Mining Web Access Logs Using a Fuzzy Relational Clustering Algorithm Based on a Robust Estimator

Olfa Nasraoui
Department of Computer Engineering and Computer Science
University of Missouri,Columbia, MO, 65211
olfa@ece.missouri.edu

Raghu Krishnapuram
Department of Mathematical and Computer Sciences
Colorado School of Mines, Golden, CO 80401
rkrishna@mines.edu

Anupam Joshi
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County, Baltimore, MD 21250
joshi@cs.umbc.edu

Abstract

The proliferation of information on the world wide web has made the personalization of this information space a necessity. Mining typical user profiles from the vast amount of historical data stored in access logs is an important component of web personalization. In the absence of *a priori* knowledge, unsupervised or clustering methods seem to be ideally suited for this job. In this paper, we define the notion of a "user session" as being a temporally compact sequence of web accesses by a user. We also define a dissimilarity measure between two web sessions that captures the organization of a web site. To cluster the user sessions based on the pair-wise dissimilarities, we introduce the Relational Fuzzy $C-$Maximal Density Estimator (RF$C-$MDE) algorithm. RF$C-$MDE is robust, and can deal with outliers that are typical in this application. We show real examples of the use of RF$C-$MDE for extraction of user profiles from log data, and and compare its performance to the standard Non Euclidean Fuzzy $C-$ Means (NERF). These examples illustrate that as a by-product of the clustering process that generates robust profiles, associations between different URL addresses on a given site can easily be inferred. In general, the URLs that are present in the same profile tend to be visited together in the same session or form a large *item set*.

Keywords:
Web mining, User profile, Session profile similarity measure, Relational clustering, Robust clustering, Maximal Density Estimator.

1 Introduction

The proliferation of information on the world wide web has made the personalization of this information space a necessity. This means that a user's interaction with the web information space should be tailored based on information about him/her. For example, a person in Switzerland searching for ski resorts is more likely to be interested in the Alps, whereas a person in Colorado would likely be interested in the Rockies. Personalization can either be done via information brokers (e.g. web search engines), or in an *end-to-end* manner by making web sites adaptive. Initial work in this area has basically focused on creating recommender systems. One of the earliest such systems was the Firefly system [Fir] which attempted to provide CDs that best match a user's professed interests. More recently, systems such as W^3IQ [JWH97, AJ98] and PHOAKS [THA97] have sought to use cooperative information retrieval techniques for personalization. There has been some recent work in actually "mining" information from the user's interaction with the web to effect personalization. The basic ideas can be found in [PE97, PE98, KJ98]. Perhaps the earliest work along similar lines was the Webwatcher project [AFJM95] at CMU. It highlights hyperlinks in a page based on the declared interests and the path traversal of a user as well as the path traversals of previous users with similar interests. There is also a recent body of work [AM98] which seeks to transform the web into a more structured, database like entity and then mine on this [OH98].

The standard K-Means algorithm was used to cluster users's traversal paths in [SZAS97]. However, it is not clear how the similarity measure was devised and whether the clusters are meaningful. In [MJHS96], associations and sequential patterns between web transactions are discovered based on the Apriori algorithm [AS94]. It is important to mention that so far, most efforts have relied on relatively simple techniques which can be inadequate for real user profile data since they are not resilient to the "noise" typically found in user traversal patterns. Moreover such data, as we elaborate later, are usually fuzzy in nature.

In the absence of any *a priori* knowledge, unsupervised classification or clustering methods seem to be ideally suited to analyze the semi-structured log data of user accesses by categorizing them into classes of user session profiles. We define the notion of a "user session" as being a temporally compact sequence of web accesses by a user. The goal of our web mining is to categorize these sessions. In this light, web mining can be viewed as a special case of the more general problem of knowledge discovery in databases [FPSSU96]. We define a new distance measure between two web sessions that captures the organization of a web site. This organizational information is inferred directly from the URLs.

Web mining involves data that is mildly to severely corrupted with "noise". Outliers and incomplete data can easily occur in the data set due to a wide variety of reasons inherent to web browsing and logging. Moreover, the noise contamination rate and the scale of the data is rarely known in advance. For example, consider the situation where we are analyzing log entries to discover typical information access patterns. Clearly, there is a significant percentage of time (sometimes as large as 20-30 percent) that a user is simply "browsing" the web and does not follow any particular pattern. Hence, there is a need for robust methods that are free of any assumptions about the noise contamination rate and scale in this task. Most of the existing robust estimators suffer from their strong dependence on a good initialization, and their reliance on a known or assumed amount of noise present in the data set (contamination rate), or equivalently an estimated scale value or inlier bound. For instance, the Least Trimmed Squares [RL87] and R-estimators [KJ78] assume a given noise contamination rate, and in the Least Median of Squares [Rou84] the noise contamination rate is fixed to 50%. When faced with more noise than assumed, all these estimators will lack robustness. And when the amount of noise is less than the assumed level, they will lack efficiency, i.e., the parameter estimates suffer in terms of accuracy, since not all the good data points are taken into account. The Maximal Density Estimator (MDE) yields a robust estimate of the parameters by minimizing an objective function that incorporates both the overall error and the contamination rate via a set of robust weights and a scale factor or inlier bound. Unlike most existing robust estimators, MDE is computationally attractive and

far less sensitive to initialization, or to the amount of noise present in the data set, as will be illustrated with several synthetic examples. An extension of this estimator, called the Fuzzy $C-$ Maximal Density Estimator (F$C-$MDE), can perform clustering or parameter estimation in the case when multiple components are present in the data set.

Categories and clusters are an important aspect of web mining. Categories in most data mining tasks are rarely well separated. The class partition is best described by fuzzy memberships [Bez81], particularly along the overlapping borders. Specifically, in web mining, certain access patterns may be regarded as belonging to more than one class with different degrees instead of belonging to only one class. For this reason, F$C-$MDE incorporates fuzzy membership degrees in the clustering process. Also, at times, it will be necessary to work with relational[1] data because it is intuitively and conceptually easier to describe the relation or similarity between two objects than to map them to numerical features. In fact, determining the nature and number of features is itself very challenging. This problem is particularly accute when the data contains nonnumeric fields. This means that relational clustering methods are expected to be highly beneficial in web mining. As will be explained later, web sessions are too complex to convert to simple numerical features. Hence, clustering the user sessions should be tackled by exploiting inter-session similarities within a relational framework. However, the Fuzzy $C-$ Maximal Density Estimator (F$C-$MDE) deals with object or feature data only. Moreover, the session dissimilarity measure defined in Section 2 is not Euclidean. Therefore, we extend the F$C-$MDE so that it can work on non-Euclidean relational data. The resulting Relational Fuzzy $C-$ Maximal Density Estimator (RF$C-$MDE) can deal with complex and subjective distance/similarity measures which are not restricted to be Euclidean. In addition, RF$C-$MDE is reliable, computationally attractive, and practically insensitive to initialization. The web site for the department of Computer Engineering and Computer Sciences at the University of Missouri, as well as that of the department of Computer Sciences and Electrical Engineering at the University of Maryland-Baltimore County were used as a testbed for RF$C-$MDE which successfully analyzed both server access logs and obtained typical session profiles of users. In both experiments, the performance of RF$C-$MDE is compared to that of the Non Euclidean Fuzzy $C-$ Means (NERF), and shown to be superior, particularly in being more resistant to noise.

We also note that as a by-product of our clustering process, associations between different URL addresses on a given site can easily be inferred from the resulting robust profiles. In general, the URLs that are present in the same profile tend to be visited together in the same session, i. e., they form a large item set. These item sets are similar to the ones computed by the Apriori algorithm [AS94], and have been applied to mining associations and sequential patterns between web transactions in [MJHS96]. However, unlike the latter, our method mines typical users profiles as a primary goal, and the discovered associations resulting from this profile extraction come with no additional computation.

