synthetic and real data workloads. Moreover, imposing extra noise on real data, the approach shows minimum deterioration among most of the other conventional distance measures.

Practical implications – The experimental results show that a probabilistic measure such as KL-divergence has proven to be quite efficient in noisy environments and thus constitute a good alternative, the web users clustering problem.

Originality/value – This work is inspired by the usage of divergence in clustering of biological data and it is introduced by the authors in the area of web clustering. According to the experimental results presented in this paper, KL-divergence can be considered as a good alternative for measuring distances in noisy environments such as the web.

Keywords Internet, User studies, Data mining, Cluster analysis

Paper type Research paper

Introduction

Nowadays the world wide web is a popular and interactive medium for information publishing and retrieval. However, its huge growth has led to an information overload that continuously expands causing various problems to web users. Usually, these problems are related to the detection of relevant information since a remarkable percentage of the returned search results are characterized by low precision or irrelevance (Srivastava *et al.*, 2000). In addition, there is often hidden information behind raw data that must be revealed in order to obtain valuable knowledge that could be quite useful in upper level applications such as e-commerce and e-learning.

Web usage mining involves the discovery of user access patterns from web usage data which is usually generated by web servers and collected in server access logs

International Journal of Web Information Systems
Vol. 5 No. 3, 2009
pp. 348-371
© Emerald Group Publishing Limited
1744-0084
DOI 10.1108/17440080910983583

(Srivastava *et al.*, 2000). As more organizations rely on the web to conduct business, collecting information about users' behavior and extracting their usage patterns can be quite important for dynamic content web sites. Customization and personalization of these sites are based on the similar access patterns of the users belonging to the clusters formed by web usage mining techniques. In addition, web usage mining can result in better structuring and management of a web site making it more effective, it enhances the operation of the overall network system as well as it improves information retrieval and content delivery on the web. What is more, it can aid in caching and prediction of future page references and particularly improve effectiveness of specific applications such as e-commerce. Finally, for organizations that deal with advertising on the web, analyzing user access patterns helps in targeting advertisements to specific groups of users.

Research background review

Clustering, in the web domain, is a technique of extracting knowledge about the visitors of a web site and exhibits great practical importance. For example, the task of providing personalized web content can considerably contribute to a site's popularity. This task can make use of the results of a clustering process which creates groups of users according to their access behavior. Furthermore, the implementation of a search engine can be based on clustering results in order to become more effective. In general, clustering is defined as the problem of creating groups of items (i.e. clusters) which are "similar" between them and "dissimilar" to the items belonging to other clusters. Web clustering can involve either users or pages. The purpose of users' clustering is to establish groups of users that present similar browsing patterns while page clustering discovers groups of pages having related content.

Web clustering is a well-studied problem and numerous clustering algorithms appear in literature which can be broadly categorized into different categories depending on the criteria employed. In a general categorization scheme, clustering algorithms are divided into partitional and hierarchical, according to whether they produce flat partitions or a hierarchy of clusters (Jain *et al.*, 1999; Xu and Wunsch, 2005). Some of the most commonly used hierarchical algorithms are single, complete, and average links (Fung, 2001) while a well-known partitional clustering algorithms is *K*-means (McQueen, 1967). Moreover, clustering algorithms may differentiate in terms of the nature of the grouping they perform and which may concern a hard assignment, i.e. data are divided into distinct clusters, where each data element belongs to exactly one cluster, or a fuzzy one, i.e. data elements are assigned to one or more clusters with different membership levels (Xu and Wunsch, 2005). Furthermore, a clustering approach may be based on a distance function to identify the objects that should be clustered together (similarity based) or to other probabilistic techniques (model-based) (Vakali *et al.*, 2006). An overview of the most popular web users or web pages clustering methodologies is presented in Table I, while future trends are discussed in Vakali *et al.* (2004).

K-means is a widely used partitional clustering algorithm since it has proved to be efficient in applications involving large datasets where the use of hierarchical approaches is computationally prohibitive (Jain *et al.*, 1999; Wijaya and Bressan, 2006). More specifically, it classifies a given set of data points to a certain number of clusters (e.g. *k* clusters) based on the notion of their similarity. This similarity based clustering

Table I.
An overview of clustering approaches

Approach	Cluster content	Clustering algorithm	Methodology
Cadez *et al.* (2002)	Web users	Partitional hard	Model based
Petridou *et al.* (2008)	Web users	Partitional hard	Similarity based
Shokry *et al.* (2006)	Web pages	Partitional fuzzy	Similarity based
Mojica *et al.* (2005)	Web pages	Hierarchical hard	Similarity based
Yang and Padmanabhan (2005)	Web users	Hierarchical hard	Similarity based
Lazzerini *et al.* (2003)	Web users	Hierarchical fuzzy	Similarity based
Castellano *et al.* (2007)	Web users	Partitional fuzzy	Similarity based
Zeng *et al.* (2002)	Web users/pages	Hard	Model based
Liu *et al.* (2005)	Web users/pages	Fuzzy	Similarity based

algorithm is characterized by the proximity measure that quantifies how "similar" two data points are. *K*-means attempts to minimize a criterion function value which measures the distance of each point from the centre of the cluster to which the point is assigned.

Once the clustering process is completed, determining the quality of the clustering results is a challenging task, since the "similarity" measure is tailored for the underlying clustering process and no standard unified criterion exists. Therefore, cluster validation is a very important issue in clustering analysis because only validated clustering results may be appreciated and exploited in applications. In this context, various clustering validity indices have been proposed to measure the quality of clustering results (Halkidi *et al.*, 2002a, b).

But, whatever clustering algorithm or distance measure is chosen, it is imperative to assess whether the results are susceptible to noise (Kerr and Churchill, 2001). In the web environment, "noise" refers to visits which are executed by chance or by mistake, e.g. when paying a visit to an originally "promising" page which tends out to be irrelevant to our interest or mistakenly pressing a hyperlink. In general, it would be desirable that a user clustering process should not be misguided or considerably affected by such random events.

Motivation and contribution

This work is inspired by the usage of divergence in clustering of biological data (Kasturi *et al.*, 2003). The physical significance of the divergence of a vector is the rate at which "density" exits a given region of space. The notion of divergence motivated the development of data mining approaches to discover patterns of gene expression from array-derived gene expression data. Such a divergence approach for analyzing biological data led to superior patterns compared to those that a hierarchical clustering algorithm produced using the Pearson correlation distance measure (Kasturi *et al.*, 2003).

Although there is a number of common features that characterize both web and biological data (e.g. huge exploration spaces, dynamic data nature, and data representation), the divergence has not been explored for web data analysis applications. Therefore, in this paper, we exploit the Kullback-Leibler (KL) divergence, an information theoretic distance, as a way to measure distances between web users. Considering that KL-divergence measures the difference between two probability distributions and not their distance as this calculated by conventional distance measures such as Euclidean or Manhattan distances (hence "divergence" rather than "distance"), it is expected to be robust against noise. In particular, in the case of the web, which is characterized by large-scale and complex data that dynamically changes over time, accurate and fast responses are needed, so it is important to assess the impact of noise in a web mining process.

The idea of KL-divergence has originally been introduced by the authors in Petridou *et al.* (2006), but here it is thoroughly investigated since the theoretical background is setup and the divergence-oriented scheme is experimented in comparison to a variety of other distance measures. More specifically, we assess the results of the KL-divergence-oriented web clustering and compare it with the widely used approaches based on distance measures such as Euclidean, Standardized Euclidean (S-Euclidean), Manhattan, and Chebychev. Our experimentation involves different datasets with a varying percentage of noise, namely synthetic data characterized by the absence of noise, real data which are noisy because of their nature as well as real data on which we imposed extra noise. The clustering evaluation process indicated that the divergence-based scheme exhibits appreciable tolerance in the presence of noise in comparison with the other distance measures.

The remainder of this paper is organized as follows: first, the typical distance measures used in clustering approaches are discussed and the role of the KL-divergence is described. Then, our KL-divergence clustering approach is presented and the description of the evaluation process follows. Next, our implementation and experimental results are provided along with a discussion about the influence of noise in the clustering approach under the various distance measures. Finally, our conclusions and future work are presented.

Distance measures

Notation

We consider a particular web usage framework where we have (as a source) server log files which capture the users' navigational behavior. Then, we define the following basic terms and notation used throughout this paper:

- A user set is denoted as $U = \{u_1, \ldots, u_n\}$, where u_i, $i = 1, \ldots, n$, represents the ith of the n users.
- A user's pattern (or user's vector or user's observation) $\mathbf{X}(i, :)$, $i = 1, \ldots, n$, represents the accessing behavior of the user u_i. More specifically, it is a multivariate vector consisting of m measurements:

$$\mathbf{X}(i, :) = (X(i, 1), \ \ldots, X(i, m)),$$

  where the $X(i, j)$ element, $j = 1, \ldots, m$, indicates the number of times the user u_i visits the page j. All the $\mathbf{X}(i, :)$ vectors are organized in the two dimensional $n \times m$ users' pattern table $\mathbf{X}$.

Example 1. Consider the vector $\mathbf{X}(3, :) = (22, 11, 0, 54, 0)$, where $m = 5$. Then, the user identified as u_3 has 22, 11, and 54 visits to pages identified as 1, 2, and 4, respectively, but no visits to pages 3 and 5:

- The probability distribution $\mathbf{P}(i, :)$ of the user u_i is a vector of m values produced by the normalization of its $\mathbf{X}(i, :)$ pattern, i.e. $\mathbf{P}(i,:) = \mathbf{X}(i,:)/\sum_{j=1}^{m} X(i,j)$. Thus:
$$\mathbf{P}(i,:) = (P(i,1), \ldots, P(i,m)),$$
where the $P(i, j)$ element, $j = 1, \ldots, m$, indicates the probability with which the user u_i visits the page j. All the $\mathbf{P}(i, :)$ vectors are organized in the two dimensional $n \times m$ distribution (or normalized) table.

Example 2. Considering the above u_3 user, its probability distribution vector results from $\mathbf{P}(3,:) = \mathbf{X}(3,:)/\sum_{j=1}^{5} X(3,j)$ and thus $\mathbf{P}(3, :) = (0.25, 0.13, 0, 0.62, 0)$. Therefore, the user u_3 visits pages identified as 1, 2, and 4 with probabilities 0.25, 0.13, and 0.62, respectively, whereas the probability to visit pages 3 and 5 is 0:

- A distance measure d is a metric (or quasi metric) used to quantify the similarity of users' patterns (or their probability distributions). Since some clustering algorithms work on a table of distance values instead of the pattern table, i.e. $\mathbf{X}$ or distribution table, i.e. $\mathbf{P}$, it is useful to precompute all the $n(n-1)/2$ pairwise distance values for the n users' probability distributions and store them in an $n \times n$ (symmetric) distance table. The normalized value of d will be denoted by d^* and will be stored in an $n \times n$ (symmetric) normalized distance table which is denoted as $\mathbf{D}$.

Notation summary is given in Table II.

Distance measures and the role of divergence

Since similarity is fundamental to the definition of a cluster, a measure of the similarity between two user patterns is essential to most clustering approaches. In practice, it is most common to calculate the dissimilarity between two patterns using a distance measure. However, because of the variety of distance measures, their choice must be done carefully.

Different measures are appropriate to capture dissimilarities between patterns according to their representation (Jain *et al.*, 1999). For example, patterns can be represented using string or tree structures. Several measures of similarity between strings are described in Baeza-Yates and Frakes (1992) while a good summary of similarity measures between trees is given in Zhang (1995). However, patterns are typically represented as vectors whose values can be either quantitative (continuous values, e.g. weight; discrete values, e.g. the number of visits of a web user; or interval values, e.g. the duration of an event) or qualitative (nominal, e.g. "red" or ordinal, e.g. "cool"). Among the most popular distance measures used for calculating the

Table II.
Basic symbols notation

Symbol	Description
n	Number of users
m	Number of pages
U	Users' set $U = \{u_1, \ldots, u_n\}$
$\mathbf{X}$	$n \times m$ users' pattern table
$\mathbf{P}$	$n \times m$ distribution table
d	Distance measure
$\mathbf{D}$	$n \times n$ distance table

dissimilarity between vectors of either continuous or discrete values are the Euclidean, S-Euclidean, Manhattan, and Chebychev distances (Sturn, 2001). Thus, since our users' patterns are represented as vectors with discrete values and the probability distributions of these discrete values are also represented as vectors, we will focus on these well-known distance measures.

Let us suppose that $\mathbf{P}(x, :)$, $\mathbf{P}(y, :)$, and $\mathbf{P}(z, :)$, where $x, y, z = 1, \ldots, n$ are the probability distributions of the web users u_x, u_y, and u_z. The distance measure between two of them $d(u_x, u_y)$ gives a numerical value to the amount of their dissimilarity and is required to satisfy the first three of the following five properties of a metric:

(1) Non-negativity $d(u_x, u_y) \geq 0$.

(2) Symmetry $d(u_{\mathrm{x}}, u_{\mathrm{y}}) = d(u_{\mathrm{y}}, u_{\mathrm{x}})$.

(3) Identification mark $d(u_x, u_x) = 0$.

(4) Definiteness $d(u_x, u_y) = 0$ iff $\mathbf{P}(x, :) = \mathbf{P}(y, :)$.

(5) Triangle inequality $d(u_x, u_y) \leq d(u_x, u_z) + d(u_z, u_y)$.

Euclidean, S-Euclidean, Manhattan, and Chebychev distances satisfy all five properties of a metric and their formulas are given in Table III.

In the formula of S-Euclidean distance between the probability distributions $\mathbf{P}(x, :)$ and $\mathbf{P}(y, :)$, the $\boldsymbol{\sigma}_j^2$ is the variance of the column j, where $j = 1, \ldots, m$, and it is defined as $\sigma_j^2 = \sum_{i=1}^{n}(P(i,j) - \mu_j)^2$, where $\mu_j = (1/n)\sum_{i=1}^{n} P(i,j)$. The Chebychev distance between the vectors $\mathbf{P}(x, :)$ and $\mathbf{P}(y, :)$ is the maximum difference among their corresponding elements.

Example 3. Let us consider the following distribution table $\mathbf{P}$:

$$\mathbf{P} = \begin{pmatrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0.4 & 0.5 \\ 0.2 & 0.7 & 0.1 \\ 0.6 & 0.2 & 0.2 \end{pmatrix}.$$

From the above table $\mathbf{P}$, it holds that the probability distributions of u_1 and u_3 are $\mathbf{P}(1, :) = (0.3, 0.5, 0.2)$, $\mathbf{P}(3, :) = (0.2, 0.7, 0.1)$, respectively. The variances of the three columns are $\boldsymbol{\sigma}^2 = (0.047, 0.043, 0.030)$ while their mean values are $\mu = (0.3, 0.45, 0.25)$. Thus, the Euclidean distance of u_1 and u_3 is $d_{\mathrm{E}}(u_1, u_3) = 0.24$, their S-Euclidean distance is $d_{\mathrm{SE}}(u_1, u_3) = 1.21$, their Manhattan distance is $d_{\mathrm{Man}}(u_1, u_3) = 0.4$ while their Chebychev distance is $d_{\mathrm{Ch}}(u_1, u_3) = 0.2$.

Beyond the typical distance measures (Table III), here we use the relative entropy or KL-divergence which originates from information theory and is of probabilistic nature.

Distance	Formula
Euclidean	$d_{\mathrm{E}}(u_x, u_y) = \sqrt{\sum_{j=1}^{m} \lvert P(x,j) - P(y,j) \rvert^2}$
S-Euclidean	$d_{\mathrm{SE}}(u_x, u_y) = \sqrt{\sum_{j=1}^{m} \frac{1}{\sigma_j^2}(P(x,j) - P(y,j))^2}$
Manhattan	$d_{\mathrm{Man}}(u_x, u_y) = \sum_{j=1}^{m} \lvert P(x,j) - P(y,j) \rvert$
Chebychev	$d_{\mathrm{Ch}}(u_x, u_y) = \max_{j=1}^{m} \lvert P(x,j) - P(y,j) \rvert$

Table III. Formulas of Euclidean, S-Euclidean, Manhattan, and Chebychev distance measures

Given the probability distributions **P**(x, :) and **P**(y, :), where $x, y = 1, \ldots, n$, of the users u_x, u_y, the KL-divergence is a quantity which measures the difference between **P**(x, :) and **P**(y, :) and is defined as follows (Dhillon *et al.*, 2002):

$$d_{\mathrm{KL}}(u_x, u_y) = \sum_{j=1}^{m} P(x, j) \log \frac{P(x,j)}{P(y,j)}. \tag{1}$$

Based on the above definition, it is obvious that the KL-divergence is a measure of the "distance" between two probability distributions without being a typical distance measure. This is due to the fact that it is not symmetric and does not obey the triangle inequality[1] (Cover and Thomas, 1991). However, it should be noted that Kullback and Leibler themselves actually defined the KL-divergence between u_x and u_y as:

$$d_{\mathrm{KL}}(u_x, u_y) = d_{\mathrm{KL}}(u_x, u_y) + d_{\mathrm{KL}}(u_y, u_x), \tag{2}$$

which is symmetric. The property of symmetry is fundamental in order that the comparison between the KL-divergence and the other distance measures is meaningful. In our framework, we use the KL-divergence as defined by equation (2).