The rest of the paper is organized in sections where each section represents a different phase of the knowledge discovery process [FPSSU96], i.e., the categorization of user profiles. In Section 2, we define the similarity between user sessions. In Section 3, we describe the Maximal Density Estimator (MDE), and extend it to include multiple estimates with fuzzy membership degrees. In Section 4, we further extend it to work on non-Euclidean relational data, resulting in the Relational Fuzzy C Maximal Density Estimator (RF$C-$MDE). In Section 5, we define quantitative measures to help us in the interpretation and evaluation of the results of mining the access log data. We also present our experimental results. In Section 6, we conclude with a discussion of our ongoing work.

[1]Note that this term is used in its statistical sense, not in its database sense

2 Defining the Similarity Between User Sessions

2.1 Preprocessing and Segmentation of the access log data into sessions

The access log for a given web server consists of a record of all files accessed by users. Each log entry consists of :(i) user's IP address, (ii) access time, (iii) request method ("GET", "POST", $\cdots$), etc), (iv) URL of the page accessed, (v) data transmission prototcol (typically HTTP/1.0), (vi) return code, and (vii) the number of bytes transmitted. First, we filter out log entries that are not germane for our task. These include entries that: (i) result in any error (indicated by the error code), (ii) use a request method other than "GET", or (iii) record accesses to image files (.gif, .jpeg, , $\cdots$, etc), which are typically embedded in other pages and are only transmitted to the user's machine as a by product of the access to a certain web page which has already been logged.

Next, analogous to [MJHS96], the individual log entries are grouped into user sessions. A user session is defined as a sequence of temporally compact accesses by a user. Since web servers do not typically log usernames (unless *identd* is used), we define a user session as accesses from the same IP address such that the duration of time elapsed between any two consecutive accesses in the session is within a prespecified threshold. Each URL in the site is assigned a unique number $j \in \{1, \ldots, N_U\}$, where N_U is the total number of valid URLs. Thus, the i^{th} user session is encoded as an N_U-dimensional binary attribute vector $\mathbf{s}^{(i)}$ with the property:

$$s_j^{(i)} = \begin{cases} 1 & \text{if the user accessed the } j^{th} \text{ URL during the } i^{th} \text{ session} \\ 0 & \text{otherwise.} \end{cases}$$

The ensemble of all N_S sessions extracted from the server log file is denoted by $\mathcal{S}$. Note that our scheme will map one user's multiple sessions to multiple user sessions. However, this is not of concern since our attempt is to extract "typical user session profiles". If we assume that the majority of a user's sessions follow a similar profile, then clearly no difference is made. On the other hand, this notion of multiple user sessions enables us to better capture the situation when the same user displays a few (different) access patterns on this site. Our approach can be combined with other features (such as cookies) to actually map a typical profile back to a particular user.

2.2 Adaptation of Session Data to Clustering: Computing The Relation Matrix

In the absence of any a priori knowledge, an unsupervised classification or clustering method seems to be ideally suited to partition the user sessions. There are two major classes of clustering techniques, those that work with "object data" or feature vectors, and those that work on "relational data" (similarities or dissimilarities between the data). Even though the first class of clustering algorithms is more popular and has received a lot of attention, it is not suitable for clustering user sessions. This is because of the high dimensionality of the feaure space (there are usually several hundred URLs at a typical web site), and the likely correlation between the features (URLs that often co-occur in the same session). The web sessions are too complex to convert to simple numerical features, partly because the organization of the web site must be taken into account. In fact, the URLs in a site have a hierarchical or tree-like structural composition. Therefore, we define a similarity measure between two sessions that incorporates both the structure of the site, as well as the URLs involved.

We chose the relational approach to clustering since our data (sessions) are not numeric in nature. This approach requires the definition and computation of the dissimilarity/similarity

between all session pairs (i.e., the relation matrix) prior to the clustering process. We start by defining the similarity measure between two user-sessions: $\mathbf{s}^{(k)}$ and $\mathbf{s}^{(m)}$. We first consider the simple case where the individual attributes or URLs accessed in the sessions are totally independent and the structure of the site is ignored. Then, we can simply use the cosine of the angle between $\mathbf{s}^{(k)}$ and $\mathbf{s}^{(l)}$ as a measure of similarity

$$S_{1,kl} = \frac{\sum_{i=1}^{Nu} \mathbf{s}_i^{(k)} \mathbf{s}_i^{(l)}}{\sqrt{\sum_{i=1}^{Nu} \mathbf{s}_i^{(k)}} \sqrt{\sum_{i=1}^{Nu} \mathbf{s}_i^{(l)}}} \tag{1}$$

It can be seen that $S_{1,kl}$ simply measures the number of identical URLs accessed during the two sessions relative to the number of URLs accessed in both sessions. $S_{1,kl}$ has the desirable properties: $S_{1,kk} = 1$, $S_{1,kl} = S_{1,lk}$, and $S_{1,kl} > 0, \forall k \neq l$. The problem with this similarity measure is that it completely ignores the hierarchical organization of the web site, which will adversely affect the ability to capture correct profiles. For example, the session pair {/courses/cecs345} and {/courses/cecs343}, as well as the session pair {/courses/cecs345} and {/research/grants} will receive a 0 similarity score according to S_1. However, it is evident that the first two sessions are more similar than the second two, because both users in the first sessions seem to be interested in courses. Similarly, one would expect the sessions {/courses/cecs345/projects/proj1} to be more similar to {/courses/cecs345/projects} than to {/courses/cecs343}, because there is more overlap between the URLs in the first two sessions along the directory hierarchy tree. This leads us to define a similarity measure on the structural URL level that will be used in the computation of the similarity at the session level.

We model the web site as a tree with the nodes representing different URLs. The nodes are essentially the directory structure rooted at the server's *document root*, with links (such as redirects and aliases) explicitly brought in. An edge connects one node to another if the URL corresponding to the latter is hierarchically located under that of the former, for example {/courses} and {/courses/cecs345}. The root of the tree (the node with no incoming edges) corresponds to the document root (/) of the server. Taking into account the syntactic representaion of two URLs, their similarity is assessed by comparing the location of their corresponding nodes on the tree. This is done by comparing the paths from the root of the tree to the two nodes. Hence, we define the "syntactic" similarity between the i^{th} and j^{th} URLs as

$$S_u(i,j) = \min\left(1, \frac{|p_i \cap p_j|}{\max\left(1, \max\left(|p_i|, |p_j|\right) - 1\right)}\right) \tag{2}$$

where p_i denotes the path traversed from the root node to the node corresponding to the i^{th} URL, and $|p_i|$ indicates the length of this path or the number of edges included in the path. Note that this similarity which lies in $[0, 1]$ basically measures the amount of overlap between the paths of the two URLs. Now the similarity on the session level which incorporates the syntactic URL similarities is defined by correlating all the URL attributes and their similarities in two sessions as follows

$$S_{2,kl} = \frac{\sum_{i=1}^{N_U} \sum_{j=1}^{N_U} s_i^{(k)} s_j^{(l)} S_u(i,j)}{\sum_{i=1}^{N_U} s_i^{(k)} \sum_{j=1}^{N_U} s_j^{(l)}} \tag{3}$$