Divergence-oriented clustering

Problem formulation

In the proposed clustering it is important to identify the type of problem to be solved. We consider that under the proposed KL-divergence-oriented clustering process, k denotes the number of clusters while U is the set of users $U = (u_1, \ldots, u_n)$ to be clustered. Then, $C_1, \ldots, C_k$ denote each of the k clusters consisting of $|C_1|, \ldots, |C_k|$ members, respectively. Under this notation, the underlying clustering process CL is defined as the assignment of n users to k users' groups (i.e. clusters):

$$\mathrm{CL} : \{1, \ldots, n\} \rightarrow \{1, \ldots, k\},$$

such that the users assigned to each cluster are more similar to each other than to the users assigned to different clusters, based on their page preferences as these were logged in the server files. The membership of a user u_i, where $i = 1, \ldots, n$, to a cluster C_j, where $j = 1, \ldots, k$, is defined by the function f as follows:

$$f(u_i, C_j) = \begin{cases} 1 & \text{if } u_i \in C_j \\ 0, & \text{otherwise} \end{cases}$$

Let us consider an arbitrary cluster C_j, where $j = 1, \ldots, k$, of the users' set U. When a clustering process CL is applied to U, the cluster C_j is represented by a single point. We call this point cluster's centre and we denote it as c_j. In practice, c_j is one of the users u_i belonging to the cluster C_j and more specifically the user that minimizes the sum of distances between all users belonging to C_j. We denote PC(j, :) to be the probability distribution vector of c_j. Then, given that both **P**(i, :), where $i = 1, \ldots, n$, and PC(j, :), where $j = 1, \ldots, k$, are vectors, their dissimilarity can be measured by their KL-divergence, i.e. $d_{\mathrm{KL}}(u_i, c_j)$. Considering all clusters, we define the criterion function $J_{\mathrm{KL}}(U)$ to be the sum of the divergences over pages between each user and the centre of the cluster that the user is assigned to:

$$J_{\mathrm{KL}}(U) = \sum_{j=1}^{k} \sum_{f(u_i, C_j)=1} d_{\mathrm{KL}}(u_i, c_j). \tag{3}$$

Based on the above, we define the KL-divergence clustering problem as follows.

Problem 1. (KL-divergence clustering). Given a set U of n users organized in an $n \times m$ table, an integer value k, and the criterion function $J_{\mathrm{KL}}(U)$, find a CL clustering of U into k clusters such that the $J_{\mathrm{KL}}(U)$ is minimized.

The KL-divergence clustering approach

Our clustering problem, as most of the clustering problems, is NP-hard (Garey and Johnson, 1979) and, therefore, any polynomial algorithm would provide an approximate solution. Moreover, this approximation solution is not bounded-error since the KL-divergence does not satisfy the triangle inequality (Charikar *et al.*, 1999). Thus, the KL-divergence clustering approach adopts local search heuristics since it employs the well-known K-means algorithm to find the clustering solution (McQueen, 1967).

K-means is an unsupervised, hard partitional clustering algorithm which classifies a given dataset to a certain number of k clusters, fixed a priori, and although it does not guarantee convergence to a global optimum, it has proved very effective in practice. Specifically, its time complexity is $O(nkr)$, where n is the number of users in the dataset, k the number of clusters to be created, and r the number of iterations that takes the algorithm to converge. However, both k and r are relatively small compared to the number of users n, and thus they do not contribute to the algorithm's complexity (Jain *et al.*, 1999). So, the clustering is computed in time linear on the number of users: $O(n)$.

Clustering phases

The KL-divergence clustering algorithm is an unsupervised, hard partitional method used to optimize the criterion function $J_{\mathrm{KL}}(U)$, defined by equation (3). It is a two-step process, depicted in Figure 1, where the data is first preprocessed and normalized in order that we compute the pairwise distances, and then is classified by using the K-means algorithm in conjunction with the KL-divergence as the dissimilarity measure. Once the CL clustering process is completed, clustering evaluation takes place to assess the quality of our resulted clusters.

Data preprocessing and normalization. In accordance with our data source notation given in a previous subsection, we preprocessed users logs in order to exclude non-important information (e.g. requests from search engines' agents, viruses generated hits or log entries about images, css, swf files, etc.). In the obtained $n \times m$ pattern table $\mathbf{X}$, each row represents a user's pattern $\mathbf{X}(i, :)$, where $i = 1, \ldots, n$, and each column corresponds to each of the m pages which are recorded as visited by the n users in a web server's log file (multiple users access a single site). Therefore, each cell of this table indicates the number of times each user visits each page.

The $n \times m$ pattern table is then normalized in order that we form the $n \times m$ distribution (normalized) table, where each row is the probability distribution $\mathbf{P}(i, :)$, where $i = 1, \ldots, n$, of the user pattern $\mathbf{X}(i, :)$. Therefore, its elements express the probability with which each user visits each page. The normalized expression

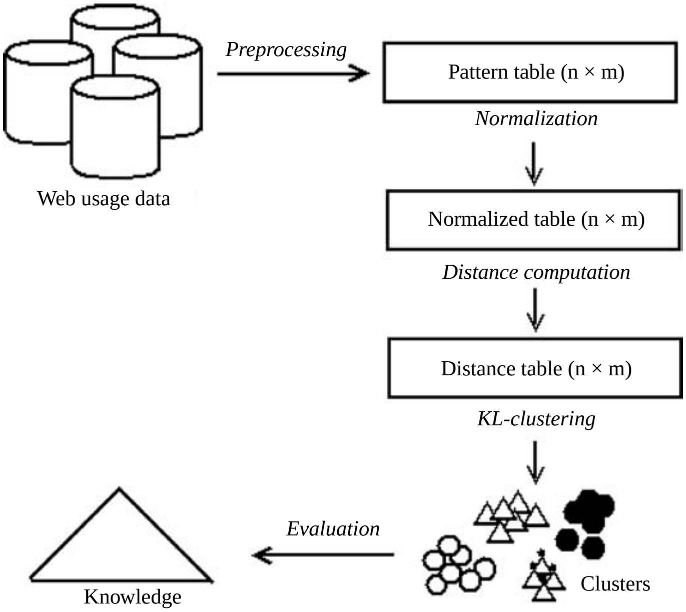


Figure 1.
The KL-divergence clustering process overview

values for each user fall in the interval [0, 1] and each row sum is 1 (unit total probability mass).

Each row of this $n \times m$ normalized table is suitable for the calculation of the $n(n-1)/2$ pairwise distance values between the n users' probability distribution vectors using the KL-divergence. After this calculation, we receive the $n \times n$ distance table $\mathbf{D}$ which is symmetric, since the divergence between $\mathbf{P}(x, :)$ and $\mathbf{P}(y, :)$, where x, $y = 1, \ldots, n$, is defined as $\mathrm{KL}(u_x, u_y) + \mathrm{KL}(u_y, u_x)$ which is symmetric (Dhillon *et al.*, 2002). The $n \times n$ distance table $\mathbf{D}$ as well as the number of k clusters to be created will be the input to the clustering algorithm.

KL-clustering. For the clustering process, we adopted the K-means algorithm to produce the k clusters. K-means begins by initializing a set of k centres c_j, where $j = 1, \ldots, k$ one for each cluster C_j, randomly selected from users' set U. We devise an initial clustering assignment, by including each of the left $n - k$ points to the closest cluster, i.e. to the one with the minimum distance value d between the cluster centre and the examined point (where d is the chosen distance measure). Then based on this initial assignment, we re-compute k new centres and we proceed to an iterative procedure where points are assigned to clusters and k centres are re-computed, until no changes in the location of objects exists. The k new centres in each iteration are points from the users' set U. In particular, the algorithm selects c_j among the points that are assigned to cluster C_j to be the point that minimizes the sum of distances within cluster C_j (i.e. distance between the cluster centre c_j and its members). In our KL-clustering the K-means algorithm performs iterations based on the table of distance values which is described in data preprocessing and normalization step.

The proposed approach is incremental in nature, since it assigns a user to a cluster only when it is determined that such an assignment will lead to an improvement in the value of the criterion function. As mentioned earlier, since each assignment optimizes the criterion function the algorithm may converge to a local minimum depending on the particular cluster centres selected. Here, to eliminate this sensitivity, the clustering phase is repeated a number of times, i.e. we compute NUM (in our experimentation NUM = 100) different clustering solutions and the one that shows the best value for the criterion function is kept. For the rest of this paper, when we refer to the clustering solution we will mean the solution shown to be the best out of these NUM potential different solutions:

Algorithm. The KL-divergence clustering algorithm.

(1) *Input.* A set U of n users organized in an $n \times m$ pattern table $\mathbf{X}$ and the number of clusters k.

(2) *Output.* criterion function $J_{\mathrm{KL}}(U)$ and assignment of the n users into the k clusters that minimizes the $J_{\mathrm{KL}}(U)$.

(3) Randomly select k points from the users' set U as the initial clusters' centres: $c_1, \ldots, c_k$.

(4) Calculate the KL-divergence $d_{\mathrm{KL}}(u_i, c_j)$ between the probability distribution $\mathbf{P}(i, :)$, $i = 1, \ldots, n$, of each user u_i and the probability distribution PC(j, :), $j = 1, \ldots, k$, of each cluster centre c_j and store them in $n \times n$ distance table $\mathbf{D}$.

(5) Assign all points to the cluster that has the closest centre.

(6) Re-compute the centre of each cluster.

(7) Repeat steps 2 and 3 until the centres do not change or when the criterion function improvement between two consecutive iterations is less than a minimum amount of improvement specified.