Unlike S_1, this similarity uses soft URL level similarieties that vary between 0 and 1 (depending on their conceptual similarity), instead of the hard similarities 0 or 1 (depending on whether they are different or identical respectively). For the special case when all the URLs accessed during session $s^{(k)}$ have zero similarity with the URLs accessed during session $s^{(l)}$, i.e., $S_u(i,j) = 0$ if $i \neq j$, $S_{2,kl}$ reduces to $S_{2,kl} = \frac{\sum_{i=1}^{N_U} s_i^{(k)} s_i^{(l)}}{\sum_{i=1}^{N_U} s_i^{(k)} \sum_{j=1}^{N_U} s_j^{(l)}}$ and when the two sessions are idential, this value

further simplifies to $S_{2,kk} = \frac{1}{\sum_{i=1}^{N_U} s_i^{(k)}}$ which can be considerably small depending on the number of URLs accessed. This means that this similarity measure will be rather unintuitive, because ideally the similarity should be maximal for two identical sessions. Besides identical sessions, this similarity will generally be underestimated for session pairs who share some identical URLs while the rest of the unshared URLs have low syntactic similarity. In general for such sessions where the syntactic URL similarities are low, $S_{1,kl}$ provides a higher and more accurate session similarity. On the other hand, when the syntactic URL similarities are high, $S_{2,kl}$ is higher and more accurate. Therefore, we define a new similarity between two sessions that takes advantage of the desirable properties of S_1 and S_2 as follows:

$$S_{kl} = \max(S_{1,kl}, S_{2,kl}). \tag{4}$$

For the purpose of relational clustering, this similarity is mapped to the dissimilarity measure $d_s^2(k,l) = (1 - S_{kl})^2$. Note that squaring the complement of the similarity has the effect of amplifying the difference between similar and different sessions. This dissimilarity measure satisfies the desirable properties: $d_s^2(k,k) = 0$, $d_s^2(k,l) >= 0, \forall k,l$, and $d_s^2(k,l) = d_s^2(l,k), \forall k,l$. However, unlike a metric distance it is possible for two distinct sessions to have zero dissimilarity. This occurs whenever $\sum_{i=1}^{N_U}\sum_{j=1}^{N_U} s_i^{(k)} s_j^{(l)} S_u(i,j) = \sum_{i=1}^{N_U} s_i^{(k)} \sum_{j=1}^{N_U} s_j^{(l)}$, or equivalently $\sum_{j=1}^{N_U} s_i^{(k)} s_j^{(l)} S_u(i,j) = s_i^{(k)} \sum_{j=1}^{N_U} s_j^{(l)}$ for all $i = 1, \ldots, N_U$. This is particularly true if the URL level similarities are 1 for all the URLs accessed in the two sessions. A typical example consists of the sessions {/courses/cecs345} and {/courses/cecs345/syllabus.html}. This property is actually desirable for our application, because we consider these two sessions to fit the same profile. The session dissimilarity measure also violates the triangular inequality for metric distances in some cases. For instance, the dissimilarity between the sessions {/courses/cecs345/syllabus} and {/courses/cecs345} is zero. So is the dissimilarity between {/courses/cecs345} and {/courses/cecs401}. However, the dissimilarity between {/courses/cecs345/syllabus} and {/courses/cecs401} is not zero (it is 1/4). This illustrates another desirable property for profiling sessions which is that the dissimilarity becomes more stringent as the accessed URLs get farther from the root because the amount of specificity in user accesses increases correspondingly. Hence, the proposed dissimilarity measure fits our subjective criteria of session similarity.

3 The Fuzzy $C-$ Maximal Density Estimator

3.1 The Maximal Density Estimator

The Maximal Density Estimator (MDE) [NK96] is a new robust estimator that is free of any presuppositions about the noise proportion. Let $\mathcal{X} = \{\mathbf{x}_j \mid j = 1, \ldots, N\}$ be a set of feature vectors in an $n-$dimensional feature space with coordinate axis labels $x_1, x_2, \ldots, x_n$, where $\mathbf{x}_j = (x_{j1}, x_{j2}, \ldots, x_{jn})$. Let θ represent the parameters to be estimated. The Maximal Density Estimator (MDE) yields a robust estimate of the parameters by minimizing an objective function that incorporates both the overall error and the contamination rate via a set of robust weights and a scale factor or inlier bound, as follows:

$$\min_{\theta,\sigma} \quad \left\{ J = \sum_{j=1}^{N} w_j \frac{d_j^2}{\sigma} - \alpha \sum_{j=1}^{N} w_j \right\}. \tag{5}$$

In (5), d_j^2 is the residual of data point $\mathbf{x}_j$ with respect to the fit computed from the parameters θ, and w_j is a positive weight associated with point $\mathbf{x}_j$. The weight w_j can be considered as the degree of membership of data point $\mathbf{x}_j$ in the inlier set or the set of good points. The first term of

this objective function tries to minimize the scaled residuals of the good points. The second term of this objective function tries to use as many good points (inliers) as possible in the estimation process, via their high weights. Thus the combined effect is to optimize the density, i.e., the ratio of the total number of good points to the volume. In the first term, the distances are normalized by the scale measure to counteract the tendency of the scale to shrink towards zero, and to penalize the inclusion of outliers in the estimation process in a way that is independent of scale. We choose the value of α to balance the two terms. In 2-D, we choose $\alpha = 1$. For the general n-dimensional case α should be close to n, since the ratio of the first term to the second term approaches the average of a χ^2 distribution for Gaussian data. Finally, we should note that d_j^2 should be a suitable distance measure, tailored to detect desired shapes, such as the Euclidean distance for spherical clusters, or the Gustafson-Kessel (GK) distance [GK79] for ellipsoidal clusters characterized by a covariance matrix, etc.

Since the objective function depends on several variables, we can use the alternating optimization technique, where in each iteration a set of variables is optimized while fixing all others. If the weights are fixed, then the optimal prototype parameters are found by setting

$$\frac{\partial J}{\partial \theta} = \frac{1}{\sigma} \quad \sum_{j=1}^{N} w_j \frac{\partial d_j^2}{\partial \theta} = \mathbf{0}.$$

For instance if d_j^2 is the squared Euclidean distance $d_j^2 = \|\mathbf{x}_j \ - \ \mathbf{c}\|^2$, we obtain

$$\mathbf{c} = \frac{\sum_{j=1}^{N} w_j \mathbf{x}_j}{\sum_{j=1}^{N} w_j}. \tag{6}$$

To derive the optimal scale regardless of the distance measure being used, we fix the prototype parameters, and set

$$\frac{\partial J}{\partial \sigma} = 0.$$

Further simplification of this equation depends on the definition of the weight function. We choose to use the Gaussian weight function

$$w_j = \exp \frac{-d_j^2}{2\sigma} \tag{7}$$

to obtain the following update equation for the scale parameter

$$\sigma = \frac{1}{(2+\alpha)} \frac{\sum_{j=1}^{n} w_j d_j^4}{\sum_{j=1}^{n} w_j d_j^2}. \tag{8}$$

Therefore, the algorithm consists of iterative updates of the prototype parameters, followed by updates of the scale parameter and the weights until convergence, or until a fixed maximum number of iterations is reached.

The following examples illustrates the performance of MDE in the presence of noise with varying contamination rates, as well as its low sensitivity to initialization. Fig. 1(a) shows a noiseless Gaussian data set. Fig. 1(b) shows the results of MDE where the center location is depicted using a cross symbol and the circular contour shows the inlier bound which delimits the inlier set ($w_j > \epsilon = 10^{-7}$). Fig. 1(c)-(d) show the result of MDE on the same data set with added noise amounting to the contamination rates 50% and 88%, respectively. It can be seen from this simulation example that MDE is quite insensitive to the amount of noise, and that it succeeds in estimating an acceptable inlier bound in all these cases.

To illustrate the effect of the initialization on the performance of the MDE, Fig. 2(a)-(d) show different initializations which all result in the same center and inlier bound displayed in Fig. 1(d). These initializations have centers which are located far from all good points and varying initial inlier bounds. Hence, the initial inlier sets predominantly consist of noise points. Most existing robust estimators (e.g. M-estimators with redescending weight functions [RL87] cannot recover from such bad initializations.

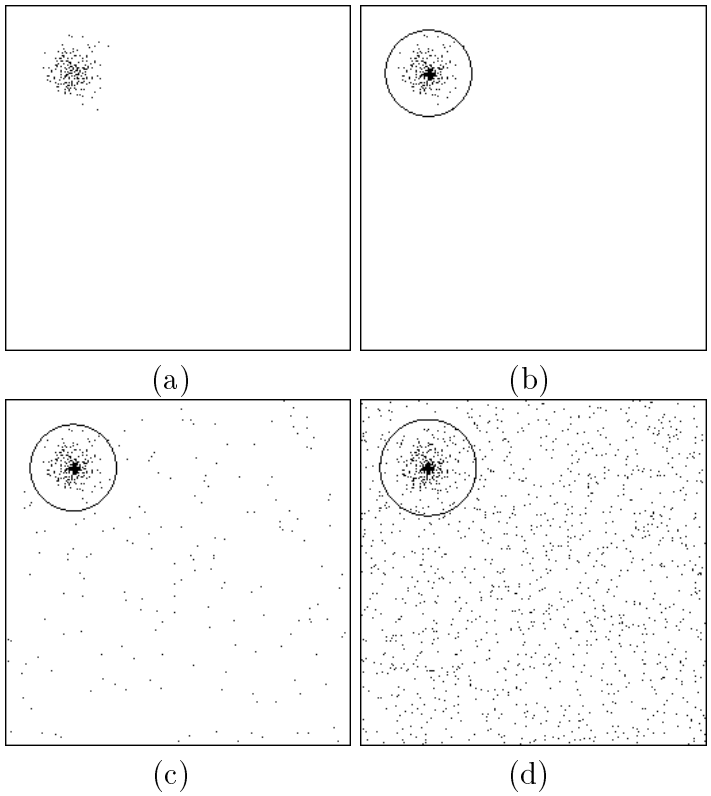

Figure 1: Effect of noise contamination rate on MDE.
(a) Original data set. (b) Results of MDE,
(c) Data set in (a) contaminated with 50% uniform noise and results of MDE.
(d) Data set in (a) contaminated with 88% uniform noise and results of MDE.

3.2 Fuzzy Clustering with the Maximal Density Estimator

It is known that for complex data sets containing overlapping clusters, fuzzy partitions model the data better than their crisp counterparts. In particular, fuzzy memberships are richer than crisp memberships in describing the degrees of belonging of data points lying in the areas of overlap. Moreover, fuzzy partitions generally make the optimization process less prone to local or sub-optimal solutions. With a fuzzy partition, a data point $\mathbf{x}_j$ belongs to each cluster, $\mathcal{X}_i$, to a varying degree called fuzzy membership u_{ij}. A fuzzy partiton must satisfy the following constraints [Bez81],

$$0 \leq u_{ij} \leq 1$$

$$\sum_{i=1}^{C} u_{ij} = 1 \; \forall j = 1 \cdots N$$

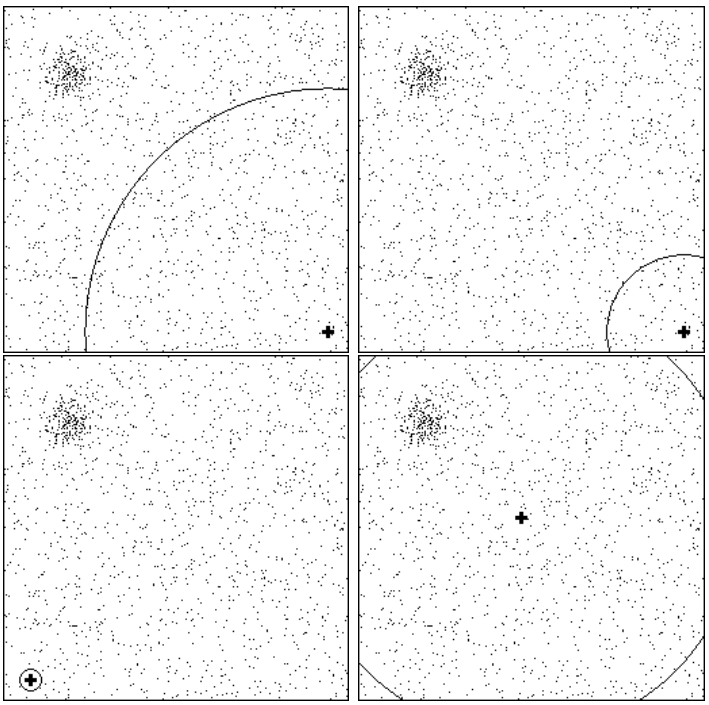

Figure 2: Different initializations for the MDE, all resulting in Fig. 1(d).

$$0 \leq \sum_{j=1}^{N} u_{ij} \leq N \ \forall i = 1 \cdots C$$

Using fuzzy memberships offers other advantages in terms of smoothing the surface of the objective function and reducing its discontinuity [NK95], thus making the problem analytically more tractable. One way to use a fuzzy partition while estimating the cluster parameters with the MDE is to use the following criterion

$$\min_{\boldsymbol{\Theta},\sigma_i,u_{ij}} \quad \left\{ J = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^m w_{ij} \frac{d_{ij}^2}{\sigma_i} - \alpha \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^m w_{ij} \right\}, \tag{9}$$

where $m \in [1, \infty)$ is a fuzzifier parameter that controls the degree of fuzziness of the resulting partition, with values closer to 1 corresponding to crisper partitions, and w_{ij} is given by

$$w_{ij} = \exp \frac{-d_{ij}^2}{2\sigma_i}. \tag{10}$$

Since each cluster is independent of the rest, it is easy to show that the optimal update equations are similar to the ones obtained for estimating the parameters for one cluster. The scale parameter of the i^{th} cluster is given by

$$\sigma_i = \frac{1}{(2+\alpha)} \frac{\sum_{j=1}^{N} u_{ij}^m w_{ij} d_{ij}^4}{\sum_{j=1}^{N} u_{ij}^m w_{ij} d_{ij}^2}. \tag{11}$$

To find the optimal prototype parameters θ_i of the i^{th} cluster, we set

$$\frac{\partial J}{\partial \theta_i} = \frac{1}{\sigma_i} \quad \sum_{\mathbf{x}_j \in \mathcal{X}_i} u_{ij}^m w_{ij} \frac{\partial d_{ij}^2}{\partial \theta_i} = \mathbf{0}.$$

For instance for the case of location estimation, d_{ij}^2 is the squared Euclidean distance $d_{ij}^2 = \|\mathbf{x}_j - \mathbf{c}_i\|^2$, and the center $\mathbf{c}_i$ is given by

$$\mathbf{c}_i = \frac{\sum_{j=1}^N u_{ij}^m w_{ij} \mathbf{x}_j}{\sum_{j=1}^N u_{ij}^m w_{ij}} \tag{12}$$

To simplify the algorithm, we de-couple the robust parameter estimation process via the robust weights, w_{ij}, from the partitionning process, via the memberships, u_{ij}. Since the parameter estimates in (12) which optimize (9) are already robust, we use the simpler Fuzzy c-Means [Rus69, Bez81] objective function to update the memberships u_{ij} in each iteration as follows:

$$u_{ij} = \frac{\left(\frac{1}{d_{ij}^2}\right)^{\frac{1}{m-1}}}{\sum_{k=1}^C \left(\frac{1}{d_{kj}^2}\right)^{\frac{1}{m-1}}}. \tag{13}$$

Therefore, the optimization process will consist of alternating updates of the memberships, as given by (13), and the cluster centers and scale parameters as given by (12) and (11). We call the resulting clustering algorithm the Fuzzy $C-$ Maximal Density Estimator (F$C-$MDE).

We illustrate the performance of F$C-$MDE in the presence of varying amounts of noise with a data set containing clusters of different sizes and densities in Fig. 3(a). F$C-$MDE was performed with $C = 5$ and $m = 2.2$. It was initialized using FCM with $m = 3$. The initial scale parameters were all initialized to the fuzzy average intracluster distance, $\sigma_i = \frac{\sum_{j=1}^N u_{ij}^m d_{ij}^2}{\sum_{j=1}^N u_{ij}^m}$. Fig. 3(b)-(d) show the centers and the inlier bounds found by F$C-$MDE for the clean data set and the data sets contaminated with 24% and 52% noise respectively. It is clear that the estimated centers and inlier bounds are quite insensitive to the amount of noise added in this example.