Distances' scaling

Our KL-divergence clustering algorithm aims to minimize the $J_{\mathrm{KL}}(U)$ criterion function, defined in equation (3). Most of the earlier approaches that use Euclidean, S-Euclidean, Manhattan, and Chebychev distances along with the K-means algorithm, define a similar $J(U)$ criterion function to be minimized:

$$J(U) = \sum_{j=1}^{k} \sum_{f(u_i, C_j)=1} d(u_i, c_j). \tag{4}$$

However, in equation (4) instead of d_{KL} we have d to be the chosen distance (Euclidean, S-Euclidean, Manhattan, or Chebychev), as defined in Table IV, between the probability distribution $\mathbf{P}(i, :)$ of user u_i which belongs to cluster C_j and the probability distribution PC(j, :) of the cluster's centre c_j.

However, comparing criterion functions based on different distance measures could be considered meaningless since each distance measure changes in different scale (i.e. the Euclidean distance becomes zero if two user patterns is identical and $\sqrt{2}$ if there is nothing in common between them while the corresponding values for the Manhattan distance is 0 and 2). One way to overcome this difficulty is to normalize the distance values. In this case, the criterion function $J(U)$ is defined as:

$$J(U) = \sum_{j=1}^{k} \sum_{f(u_i,C_j)=1} d^*(u_i, c_j), \tag{5}$$

where d^* is the normalized value of d.

Let us suppose that $d(u_x,u_y)$ is the distance between the probability distributions $\mathbf{P}(x, :)$ and $\mathbf{P}(y, :)$ of users u_x and u_y, where $x, y = 1, \ldots, n$, and max d is the maximum distance for all possible $\mathbf{P}(x, :)$, $\mathbf{P}(y, :)$ probability distribution vectors. This assumption as well as the following equations are similar for every distance chosen, i.e. considering that $d = d_{\mathrm{KL}}$ then the max d is the maximum KL-divergence between all possible $\mathbf{P}(x, :)$ and $\mathbf{P}(y, :)$ whereas if $d = d_{\mathrm{E}}$ then the max d is the maximum Euclidean distance, etc. Then:

$$0 \leq d(u_x, u_y) \leq \max d. \tag{6}$$

Defining the normalized distance $d^*(u_x, u_y)$ to be:

$$d^*(u_x, u_y) = \frac{d(u_x, u_y)}{\max d}, \tag{7}$$

then, from equations (6) and (7) we conclude that:

$$0 \leq d^*(u_x, u_y) \leq 1. \tag{8}$$

After this normalization, the $n \times n$ distance table D is both symmetric and normalized since its values are in a range of 0-1. This table will form the input to the K-means algorithm with the k clusters to be created.

From the definition of the criterion function $J_{\mathrm{KL}}(U)$ (equation (3)) and $J(U)$ (equation (4)) we also conclude that:

$$0 \leq J_{\mathrm{KL}}(U), J(U) \leq n. \tag{9}$$

Clustering validation

Clustering results evaluation is necessary due to the fact that there is a wide range of clustering algorithms which end up to different clusters, so it is not feasible to compare them in absolute terms. A widely used cluster evaluation approach is based on the visualization of the dataset which can be quite revealing in verifying clustering results. However, graphical analysis is not easily applicable in case of large multidimensional datasets. Moreover, entropy is a popular method for evaluating clustering results

Table IV. Clustering and criterion function notation

Symbols	Description
CL	Clustering process
k	Number of clusters
C_j	Cluster, $j = 1, \ldots, k$
c_j	Cluster centre, $j = 1, \ldots, k$
CP(j, :)	Probability distribution vector of c_j, $j = 1, \ldots, k$
$f(u_i, C_j)$	Function membership of user u_i to cluster C_j
$J_{\mathrm{KL}}(U)$	Criterion function

(Zhao and Karypis, 2005), but it is restricted to supervised clustering approaches. An alternative way to evaluate and assess the results of a clustering process of multidimensional data is the cluster validation (Halkidi *et al.*, 2002a, b). Cluster validation aims at the quantitative evaluation of clustering results and is based on certain validity indices characterized as (Halkidi *et al.*, 2002a, b; Stein *et al.*, 2003):

- *Internal validity indices*: their goal is to evaluate the results of a clustering algorithm using only quantities that involve the data themselves. Such internal measures base their calculations solely on the clustering that has to be evaluated and thus they are mostly used in unsupervised clustering.
- *External validity indices*: they perform clustering validation with reference to external knowledge, such as a pre-specified structure which reflects an intuition about the clustering structure of a dataset. It should be noted that external measures are not applicable in real world situations where unsupervised clustering approaches are applied, since reference classifications are usually not available. Overall, the main drawback of the external validity indices as compared to internal ones is their computational cost, considering that they measure the degree to which the data clusters conform to an a priori specified scheme.
- *Relative validity indices*: they aim at revealing the best clustering scheme that a clustering algorithm can define under certain assumptions and parameters. Namely, the basis of the relative validity indices is the evaluation of a clustering structure by comparing it to other clustering schemes, resulting from the same algorithm but with different parameter values. When applied appropriately, internal indices can serve as relative (Jain *et al.*, 1999), e.g. looking for the optimal number of clusters, we evaluate the clustering results of the same algorithm that takes the number of clusters as a parameter.

In the case of our synthetic datasets experimentation, the underlying structure of the data is known prior to clustering and thus the application of external indices in clusters' validation is feasible. However, our real data workload experimentation implies no external knowledge of a pre-specified classification, so we will have to perform clustering validation based solely on quantities that involve the data themselves. Based on the above remarks, internal validity indices are the most typical choice in this case.

According to the Table V validity indices notation, Table VI presents the Davies-Bouldin (DB) index (Davies and Bouldin, 1979) and Dunn's index (Dunn, 1974), which are two of the most commonly used internal validity indices. Given their definitions, DB index tries to minimize the within-cluster scatter maximizing at the same time the between-cluster separation while Dunn's index measures the ratio between the smallest cluster distance and the largest intra-cluster distance in a partitioning. Moreover, DB index considers the average distance of all elements in a cluster to their cluster centre (cluster's diameter) and the distance between cluster centres and thus it is supposed to be more robust than Dunn's index (Boutin and Hascoer, 2004). On the other hand, Dunn's index is very instable when in presence of outliers since it considers only two distances, namely the minimum of the distances between the two closest points belonging to different clusters and the maximum of the distances between the two most remote points in each cluster (Gunter and Bunke, 2003).

By the definition of DB index in Table VI, for each cluster C_i, a cluster C_j is chosen such that the specified quotient is maximized, meaning that dist(C_i, C_j), as defined in

Table V, must be the minimum. Thus, the DB index measures the average of similarity between each cluster and its most similar one. Obviously, it is desirable for the clusters to have the minimum possible similarity to each other, therefore we seek clusterings that minimize DB index values (Gunter and Bunke, 2003). According to the definition of Dunn's index (Table VI), if a dataset contains well-separated clusters, the distances among the clusters, expressed by $d_{\min}$, are usually large and the distances inside the clusters, expressed by $d_{\max}$, are expected to be small. As a consequence, a large value of Dunn's index means better cluster configuration (Gunter and Bunke, 2003).

Experimentation

To evaluate our KL clustering approach we carried out experimentation that involves both synthetic and real datasets. We performed experimentation on synthetic data, in order to check whether our proposed method actually "understands" and captures the underlying users' behavior model, which was originally used to generate the synthetic data. In case of real datasets, users' navigational behavior is recorded in web servers' log files. The size of a log file can grow very large, containing at the same time many useless for the clustering approach information. Thus, as it has already been discussed, data preprocessing preceded in order to obtain meaningful, for the clustering, data. Next, we corrupted our noisy data with some extra noise to study our approach's tolerance in noise increase, through the evaluation process that followed clustering. The overall procedure is depicted in Figure 2.

Regarding the clustering evaluation under the synthetic datasets experimentation, we initially assessed the criterion functions values defined in equations (3) and (4) using large-scale multidimensional datasets. In the case of our real datasets, we employed the criterion functions values as well as the DB index defined in Table VI. Moreover, we proceeded to graphical representations in order to study the quality of the underlying distance measures as well as the quality of the obtained clusters before and after the addition of extra noise.

Table V.
Notation of validity indices

Symbol	Description
$d_{\min}$	The minimum of the distances between the two closest points belonging to different clusters
$d_{\max}$	The maximum of the distances between the two most remote points in each cluster
$\mathrm{diam}(C)$	The diameter of cluster C_i, $i = 1, \ldots, k$ is the average distance of all cluster's members to the cluster's centre
$\mathrm{dist}(C_i, C_j)$	Distance between the centres of clusters C_i, C_j, $i, j = 1, \ldots, k$

Table VI.
Formulas of Dunn's and Davies-Bouldin validity indices

Symbol	Description
Dunn's index	$\mathrm{Dunn} = (d_{\min})/(d_{\max})$
Davies-Bouldin index	$\mathrm{DB} = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j}[(\mathrm{diam}(C_i) + \mathrm{diam}(C_j))/\mathrm{dist}(C_i, C_j)]$

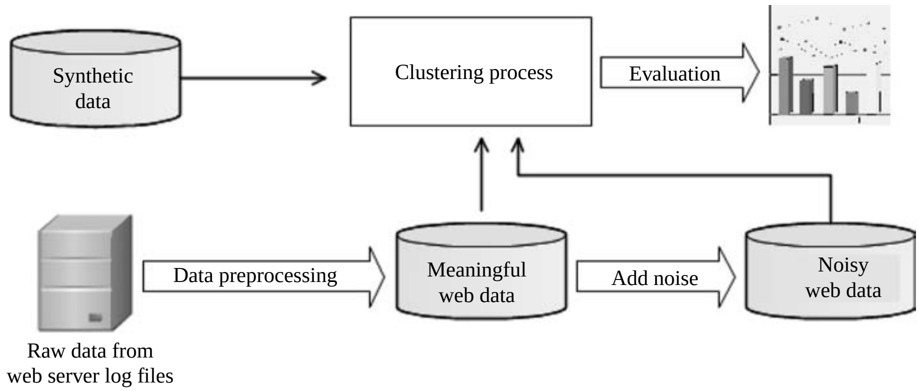
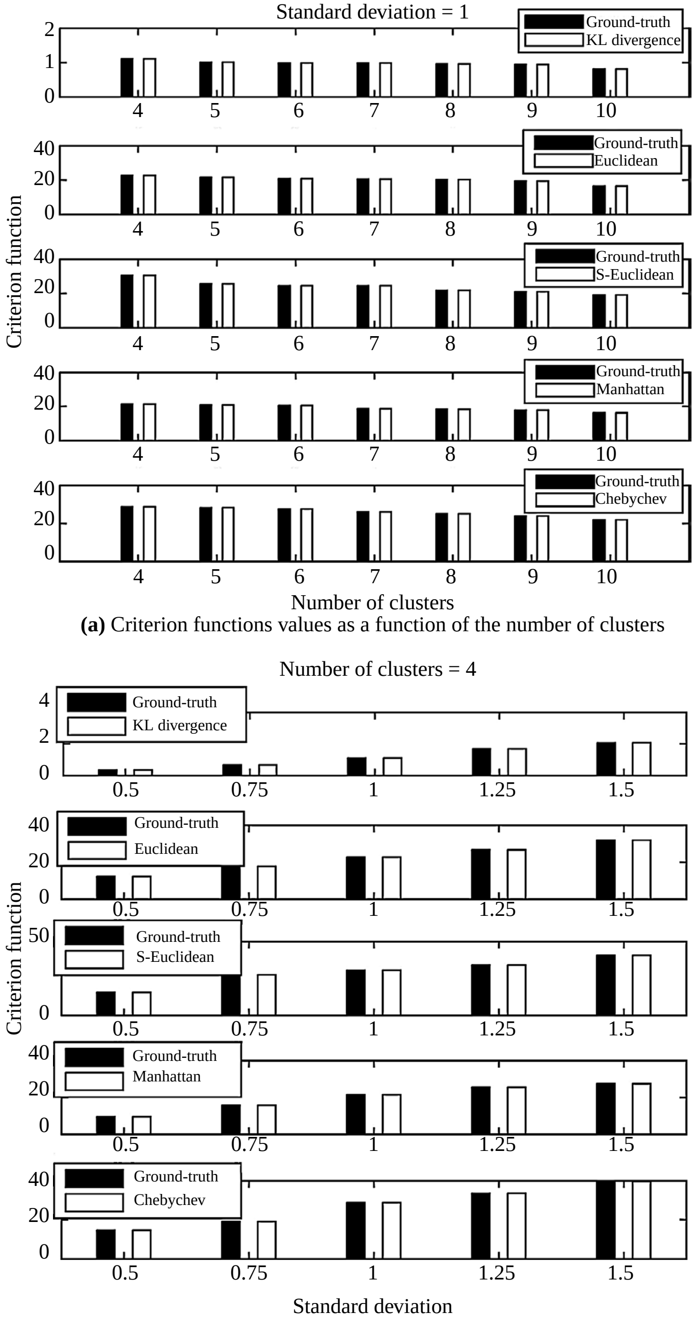


Figure 2.
Mining process

Clustering over synthetic datasets
We generated data based on a specific model and then checked whether the suggested method succeeded in discovering this model. More specifically, our synthetic datasets were generated as follows: we produced an $n \times m$ pattern table $\mathbf{X}$, whose data was divided in advance into k clusters (predefined clusters). For this $n \times m$ pattern table $\mathbf{X}$ we fixed the dimensionality m of the data. Then, for each cluster we selected random number of members while for each jth dimension ($j = 1, \ldots, m$) we selected a mean value μ_{ij}, which was uniformly distributed in [0…99]. Points were then generated by adding a value sampled from the normal distribution $N(\mu_{ij}, \sigma^2)$. For our experiments we fixed the values of users' to be around $n = 800$ and $m = 200$ pages. We created different datasets using $k = 4, 5, \ldots, 10$ clusters and standard deviation $\sigma = 0.5$, 0.75, 1, 1.25, 1.5.

The results for the synthetically generated data are shown in Figure 3(a) and (b), where we present criterion functions' values using KL-divergence, Euclidean, S-Euclidean, Manhattan, and Chebychev distances in comparison with the criterion functions' values of the original model that has been used for the synthetic data generation ("Ground-truth" bar). The calculation of the ground-truth criterion function is also based on the chosen distance and as a result we have a ground-truth bar for each distance. For all cases of synthetic data, both our proposed method and the other widely used distances, coincide with their corresponding ground-truth criterion function, i.e. all approaches discover the underlying model. Thus, our proposed scheme could be used in conjunction with distance-based clustering as effectively as the other approaches.

As shown in Figure 3(a) the values of criterion functions are decreasing, in accordance with the underlying number of clusters increase. This is natural since, as the number of clusters increases, data points (i.e. users) tend to be more close to their cluster centres. The criterion functions values in standard deviation terms is shown in Figure 3(b), where, as expected, the criterion functions values are shown to be increasing as the standard deviation increases (i.e. points "deviate" from their clusters' centres).



(a) Criterion functions values as a function of the number of clusters

(b) Criterion functions values as a function of the standard deviation

Figure 3.
Synthetic datasets

Experimentation under real datasets

Data workload

Our real data experimentation was based on two distinct sources of log files that recorded users' navigational behavior on an academic oriented host and on a general public, more popular server, in order that we experiment with both a medium-sized and a large-scale data source. More specifically, the first source dataset logs users' navigation during their visits to the web pages hosted by the AUTH Computer Science Department (CSD) web server[2], while the second dataset logs file from a busy National Aeronautics and Space Administration (NASA) web server[3].

In our approach, the prior to clustering data preprocessing removed any log entry that was not needed for the mining process, i.e. image files, css, swf, agent, or spider requests. In addition, log entries with status other than 200 which indicates success and 304 which indicates redirection, were removed. Furthermore, data cleaning involves removing log entries that are negligible to influence the results of the clustering process. In our case, these entries refer to users having less than 5 visits, because even though they pay only a few visits, they are too many in number and could mislead the clustering process. After data preprocessing on both datasets, the number of meaningful records for the web mining process has significantly been reduced, so as to work with useful for the clustering information. The details of our datasets are summarized in Table VII.

Considering the two log files that resulted from the preprocessing, we created two $n \times m$ pattern tables, one for each log file, where each row of them corresponds to a user and each column to a page that appears in the respective log file. Consequently, regarding the CSD and NASA datasets, the size of the produced tables is 445×175 and 456×70, respectively. Those two tables were normalized before the clustering process takes place.