4 The Relational Fuzzy $C-$ Maximal Density Estimators

The F$C-$MDE algorithm was originally formulated to deal with object or feature data only. For relational applications, F$C-$MDE must be extended so that it can work on relational data. In order to formulate the relational dual of FCMDE, we follow the model established by Hathaway et al. [HDB89] when they derived the relational dual of FCM. First, the squared Euclidean distance, $d_{ik}^2 = \|\mathbf{x}_k - \mathbf{c}_i\|^2$, from feature vector $\mathbf{x}_k$ to the center of the i^{th} cluster, $\mathbf{c}_i$, is written in terms of the relation matrix $\mathbf{R}$ as follows:

$$d_{ik}^2 = (\mathbf{R}\mathbf{v}_i)_k - \mathbf{v}_i \mathbf{R} \mathbf{v}_i / 2. \tag{14}$$

Here, $\mathbf{R} = [R_{jk}]$ is the dissimilarity between $\mathbf{x}_j$ and $\mathbf{x}_k$, and $\mathbf{v}_i$ is the membership vector defined by

$$\mathbf{v}_i = \frac{(u_{i1}^m w_{i1}, \ldots, u_{iN}^m w_{iN})^t}{\sum_{j=1}^N u_{ij}^m w_{ij}}. \tag{15}$$

Equation (14) allows the computation of the distance between the data points and cluster prototypes in each iteration when only the relational data, $\mathbf{R}$, are given. Therefore, a relational dual of F$C-$MDE exists for the special case where the object data and relational data satisfy

$$\mathbf{R} = [R_{jk}] = \|\mathbf{x}_j - \mathbf{x}_k\|^2 \tag{16}$$

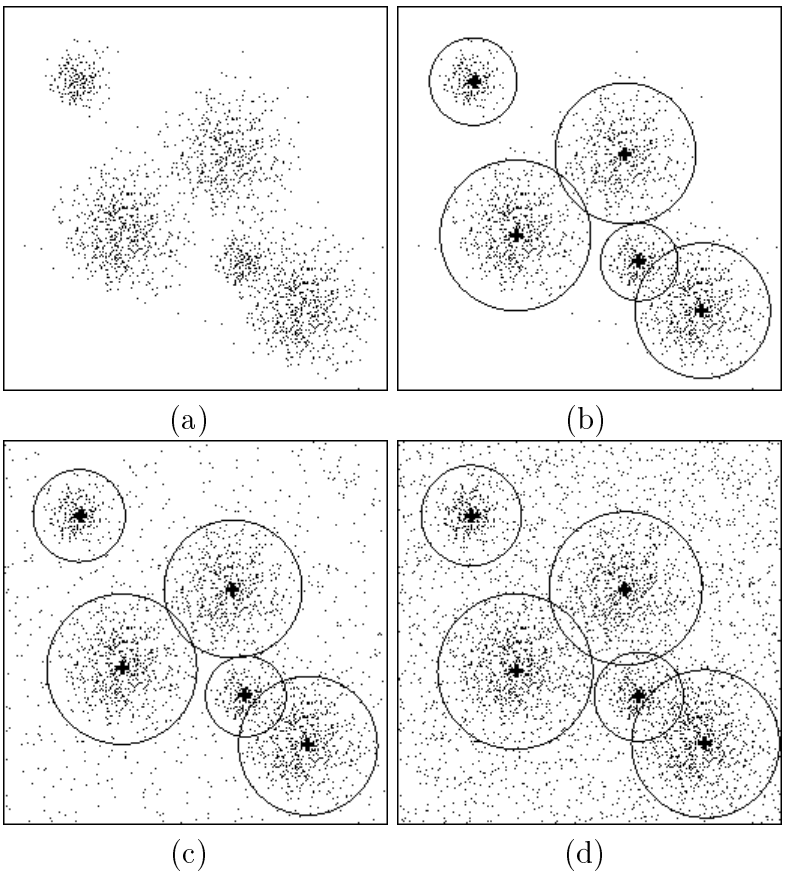

Figure 3: Performance of F$C-$MDE on data with varying noise contamination rates.
(a) Original data set. (b) Result of F$C-$MDE.
(c)Data set in (a) contaminated with 24% uniform noise and result of F$C-$MDE.
(d) Data set in (a) contaminated with 52% uniform noise and result of F$C-$MDE.

This means that even when only relational data is available in the form of an $N \times N$ relation matrix, the relational dual of F$C-$MDE is expected to perform in an equivalent way to the F$C-$MDE provided that the relation matrix, $\mathbf{R}$, is Euclidean, i.e., there exists a set of N points in $\mathcal{R}^{N-1}$, called a realization of $\mathbf{R}$, satisfying (16).

When a realization does not exist for the relation matrix, $\mathbf{R}$, the relational dual of the FCMDE may fail mainly because some of the distances computed using (14) may be negative. To overcome this problem, we use the β-spread transform [HB94] to convert a non-Euclidean matrix $\mathbf{R}$ into an Euclidean Matrix $\mathbf{R}_\beta$ as follows

$$\mathbf{R}_\beta = \mathbf{R} + \beta\,(\mathbf{M} - \mathbf{I}) \tag{17}$$

where β is a suitably chosen scalar, $\mathbf{I} \in \mathcal{R}^{n\times n}$ is the identity matrix and $\mathbf{M} \in \mathcal{R}^{n\times n}$ satisfies $M_{jj} = 1$ for $1 \leq i, j \leq n$. It was suggested in [HB94] that the distances d_{ik}^2 be checked in every iteration for negativity, which indicates a non-Euclidean relation matrix. In that case, the β-spread transform should be applied with a suitable value of β to make the d_{ik}^2 positive again. An underestimate for the lower bound on β was derived [HB94] and related to the necessary shift that is needed to make the distances positive. This result can be summarized as

$$\Delta\beta = \max_{i,k}\{-2d_{ik}^2 / \left\|\mathbf{v}_j - \mathbf{e}_k\right\|^2\}, \tag{18}$$

where $\mathbf{e}_k$ denotes the k^{th} column of the identity matrix. The resulting Relational F$C-$MDE algorithm (RF$C-$MDE) can deal with complex and subjective dissimilarity/similarity measures which are not restricted to be Euclidean or metric.

5 Experimental Results

5.1 Measures for Evaluation of Results

We interpret the results of applying RF$C-$MDE on the user session relational data by using the following quantitative measures. First, the user sessions are assigned to the closest clusters based on the distances computed in (14). This creates C clusters $\mathcal{X}_i = \left\{ \mathbf{s}^{(k)} \in \mathcal{S} \mid d_{ik} < d_{jk} \; \forall j \neq i \right\}$, for $1 \leq i \leq C$. After the crisp assignment of user sessions to the automatically determined number (C) of clusters, the sessions in cluster $\mathcal{X}_i$ are summarized in a typical session "profile" vector $\mathbf{P}_i = \left(P_{i1}, \ldots, P_{i_{N_U}} \right)^t$. The components of $\mathbf{P}_i$ are URL weights which represent the probability of access of each URL during the sessions of $\mathcal{X}_i$ as follows

$$P_{ij} = p\left(\mathbf{s}_j^{(k)} = 1 | \mathbf{s}^{(k)} \in \mathcal{X}_i \right) = \frac{\left| \mathcal{X}_{i_j} \right|}{|\mathcal{X}_i|}, \tag{19}$$

where $\mathcal{X}_{i_j} = \left\{ \mathbf{s}^{(k)} \in \mathcal{X}_i \mid s_j^{(k)} > 0 \right\}$. The URL weights P_{ij} measure the significance of a given URL to the i^{th} profile. Besides summarizing profiles, the components of the profile vector can be used to recognize an invalid profile which has no strong or frequent access pattern. For such a profile, all the URL weights will be low.