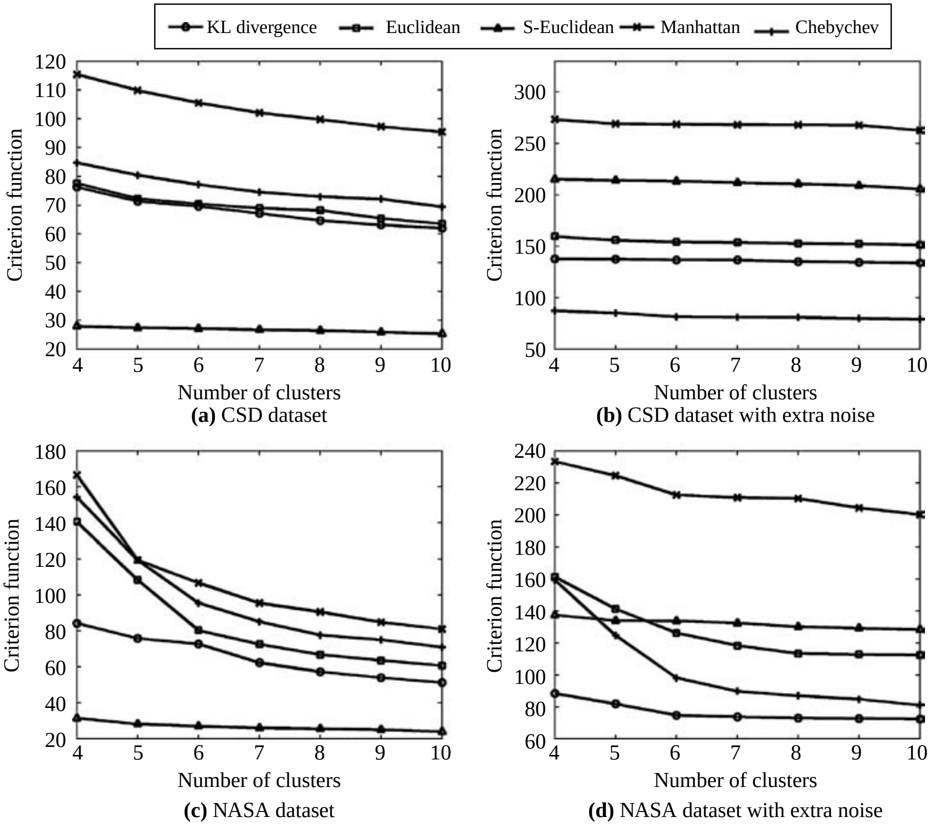
Clustering evaluation

Regarding hard partitional clustering methods, as the K-means employed in our approach, a common evaluation process relies on a cost function which is attempted to be optimized. In our case this cost function refers to the criterion function $J_{\mathrm{KL}}(U)$ (equation (3)) when the KL-divergence is used, and to the criterion function $J(U)$ (equation (4)), when the Euclidean, S-Euclidean, Manhattan, or Chebychev distance is chosen. The optimal value of both $\mathrm{J}_{\mathrm{KL}}(U)$ and $J(U)$ is defined as the minimum one, since they both express the sum of distances between the items belonging to a cluster and the cluster's centre. Criterion functions values were the first measure used to evaluate the results of the clustering process. A good clustering approach aims to retain low values of the criterion function as this is indicative of an appropriate clustering scheme.

Figure 4 presents the criterion functions' values for both real datasets (CSD, NASA) as a function of the number of clusters, when the clustering process uses the

Dataset	Time period	Users	Pages	Before preprocessing (MB)	After preprocessing (MB)
CSD	May-June 2004	445	175	100	5
NASA	Jul 1995	456	70	200	40

Table VII. Dataset details



(a) CSD dataset

(b) CSD dataset with extra noise

(c) NASA dataset

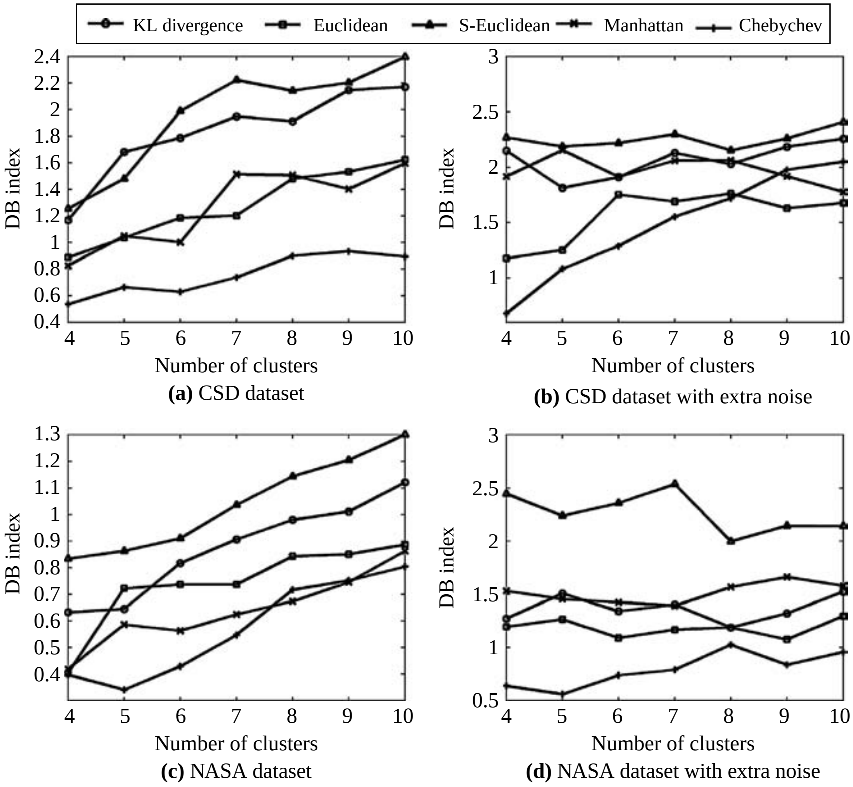
(d) NASA dataset with extra noise

Figure 4.
Criterion function values as a function of the number of clusters

five distance measures. More specifically, when KL-divergence is used, the criterion function $J_{\mathrm{KL}}(U)$ (equation 3) is computed, while in the case of the other distances we calculate the criterion function $J(U)$ (equation (4)), considering the respective distance measure d. Figure 4(a) and (c) show the values obtained by the original data whereas Figure 4(b) and (d) depict the corresponding values with the addition of extra noise to real data. As it was mentioned in the Introduction, in case of web data, noise refers to users' visits that happen accidentally or by mistake. So, we added noise to our datasets by slightly altering the probability with which a user visits a page, using a random function. The number of clusters used in all cases fluctuates in the interval [4. . .10].

We can observe that in all subfigures the criterion functions values decrease for both datasets as the number of clusters increases, and this is expected considering that the increase of the number of clusters results in more cohesive clusters. Furthermore, the values for all curves are in the interval [0. . .445] for CSD and [0. . .456] for NASA as expected by equation (9). In case of Figure 4(b) and (d), the addition of noise results in increased criterion functions values for all distances and number of clusters with the exception of Chebychev distance whose values remain at the same levels. This is expected since the Chebychev distance between two vectors is the maximum difference among their corresponding elements and thus it is not significantly changed by the addition of noise.

Despite the fact that curves in Figure 4 fall in the interval [0…*n*], a direct comparison of them would be meaningless since we use different distance measures. Thus, we proceed to a second evaluation method which is based on the DB validity index. The DB index values for all distances as a function of the number of clusters in both CSD and NASA datasets are depicted in Figure 5(a) and (c), respectively. When extra noise is added to the datasets, the obtained DB index values are shown in Figure 5(b) and (d). Here, we can see that the performance of KL-divergence approach is comparable to the approaches using the other four deterministic distance measures. This holds for both datasets where we can observe that the KL-divergence curve is among the others. Such a comparison is not meaningless, as in the case of criterion functions curves, since the DB values come from the division of distances and, in spite of the chosen distance, they are "pure" numbers. More specifically, as DB index takes into consideration the average distance of all items belonging to a cluster to their cluster centre and the distance between the cluster centres, it can be considerably affected by the chosen clusters' centres. Thus, even though the distance between items and their cluster centre decreases as the number of clusters increases (which was clear in criterion functions curves), DB index does not exhibit steadily



(a) CSD dataset

(b) CSD dataset with extra noise

(c) NASA dataset

(d) NASA dataset with extra noise

Figure 5. DB index values as a function of the number of clusters

decreasing curves as it considers the inter-cluster distances (distances between clusters' centres) too. This is clearly depicted in all subfigures of Figure 5.

With the addition of noise (Figure 5(b) and (d)) we can observe that values of DB index increase, in both datasets and for all distance measures used. This is explained by the fact that the addition of noise clearly affects clustering, no matter which distance metric is used. DB index values for S-Euclidean, Manhattan, and Chebychev distances, seem to be non-beneficial for the clustering process, since intervals that DB Index fluctuates with and without the addition of noise diverge less in the case of KL-divergence and Euclidean distances. Observing the values of DB index in Figure 5(a) and (b), we can see that there is a similar increase rate when using KL-divergence and Euclidean distance. This is quite interesting since a pure similarity distance such as Euclidean and a probabilistic one such as KL-divergence react in a similar way to the addition of noise. Moreover, the high values of DB index in the case of Chebychev distance prove that low values of its criterion function were not indicative for the clustering quality. For the CSD dataset, Table VIII presents the intervals that DB index fluctuates (for the different number of clusters *k*) before and after the addition of noise for all distance measures.

Under the NASA dataset (Figure 5(c) and (d)), we can also see that noise addition in the case of KL-divergence, Euclidean, and Chebychev distances causes similar deterioration considering the DB index fluctuation intervals, while the usage of S-Euclidean and Manhattan distances is non-beneficial for the clustering process. Table IX shows the DB index fluctuation intervals before and after the addition of noise for all distance measures.

The observed deteriorations differentiate concerning the two datasets for the respective distances, and this is due to the nature of the dataset. For example, if we consider the average deterioration when KL-divergence is used over all numbers of clusters in terms of DB index, we will find a percentage of 16 percent for the CSD dataset and 63 percent for the NASA dataset, while using Euclidean distance these percentages become 19 and 70 percent, respectively, for the two datasets.