Several classical cluster validity measures can be used to assess the goodness of the partition. The intra-cluster or within-cluster distance represents an average of the distances between all pairs of sessions within the the i^{th} cluster, and is given by $\overline{D}_{Wi} = \frac{\sum_{\mathbf{s}^{(k)} \in \mathcal{X}_i} \sum_{\mathbf{s}^{(l)} \in \mathcal{X}_i, l \neq k} d_{kl}^2}{|\mathcal{X}_i|(|\mathcal{X}_i|-1)}$. This is inversely related to the compactness or goodness of a cluster. A good guideline to use when evaluating clusters based on the intra-cluster distances is to compare these values to the total average pairwise distance of all sessions. The latter corresponds to the intra-cluster distance if all the user sessions were assigned to one cluster (i.e., no category information is used). Also it is important to recall that all distances are in $[0, 1]$. The inter-cluster or between-cluster distance represents an average of the distances between sessions from the i^{th} cluster and sessions from the j^{th} cluster, and is given by $\overline{D}_{Bij} = \frac{\sum_{\mathbf{s}^{(k)} \in \mathcal{X}_i} \sum_{\mathbf{s}^{(l)} \in \mathcal{X}_j} d_{kl}^2}{|\mathcal{X}_i||\mathcal{X}_j|}$. For a good partition, the inter-cluster distances should be high because they measure the separation between clusters.

5.2 Examples of Profile Extraction

The web mining procedure described in Section 2 was used to extract typical user session profiles from the log data of the web sites of two departments at two different universities. We first describe the results for the department of Computer Engineering and Computer Sciences at the University of Missouri, Columbia. The log data from accesses to the server during a period of 12 days was used. After filtering out irrelevent entries, the data was segmented into 1703 sessions. The maximum elapsed time between two consecutive accesses in the same session was set to 45 minutes. The number of distinct URLs accessed in valid entries was 369. The initial distance values d_{ik}^2 were obtained by randomly choosing C rows, for $1 \leq i \leq C$, from the relation matrix

and computing the fuzzy memberships using (13), which treats the transactions corresponding to the selected rows as initial prototypes.

After clustering the relational data with RF$C-$MDE and $C = 35$, the sessions were assigned to the clusters in a minimum distance classifier sense. As a result, only 15 clusters that had cardinalities exceeding 30 were kept, hence making a sufficiently strong profile. Table 1 illustrates four profiles computed using (19), where only the significant URLs ($P_{ij} \geq 0.15$) are displayed, and the individual components are displayed in the format $\{P_{ij}$ - j^{th} URL$\}$. The sessions were assigned to the closest cluster and the session clusters or profiles were examined qualitatively. The results are summarized in Table 2, which also lists the cardinality and the intra-cluster distance for all clusters.

Table 1: Examples of Profiles Discovered by RF$C-$MDE from Missouri Data

i	$\mathbf{P}_i$
4	{.78 - /cecs_computer.class} {.96 - /courses.html} {.99 - /courses_index.html} {.96 - /courses100.html} {.34 - /courses300.html} {.2 - /courses_webpg.html} {.28 - /courses200.html} {.87 - /}
6	{.60 - /~joshi/courses/cecs352} {.29 - /~joshi/courses/cecs352/slides-index.html} {.18 - /~joshi/courses/cecs352/text.html} {.18 - /~joshi/courses/cecs352/handout.html} {.19 - /~joshi/courses/cecs352/outline.html} {.17 - /~joshi/courses/cecs352/environment}
7	{.47 - /cecs_computer.class} {0.85 - /degrees_undergrad.html} {0.84 - /degrees_index.html} {0.85 - /bsce.html} {0.46 - /bscs.html} {0.32 - /bacs.html} {0.31 - /courses.html} {0.31 - /courses_index.html} {0.31 - /courses100.html} {0.21 - /courses300.html} {0.18 - /courses200.html} {0.84 - /degrees.html} {0.22 - /general.html} {0.22 - /general_index.html} {0.22 - /facts.html} {0.16 - /research.html} {0.65 - /}
13	{.18 - /~shi} {.64 - /~shi/cecs345} {.38 - /~shi/cecs345/java_examples} {.19 - /~shi/cecs345/references.html} {.20 - /~shi/cecs345/Lectures/05.html} {.20 - /~shi/cecs345/Lectures/06.html} {.29 - /~shi/cecs345/Lectures/07.html} {.18 - /~shi/cecs345/Projects/1.html}

The results show that RF$C-$MDE succeeded in delineating many different profiles in the user sessions. Except for the 14^{th} cluster, all clusters correspond to real profiles reflecting distinct user interests. The profiles followed the access patterns of typical users – the general "outside visitor" is captured in profiles 5 and 12, prospective students in profile 4 and 9, students in CECS352 in profile 6, etc. The goodness of these clusters is recognizable through their low intra-cluster distances (considerably lower than the total average pairwise session distance of 0.9), and their high inter-cluster distances (the majority between .9 and 1). Note that many sessions that do not belong to any profile are lumped in the 14^{th} profile which is easily recognized as a spurious cluster by using the quantitative evaluation measures. In fact, this particular cluster had no significant URLs ($P_{ij} < 0.15$ for all j) and its intra-cluster distance (0.92) was even higher than the total average pairwise distance of all sessions. In addition to the intra-cluster distance, the

Table 2: Summary of RF$C-$MDE User Session Profiles for Missouri Data

i	$\lvert\mathcal{X}_i\rvert$	description	$\overline{D}_{Wi}$
1	56	Dr. Shang's course pages	0.15
2	33	Access statistics pages	0.1
3	64	Site manager's pages	0.13
4	206	General course inquieries	0.2
5	94	main page, people, faculty, research and degree pages	0.68
6	141	Dr. Joshi's cecs352 course pages	0.24
7	68	Undergraduate degree and general course inquieries	0.3
8	66	Accesses to adminstrator's pages	0.24
9	162	Accesses to the cecs227 class pages	0.25
10	62	Dr Joshi's main page and research papers	0.29
11	64	Sessions combining a variety of Dr Joshi's courses and research pages	0.66
12	191	Short sessions mostly limited to main page and class list	0.26
13	142	Dr Shi's cecs345 general course enquieries	0.35
14	119	Mixture of unrelated accesses that don't make a strong profile	0.92
15	152	Dr. Saab's course pages	0.55

robust weights, w_{ij}, are extremely useful for extracting the core members of each class, as well as for distinguishing between strong and spurious profiles. In fact, only the core members of each cluster are expected to have a significant robust weight. For example when only sessions with weights exceeding 0.04 are considered, profiles Nos. 5, 14, and 15 end up having less than 10 members, hence making weak profiles. Note that the spurious cluster No. 14 was eliminated by this method as all its members have low robust weights. Also, several sessions were identified as noise with respect to certain profiles because of their low weights. For example, the following two sessions from profile No. 13, having low robust weights, were considered as noise:

{/courses400.html, /people_index.html, /~shi/publications, /~shi/cecs345, /~joshi/scipad}
{/~shi/cecs345, /~saab/cecs303/private, /~saab/cecs303/private/solution,
/~saab/cecs303/private/solution/hw1.html}

As can be seen, these sessions do not have a specific interest in the 13^{th} profile which concerns Dr Shi's pages.

The Non Euclidean Relational Fuzzy $C-$Means (NERF) [HB94] was also used to cluster the sessions relation matrix, and resulted in only 12 significant profiles, as shown in Table 3. RF$C-$MDE fared better than NERF in the sense that the spurious cluster corresponding to the 12^{th} cluster found by NERF (which corresponds to the 14^{th} profile found by RF$C-$MDE) had 339 more sessions. Also, NERF completely missed the clusters corresponding to the 11^{th} and 15^{th} profiles. Moreover, as can be inferred from the higher cardinality and the average intra-cluster distance for comparable profiles (such as the 5^{th} profiles found by RF$C-$MDE and NERF), NERF's clusters tend to contain more irrelevent sessions or noise. Unlike in the case of RF$C-$MDE, we cannot distinguish noise in the profiles obtained by NERF because it has no robust weights. This is because the fuzzy memberships, u_{ij}, are relative, and regardless of whether a data point is noisy, its memberships across all classes always sum to one, i.e., $\sum_{i=1}^{C} u_{ij} = 1 \; \forall j = 1 \cdots N$.

We also note that as a by-product of the clustering process, associations between different URL addresses on a given site can easily be inferred from the resulting robust profiles. In

general, the URLs that are present in the same profile tend to be visited together in the same session. For example, by looking at the 7^{th} profile in Table 1, we can deduce that the URLs making up that profile tend to co-occur or form a large item set, similar to the ones computed by the Apriori algorithm [AS94]. Such item sets applied to mining associations and sequential patterns between web transactions in [MJHS96]. However, unlike the latter, our method mines typical users profiles as a primary goal, and the discovered associations resulting from this profile extraction come with no additional computation.

Table 3: Summary of NERF User Session Profiles for Missouri Data

i	$\lvert\mathcal{X}_i\rvert$	description	$\overline{D}_{Wi}$
1	70	Dr. Shang's pages	0.2
2	34	Access statistics pages	0.12
3	64	Site manager's pages	0.13
4	213	General course inquieries	0.21
5	109	Main page, people, research, and faculty pages	0.78
6	163	Dr. Joshi's cecs352 course pages	0.28
7	69	Undergraduate degree and general course inquieries	0.31
8	66	Accesses to adminstrator's pages	0.24
9	172	Accesses to the cecs227 class pages	0.28
10	80	Dr Joshi's main page and research papers	0.35
11	206	Short sessions mostly limited to main page and class list	0.29
12	458	Mixture of unrelated accesses	0.87

The next experiment consists of extracting typical user session profiles during the late evening from the log data of the web site for the Computer Science department at the University of Maryland, Baltimore County. The log data from accesses to the server during a period of 6 hours in the late evening was used. After filtering out irrelevent entries, the data was segmented into 678 sessions. The maximum elapsed time between two consecutive accesses in the same session was set to 45 minutes. The number of distinct URLs accessed in valid entries was 1681.

After clustering the relational data with RF$C-$MDE and $C = 35$, the sessions were assigned to the clusters in a minimum distance classifier sense. As a result, only 12 clusters that had cardinalities exceeding 10 were kept, i.e., those making a sufficiently strong profile. Table 4 illustrates four profiles computed using (19), where only the significant URLs ($P_{ij} \geq 0.15$) are displayed, and the individual components are displayed in the format $\{P_{ij}$ - j^{th} URL$\}$. The sessions were assigned to the closest cluster and the session clusters or profiles were examined qualitatively. The results are summarized in Table 5, which also lists the cardinality and the intra-cluster distance for all clusters.

The results show that RF$C-$MDE succeeded in delineating many different profiles in the user sessions. Except for the 11^{th} cluster, all clusters correspond to real profiles reflecting distinct user interests. The profiles seem to reflect the intersts of leisurly users that browse the world wide web in the late evening, as most profiles reflect an interest in free internet games offered by some of the department's graduate students (Profiles No. 2, 5, 6, and 8). Also profile No. 10 reflects an interest in a large set of pictures of a famous supermodel, posted on a graduate student's homepage. Note that many sessions that do not belong to any profile are lumped in the 11^{th} profile which is easily recognized as a spurious cluster by using the quantitative evaluation measures. In fact, this particular cluster had no significant URLs ($P_{ij} < 0.15$ for all j) and its intra-cluster distance (0.99) was even higher than the total average pairwise distance of all

Table 4: Examples of Profiles Discovered by RF$C-$MDE from Maryland Data

i	$\mathbf{P}_i$
2	{.25 - /~mshadl1/Other_Links.html} {.75 - /~mshadl1/profiler.html} {.17 - /~mshadl1/Episode_Guide.html}
4	{.24 - /courses/undergraduate/201/fall98/lectures/index.shtml} {.26 - /courses/undergraduate/201/fall98/projects/index.shtml} {.21 - /courses/undergraduate/201/fall98/projects/p4/index.shtml} {.21 - /courses/undergraduate/201/fall98}
8	{.68 - /~sli2/tetris} {.77 - /~sli2/directory.html} {.67 - /~sli2/tetris/content.html} {.19 - /~sli2/cube/content.html}
9	{.45 - /agents} {0.19 - /agents/news}
12	{.98 - /} {.19 - /people/faculty/faculty.shtml}

Table 5: Summary of RF$C-$MDE User Session Profiles for Maryland Data

i	$\lvert\mathcal{X}_i\rvert$	description	$\overline{D}_{Wi}$
1	13	Graduate course and degree enquieries (/www/courses/graduate) and (/www/graduate)	0.78
2	48	Accesses to ~mshadl1 pages (graduate student offering a page about the Hit TV series "Profiler")	0.49
3	20	Accesses to ~sletsc1 pages (a page about model ceramic cottages)	0.64
4	58	Inquieries about undergraduate course No. 201	0.81
5	27	Accesses to ~sli2 pages, particularly (/~sli2/cube) (a free game offered by a graduate student)	0.43
6	20	Accesses to (/ etoton1/games) (a free game offered by aother graduate student)	0.27
7	15	Accesses to / ebert (professor) pages	0.68
8	79	Accesses to ~sli2 pages, particularly (/~sli2/tetris) (another free game offered by a graduate student)	0.55
9	58	Inquieries about agents (/agents) (The Intelligent Software Agents page)	0.71
10	19	Accesses to (/ rmobar/Claudia.html) (Supermodel pictures offered by a graduate student)	0.35
11	200	Mixture of unrelated accesses that don't make a strong profile	0.99
12	42	Short sessions mostly limited to main page and faculty list	0.66

sessions (0.98). In addition to the intra-cluster distance, the robust weights, w_{ij}, are extremely useful for extracting the core members of each class, as well as for distinguishing between strong and spurious profiles. In fact, only the core members of each cluster are expected to have a significant robust weight. For example when only sessions with weights exceeding 0.029 are

considered, only profiles Nos. 5, 6, 8, 9, 10, and 12 still have more than 10 members, hence making a strong profile. Note that the weak profile No. 11 was eliminated as all its members have low robust weight. Also the following session from profile No. 5, having low robust weight, was considered as noise, obviously, because it is not clear that the user's interest relates to that of the 5^{th} profile:

{/~sli2/cube/cube.html, /~evans/Hemingway1.html, /~cef/science96/drift.html}Note that many sessions that do not belong to any profile are lumped in the 11^{th} profile which is easily recognized as a spurious cluster by using the quantitative evaluation measures. In fact, this particular cluster had no significant URLs ($P_{ij} < 0.15$ for all j) and its intra-cluster distance (0.99) was higher than the total average pairwise distance of all sessions (0.98). In addition to the intra-cluster distance, the robust weights, w_{ij}, are extremely useful for extracting the core members of each class, as well as for distinguishing between strong and spurious profiles. Only the core members of each cluster are expected to have significant robust weights. For example, when sessions with weights exceeding 0.029 are considered, only profiles Nos. 5, 6, 8, 9, 10, and 12 still have more than 10 members, hence making a strong profile. Note that the weak profile No. 11 was eliminated as all its members have low robust weight. Also the following session from profile No. 5, having a low robust weight, was considered as noise, because it is not clear that the user's interest relates to that of the 5^{th} profile:

{/~sli2/cube/cube.html, /~evans/Hemingway1.html, /~cef/science96/drift.html}

The Non Euclidean Relational Fuzzy $C-$Means (NERF) [HB94] was also used to cluster the sessions relation matrix, and resulted in only 4 significant profiles, as shown in Table 6. RF$C-$MDE fared better than NERF in the sense that the spurious cluster (No. 3) found by NERF, which corresponds to the 11^{th} profile found by RF$C-$MDE, had 264 more sessions. Also, NERF completely missed the clusters corresponding to the 3^{rd}, 4^{th}, 7^{th}, 8^{th}, 9^{th}, and 10^{th} profiles, and lumped the 2^{nd}, 6^{th} profiles found by RF$C-$MDE into a single cluster (This corresponds to the 3^{rd} profile found by NERF). Moreover, as can be inferred from the higher cardinality and the average intra-cluster distance for comparable profiles (such as the 5^{th} and 1^{st} profiles found by RF$C-$MDE and NERF respectively), NERF's clusters tend to contain more irrelevent sessions or noise.