Next, it is important to proceed to a graphical analysis in order to obtain evidence about the quality of the employed distance measures as well as of the clustering results. Thus, for the obtained clusters' visualization we have chosen to depict distance tables for all the employed distance measures.

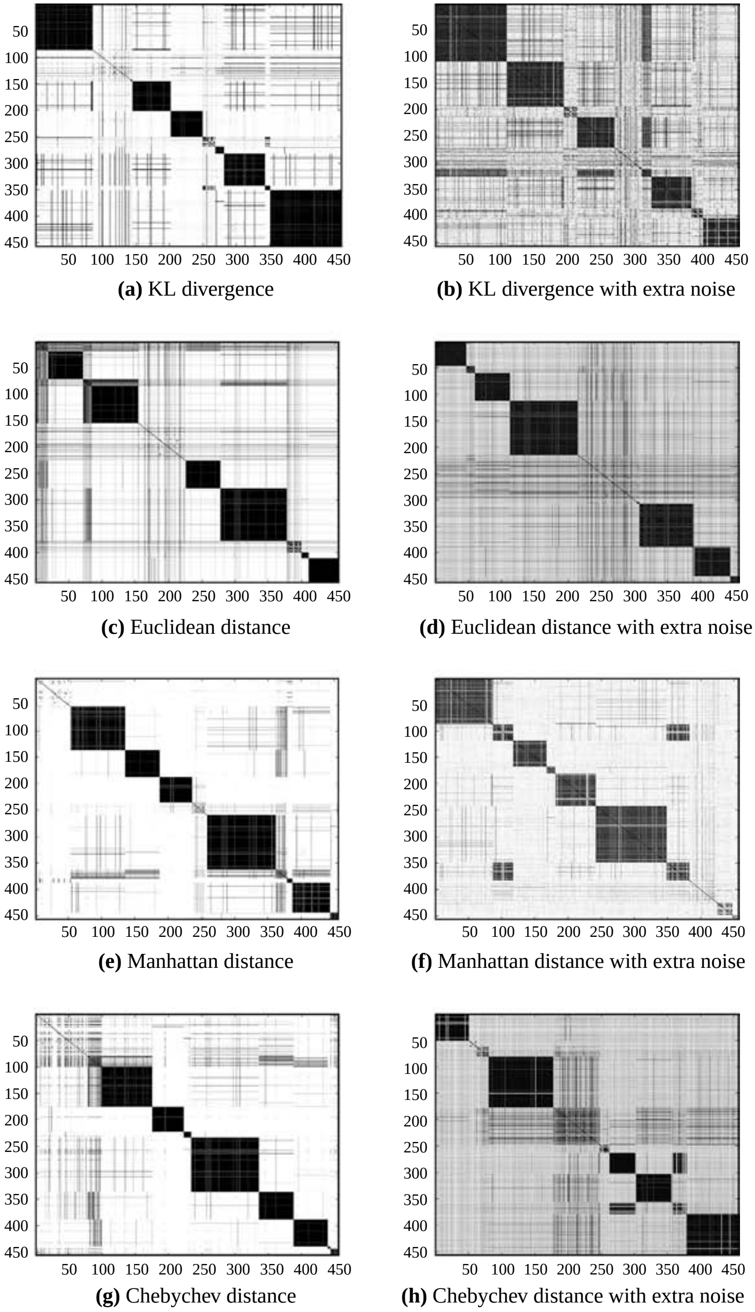
The conclusions about the quality of the distance measures we use are confirmed by the visualization of the obtained clusters. In Figure 6, we indicatively present the clustering outline of the NASA dataset for $k = 10$ clusters. More specifically, the

Table VIII.
DB index fluctuation intervals – CSD dataset

	KL-divergence	Euclidean	S-Euclidean	Manhattan	Chebychev
Before noise addition	[1.1 ... 2.1]	[0.8 ... 1.6]	[1.2 ... 2.3]	[0.8 ... 1.6]	[0.5 ... 0.9]
After noise addition	[1.9 ... 2.2]	[1.2 ... 1.7]	[2.2 ... 2.5]	[1.7 ... 2.2]	[0.7 ... 2.1]

Table IX.
DB index fluctuation intervals – NASA dataset

	KL-divergence	Euclidean	S-Euclidean	Manhattan	Chebychev
Before noise addition	[0.6 ... 1.1]	[0.4 ... 0.9]	[0.8 ... 1.3]	[0.4 ... 0.9]	[0.4 ... 0.8]
After noise addition	[1.2 ... 1.5]	[1.1 ... 1.3]	[1.9 ... 2.5]	[1.4 ... 1.7]	[0.5 ... 1.1]



(a) KL divergence

(b) KL divergence with extra noise

(c) Euclidean distance

(d) Euclidean distance with extra noise

(e) Manhattan distance

(f) Manhattan distance with extra noise

(g) Chebychev distance

(h) Chebychev distance with extra noise

Figure 6.
The clustering outline of the NASA dataset for $k = 10$

subplots of Figure 6 present the similarities and dissimilarities between user patterns in terms of the underlying distance measure before and after the addition of extra noise. Particularly, each plot depicts the $n \times n$ distance table D (in accordance with the chosen distance measure) whose rows and columns have been rearranged so that users clustered together are put in consecutive rows (columns). Therefore, each user does not necessarily correspond to the same axes' point in the different subplots. Moreover, the darker the shade of a cell (i,j), where $1 \leq i,j \leq n$, the more similar the users at positions i and j are. Thus, given that clusters contain the most similar users, the darker rectangles appear on the subplots diagonal and reveal the clusters of our dataset.

It is apparent that all approaches group users in NASA dataset in a similar way in terms of clusters' membership. For example, the larger cluster which corresponds to the larger rectangular of each diagonal contains 106, 100, 167, 103, 102 users in case of KL-divergence (Figure 6(a)), Euclidean (Figure 6(c)), Manhattan (Figure 6(e)), and Chebychev (Figure 6(g)) distance, respectively. As we observe, with the addition of extra noise (Figure 6(b), (d), (f), and (h)) clusters' cardinality remains to a similar level, e.g. 107, 102, 186, 107, 97 for the largest cluster in each approach. We omit figures for S-Euclidean, since S-Euclidean is shown to have the worst outcome as presented in Figure 5(c) and (d) and Table VIII and thus the visualization has no clear clustering distinction. Furthermore, all approaches succeed in finding coherent clusters (i.e. black rectangles of the diagonal) but they also provide less compact clusters (i.e. gray rectangles of the diagonal). Thus, before the addition of extra noise we notice that the KL-divergence is comparable to the other approaches as shown in the depicted rectangles corresponding to the $k = 10$ clusters.

As discussed earlier, the addition of noise deteriorates the quality of the clustering results. This deterioration is clearly depicted in Figure 6(b), (d), (f), and (h) where the rectangles of the diagonal are less dark compared to those of Figure 6(a), (c), (e), and (g), respectively, indicating that the corresponding clusters are less coherent. The degree to which the shade of the rectangles fades reveals the amount of deterioration. Figure 6(b) proves the tolerance of KL-divergence, since its diagonal retains the dark shade and clearly depicts the compactness of clusters with the exception of cluster 5 which contains 37 users. The corresponding cluster in case of the Euclidean distance is also less coherent but it consists of 89 users and this is represented by the larger rectangular of Figure 6(d). In terms of Manhattan distance (Figure 6(f)), the rectangles of the diagonal are less dark compared to those of Figure 6(b) and (d) indicating that clusters are more affected by noise. What is more, the last clusters containing 42, 23, and 9 users, respectively, seem to be the most affected. The diagonal of Figure 6(h) also denotes the deterioration that noise causes to the clustering results since three clusters with 11, 19, and 70 users, respectively, are vaguely depicted. The visualization of the obtained clusters is in accordance with the ascertainments based on DB index (Table IX).

In conclusion, all evaluation methods, i.e. criterion function, DB index and clusters' visualization prove that KL-divergence can be considered as a good alternative for measuring distances in noisy environments such as the web.

Conclusions and future work

In this paper, we compare the results of a divergence-oriented clustering approach with those of other clustering that use typical distance measures. The experimentation carried out involved both synthetic and real datasets. In the synthetic datasets' experimentation, clustering evaluation was carried out by using the values of the criterion functions J_{KL} and $J(U)$. The results of our evaluation showed that KL-divergence along with the other four distances successfully reveal the underlying structure of synthetic data, thus KL-divergence can be successfully used as an alternative option for measuring distances in web users clustering. In the real datasets' experimentation, we used two datasets and the evaluation of the clustering results was performed using the above criterion functions, DB as an internal validity index and a graphical representation of the obtained clusters. In this case, we also experimented by imposing noise on real data in order to estimate the tolerance of our proposed distance in comparison with the other distance measures. The results indicated that a clustering approach using KL-divergence exhibits significant tolerance to the addition of noise compared to the other typical distance measures and this is very important especially in case of the web, which is a noisy by nature environment.