As in the previous experiment with the University of Missouri data, We note that as a by-product of the clustering process, associations between different URL addresses on the UMBC site can easily be inferred from the resulting robust profiles. In general, the URLs that are present in the same profile tend to be visited together in the same session. For example, by looking at the 8^{th} profile in Table 1, we can deduce that the URLs making up that profile tend to co-occur to form a large item set.

Table 6: Summary of NERF User Session Profiles for Maryland Data

i	$\lvert\mathcal{X}_i\rvert$	description	$\overline{D}_{Wi}$
1	13	Graduate enquieries (/www/courses/graduate) and (/www/graduate)	0.78
2	112	Accesses to ~sli2 pages, particularly (/~sli2/cube)	0.62
3	66	Accesses to (~mshadl1) and (/ etoton1/games)	0.67
4	464	Mixture of unrelated accesses that don't make a strong profile	0.99
5	35	Short sessions mostly limited to main page	0.58

6 Conclusion

In this paper, we have presented a new approach for automatic discovery of user session profiles in web log data. We defined the notion of a "user session" as being a temporally compact sequence of web accesses by a user. A new similarity measure to analyze session profiles is presented which captures both the individual URLs in a profile as well as the structure of the site. The RF$C-$MDE algorithm was successfully used to cluster the sessions extracted from real server access logs into typical user session profiles, and even to identify the noisy sessions and profiles. The resulting clusters are evaluated subjectively, as well as based on standard statistical criteria. The criteria used were the intra-cluster and inter-cluster distances, and the significance of the components of a session "profile" vector which also summarizes the typical sessions in each cluster. Our experiments on two real data sets showed that RF$C-$MDE clearly outperformed NERF in extracting profiles from the web access logs, and in identifying noisy sessions.

We also note that as a by-product of our clustering process, associations between different URL addresses on a given site can easily be inferred from the resulting robust profiles. In general, the URLs that are present in the same profile tend to be visited together in the same session or form a large item set, similar to the ones in [AS94] [MJHS96]. In our case, item sets are discovered as a byproduct.

Note that in some applications, the frequency of accesses to a web page within the same session may be important. In that case, the definition of $\mathbf{s}_j^{(k)}$ should be modified to $\mathbf{s}_j^{(k)} =$ Number of times the j^{th} URL is accessed in the k^{th} session. In ongoing experiments, we are exploring the formalization of the discovery of meaningful associations between distinct URLs in the extracted profiles. We are also studying a multi-resolution profiling approach where clustering is applied recursively on the profiles found in previous runs. This approach has the potential of providing user session profiles that match any desired level of detail. We are extending this approach to incrementally update the profiles as more access data is logged by the server, instead of reapplying the entire web mining process on the whole data. This will considerably reduce the time requirements of our approach which we next plan to test on larger data sets such as those made available to us by Lycos. We are also looking into a multi-resolution profiling approach where clustering is applied recursively on the profiles found in previous runs. This approach has the potential of providing user session profiles that match any desired level of detail.

The main computational cost in our approach comes from the preliminary computation of the relation matrix which is $\mathcal{O}(N^2)$. The computational load can be reduced considerably if the computations of several blocks of the relation matrix are performed in parallel.

Acknowledgments

This work was partially supported by cooperative NSF awards (IIS 9801711 and IIS 9800899) to Joshi and Krishnapuram respectively, and an IBM faculty development award to A. Joshi). We also thank Lycos for making their web logs available to us.

References

[AFJM95] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell. Webwatcher: A learning apprentice for the world wide web. In *AAAI Spring Symposium on Information Gathering from Heterogenous, Distributed Environment*, Mar. 1995.

[AJ98] P. Karnam A. Joshi, C. Punyapu. Personalization and asynchronicity to support mobile web access. In *Workshop on Web Information and Data Management, ACM 7^{th} Intl. Conf. on Information and Knowledge Management*, Nov. 1998.

[AM98] G. Arocena and A. Mendelzon. Weboql: Restructuring documents, databases, and webs. In *IEEE Intl. Conf. Data Engineering '98*, Feb. 1998.

[AS94] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *20th VLDB Conference*, pages 487–499, Santiago, Chile, 1994.

[Bez81] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[Fir] Firefly. *http://www.firefly.com*.

[FPSSU96] U. Fayad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. *, ed. Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.

[GK79] E. E. Gustafson and W. C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *IEEE CDC*, pages 761–766, San Diego, California, 1979.

[HB94] R. J. Hathaway and J. C. Bezdek. Nerf c-means: Non-euclidean relational fuzzy clustering. *Pattern Recognition*, 27(3):429–437, 1994.

[HDB89] R. J. Hathaway, J. W. Davenport, and J. C. Bezdek. Relational duals of the c-means algorithms. *Pattern Recognition*, 22:205–212, 1989.

[JWH97] A. Joshi, S. Weerawarana, and E. Houstis. On disconnected browsing of distributed information. In *Seventh IEEE Intl. Workshop on Research Issues in Data Engineering (RIDE)*, pages 101–108, 1997.

[KJ78] R. Koenker and G. Basset Jr. Regression quantiles. *Econometrica*, 36:33–50, 1978.

[KJ98] R. Krishnapuram and A. Joshi. Robust clustering methods to support web mining. In *1998 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 15:1–15:8, Seattle, Jun. 1998.

[MJHS96] B. Mobasher, N. Jain, E-H. Han, and J. Srivastava. Web mining: Pattern discovery from world wide web transactions. Technical Report 96-050, University of Minnesota, Sep. 1996.

[NK95] O. Nasraoui and R. Krishnapuram. Crisp interpretation of fuzzy and possibilistic clustering algorithms. In *3rd European Congress on Intelligent Techniques and Soft Computing*, volume 3, pages 1312–1318, Syracuse NY, Aug. 1995.

[NK96] O. Nasraoui and R. Krishnapuram. A robust estimator based on density and scale optimization, and its application to clustering. In *IEEE International Conference on Fuzzy Systems*, pages 1031–1035, New Orleans, Sep. 1996.

[OH98] O.Zaiane and J. Han. Webml: Querying the world-wide web for resources and knowledge. In *Workshop on Web Information and Data Management, ACM* 7^{th} *Intl. Conf. on Information and Knowledge Management*, Nov. 1998.

[PE97] M. Perkowitz and O. Etzioni. Adaptive web sites: an ai challenge. In *Intl. Joint Conf. on AI*, 1997.

[PE98] M. Perkowitz and O. Etzioni. Adaptive web sites: Automatically synthesizing web pages. In *AAAI 98*, 1998.

[RL87] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection.* John Wiliey & Sons, New York, 1987.

[Rou84] P. J. Rousseeuw. Least median of squares regression. *J. Amer. Stat. Assoc*, 79:871–880, 1984.

[Rus69] E. Ruspini. A new approach to clustering. *Information Control*, 15:22–32, 1969.

[SZAS97] C. Shahabe, A. M. Zarkesh, J. Abidi, and V. Shah. Knowledge discovery from user's web-page navigation. In *Seventh IEEE Intl. Workshop on Research Issues in Data Engineering (RIDE)*, pages 20–29, 1997.

[THA97] L. Terveen, W. Hill, and B. Amento. Phoaks - a system for sharing recommendations. *Comm. ACM*, 40(3), 1997.