Our next step is to use the idea of KL-divergence in conjunction with other partitional (e.g. K-medoids) and hierarchical (e.g. single linkage algorithms) algorithms as well as to employ other validity indices for cluster validation.

Notes

1. For three distinct data points x, y, z the triangle inequality is satisfied if it holds that $d(u_x, u_y) \leq d(u_x, u_z) + d(u_z, u_y)$.
2. AUTH Department of Informatics, available at: www.csd.auth.gr/
3. NASA server log file, available at: http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html

References

Baeza-Yates, R. and Frakes, W. (1992), *Information Retrieval: Data Structures and Algorithms*, Prentice-Hall, Upper Saddle River, NJ.

Boutin, F. and Hascoer, M. (2004), "Cluster validity indices for graph partitioning", *Proceedings of the 8th IEEE International Conference on Information Visualisation, London*, pp. 376-81.

Cadez, I., Heckerman, D., Meek, C., Smyth, P. and White, S. (2002), "Visualization of navigation patterns on a website using model-based clustering", Technical Report MSR-TR-00-18, Microsoft Research.

Castellano, G., Fanelli, A.M., Mencar, C. and Torsello, M.A. (2007), "Similarity-based fuzzy clustering for user profiling", *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops*, pp. 75-8.

Charikar, M., Guha, S., Tardos, E. and Shmoys, D. (1999), "A constant-factor approximation algorithm for the k-median problem", *Proceedings of the 31st Annual ACM Symposium on Theory of Computing, (STOC), ACM, Atlanta, GA, May 1-4*, pp. 1-10.

Cover, T. and Thomas, J. (1991), *Elements of Information Theory*, Wiley, New York, NY.

Davies, D. and Bouldin, D. (1979), "A cluster separation measure", *IEEE Transactions on Pattern Analysis and Machine Learning*, Vol. 1 No. 4, pp. 224-7.

Dhillon, I., Mallela, S. and Kumar, R. (2002), "Enhanced word clustering for hierarchical text classification", *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), ACM, Edmonton, Canada, July 23-26*, pp. 191-200.

Dunn, J.C. (1974), "Well separated clusters and optimal fuzzy partitions", *Journal Cybern*, Vol. 4 No. 3, pp. 95-104.

Fung, G. (2001), "A comprehensive overview of basic clustering algorithms", Technical Report, University of Winsconsin, Madison, WI.

Garey, M. and Johnson, D. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, New York, NY.

Gunter, S. and Bunke, H. (2003), "Validation indices for graph clustering", *Pattern Recognition Letters*, Vol. 24 No. 8, pp. 1107-13.

Halkidi, M., Batistakis, Y. and Vazirgiannis, M. (2002a), "Cluster validity methods: part I", *SIGMOD Record*, Vol. 31 No. 12, pp. 40-5.

Halkidi, M., Batistakis, Y. and Vazirgiannis, M. (2002b), "Clustering validity checking methods: part II", *SIGMOD Record*, Vol. 31 No. 3, pp. 19-27.

Jain, A., Murty, M. and Flynn, P. (1999), "Data clustering: a review", *ACM Computing Surveys*, Vol. 31 No. 3, pp. 264-323.

Kasturi, J., Acharya, R. and Ramanathan, M. (2003), "An information theoretic approach for analyzing temporal patterns of gene expression", *Bioinformatics*, Vol. 19 No. 4, pp. 449-58.

Kerr, M. and Churchill, A. (2001), "Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments", *Proceedings of National Academy of Sciences of the United States of America 98*, pp. 8961-5.

Lazzerini, B., Marcelloni, F. and Cococcioni, M. (2003), "A system based on hierarchical fuzzy clustering for web users profiling", *IEEE International Conference on Systems, Man and Cybernetics*, pp. 1995-2000.

Liu, X., He, P. and Yang, Q. (2005), "Mining user access patterns based on web logs", *Canadian Conference on Electrical and Computer Engineering*, pp. 2280-3.

McQueen, J. (1967), "Some methods for classification and analysis of multivariate observations", *Proceedings of the 5th Berkely Symposium on Mathematical Statistics and Probability, Berkeley, CA, June-July*, Vol. 1, pp. 281-97.

Mojica, J.A., Rojas, D.A., Gomez, J. and Gonzalez, F. (2005), "Page clustering using a distance based algorithm", paper presented at the 3rd Latin American Web Congress, October 31-November 2, p. 7.

Petridou, S., Koutsonikola, V., Vakali, A. and Papadimitriou, G. (2006), "A divergence-oriented approach for web users clustering", *Proceedings of International Conference on Computational Science and its Applications (ICCSA 2006), Glasgow*, pp. 1229-38.

Petridou, S., Koutsonikola, V., Vakali, A. and Papadimitriou, G. (2008), "Time aware web users clustering", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 20 No. 5, pp. 653-67.

Shokry, R.A., Saad, A.A., El-Makkey, N.M. and lsmail, M.A. (2006), "Using new soft clustering technique in adaptive web site", *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pp. 281-6.

Srivastava, J., Cooley, R., Deshpande, M. and Tan, P.-N. (2000), "Web usage mining: discovery and applications of usage patterns from web data", *SIGKDD Explorations*, Vol. 1 No. 2, pp. 12-23.

Stein, B., Meyer zu Eissen, S. and Wibrock, F. (2003), "On cluster validity and the information need of users", *3rd IASTED International Conference on Artificial Intelligence and Applications (AIA 03), ACTA Press, Benalmídena, Spain, September 8-10*, pp. 216-21.

Sturn, A. (2001), "Cluster analysis for large scale gene expression studies", Master's thesis, Graz University of Technology, Graz.

Vakali, A., Pokorny, J. and Dalamagas, T. (2004), "An overview of web data clustering practices", *Proceedings of the EDBT 2004 Workshop*, Lecture Notes in Computer Science (LNCS) Series, Springer Verlag, Heraklion, pp. 597-606.

Vakali, A., Pallis, G. and Angelis, L. (2006), "Clustering web information sources, published in the book", in Vakali, A. and Pallis, G. (Eds), *Web Data Management Practices: Emerging Techniques and Technologies*, Idea Group Publishing, Hershey, PA, pp. 34-55.

Wijaya, D.T. and Bressan, S. (2006), "Clustering web documents using co-citation, coupling, incoming, and outgoing hyperlinks: a comparative performance analysis of algorithms", *International Journal of Web Information Systems*, Vol. 2 No. 2, pp. 69-76.

Xu, R. and Wunsch, D.I. (2005), "Survey of clustering algorithms", *IEEE Transactions on Neural Networks*, Vol. 16 No. 3, pp. 645-78.

Yang, Y. and Padmanabhan, B. (2005), "GHIC: a hierarchical pattern-based clustering algorithm for grouping web transactions", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17 No. 9, pp. 1300-4.

Zeng, H.-J., Chen, Z. and Ma, Y.-M. (2002), "A unified framework for clustering heterogeneous web objects", *Proceedings of the 3rd International Conference on Web Information Systems Engineering, IEEE Press, Grand. Hyatt, Singapore*, pp. 161-72.

Zhang, K. (1995), "Algorithms for the constrained editing distance between ordered labeled trees and related problems", *Pattern Recognition*, Vol. 28 No. 3, pp. 463-74.

Zhao, Y. and Karypis, G. (2005), "Topic-driven clustering for document datasets", *Proceedings of the SIAM International Conference on Data Mining, Newport Beach, CA*, pp. 358-69.

Further reading

Agrawal, R. and Srikant, R. (1994), "Fast algorithms for mining association rules", *Proceedings of the 20th International Conference on Very Large Databases, Santiago*, pp. 487-99.

Eirinaki, M. and Vazirgiannis, M. (2003), "Web mining for web personalization", *ACM Transactions on Internet Technology*, Vol. 3 No. 1, pp. 1-27.

Mannila, H., Toivonen, H. and Verkamo, A. (1995), "Discovering frequent episodes in sequences", *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining, (KDD-95), American Association for Artificial Intelligence, Menlo Park, CA*, pp. 210-5.

Schockaert, S., De Cock, M., Cornelis, C. and Kerre, E.E. (2007), "Clustering web search results using fuzzy ants", *International Journal of Intelligent Systems*, Vol. 22 No. 5, pp. 455-74.

Shen, D., Chen, Z., Yang, Q., Zeng, H.-Z., Zhang, B., Lu, Y. and Ma, W.-Y. (2004), "Text classification: web-page classification through summarization", *Proceedings of the 27th ACM International Conference on Research and Development in Information Retrieval, SIGIR'04, Sheffield, July 25-29*, pp. 242-9.

Vakali, A. and Papadimitriou, G. (2004), "Web engineering: the evolution of new technologies", *Guest Editorial in IEEE Computing in Science and Engineering*, Vol. 6 No. 4, pp. 10-11.

Zhang, Y.J. and Liu, Z.Q. (2004), "Refining web search engine results using incremental clustering", *International Journal of Intelligent Systems*, Vol. 19 Nos 1/2, pp. 191-9.

Corresponding author

Vassiliki A. Koutsonikola can be contacted at: vkoutson@csd.auth.gr