# Mining Web Access Logs Using Relational Competitive Fuzzy Clustering


Olfa Nasraoui
Comp. Engg. & Comp. Sc.
University of Missouri
Columbia, MO 65211
olfa@ece.missouiri.edu

Hichem Frigui
Electrical Engg.
University of Memphis
Memphis, TN 38152
hfrigui@memphis.edu

Anupam Joshi
Comp. Sc. & Elec. Engg.
University of Maryland
Baltimore, MD 21250
joshi@cs.umbc.edu

Raghu Krishnapuram
Math & Comp. Sc.
Colorado School of Mines
Golden, CO 80401
rkrishna@mines.edu



## Abstract

*The proliferation of information on the World-Wide Web has made the personalization of this information space a necessity. An important part of Web personalization is to mine typical user profiles from the vast amount of historical data stored in access logs. In this paper, we define the notion of a "user session" and a new distance measure between two web sessions that captures the organization of a web site. A competitive agglomeration clustering algorithm which can automatically cluster data into a parsimonious number of components is used to analyze server access logs and obtain typical session profiles of users.*


## 1 Introduction

Personalization is a recent and informally-articulated notion, and deals with tailoring a user's interaction with the Web information space based on information about him/her. For example, a person in Switzerland searching for ski resorts is likely to be interested in the Alps, whereas a person in Colorado is likely to be interested in the Rockies. Personalization can either be done via search engines such as Lycos, or by making Web sites adaptive. Examples are Firefly [1], $W^3IQ$ [2], PHOAKS [3] and [4]. Mining user profiles from vast amounts of historical data stored in server/access logs is a possible approach to personalization that has been recently proposed [5, 6, 7]. 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. In this light, web mining can be viewed as a special case of the more general problem of knowledge discovery in databases [8]. We define the notion of a "user session" as being a temporally compact sequence of web accesses by a user, and 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.

Categories in most data mining tasks are rarely well separated, and hence, the class partition is best described by fuzzy memberships [9]. We use our fuzzy Competitive Agglomeration (CA) algorithm [10] which can automatically cluster data into the optimal number of components. However, CA deals with object or feature data only, whereas session similarity data is relational. Moreover, the session dissimilarity measure we define is not Euclidean. Therefore, we extend the CA so that it can work on non-Euclidean relational data. The resulting Competitive Agglomeration for Relational Data (CARD) algorithm can deal with complex and subjective distance/similarity measures which are not restricted to be Euclidean.

## 2 Defining Similarity Between User Sessions

### 2.1 Preprocessing and Segmentation of the access log data into sessions

Each access 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 protocol (typically HTTP/1.0), (vi) Return code, (vii) 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 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 [6], 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 elapsed time between 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

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

The ensemble of all $N_S$ sessions extracted from the server log file is denoted $\mathcal{S}$. 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". This notion of multiple user sessions enable us to better capture the situation when the same user displays a few (different) access patterns on this site.

## 2.2 Adaptation of Session Data to Clustering

Clustering algorithms based on object data are not suitable for clustering user sessions because of the high dimensionality of the feaure space (there are usually several hundred URLs in a typical web site). 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, and adopt a relational approach to clustering since our data (sessions) are not numeric in nature. We start by considering the cosine of the angle between $\mathbf{s}^{(k)}$ and $\mathbf{s}^{(l)}$ as a measure of similarity

$$S_{1,kl} = \frac{\sum_{i=1}^{Nu} s_i^{(k)} s_i^{(l)}}{\sqrt{\sum_{i=1}^{Nu} s_i^{(k)}}\sqrt{\sum_{i=1}^{Nu} s_i^{(l)}}} \quad (1)$$

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 $S_1$. 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 first provide a syntactic model for the entire web site as a tree where 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 highest level URL in the web site (/). Taking into account the syntactic representation of two URLs, we define the "syntactic" similarity between the $i^{th}$ and $j^{th}$ URLs as

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

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. 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:

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

Unlike $S_1$, this similarity uses soft URL level similarities. For the special case when all the URLs accessed during session $\mathbf{s}^{(k)}$ have zero similarity with the URLs accessed during session $\mathbf{s}^{(l)}$, 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 identical, 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:

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

For the purpose of relational clustering, this similarity is mapped to the dissimilarity measure $d_s^2(k,l) = (1 - S_{kl})^2$.

However, unlike a metric distance it is possible for two distinct sessions to have zero dissimilarity. This occurs whenever the numerator and denominator of (3) are equal, 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 Clustering the User Sessions Using CARD

## 3.1 The Competitive Agglomeration algorithm

The Competitive Agglomeration (CA) algorithm [10] starts by partitioning the data set into a large number of small clusters. As the algorithm progresses, adjacent clusters compete for data points, and clusters that lose in the competition gradually become depleted and vanish. The final partition is a parsimonious description of the data. Let $\mathcal{X} = \{\mathbf{x}_j \mid j = 1, \ldots, n\}$ be a set of $n$ vectors and let $\mathbf{B} = (\beta_1, \ldots, \beta_c)$ represent a $c$-tuple of prototypes each of which characterizes one of the $c$ clusters. The CA algorithm minimizes:

$$J(\mathbf{B}, \mathbf{U}) = \sum_{i=1}^{c} \sum_{j=1}^{n} (u_{ij})^2 d_{ij}^2 \; - \; \alpha \sum_{i=1}^{c} \left[ \sum_{j=1}^{n} u_{ij} \right]^2 \quad (5)$$

subject to: $\sum_{i=1}^{c} u_{ij} = 1, \quad \forall j \in \{1, \cdots, n\}$. In (5), $d_{ij}^2$ represents the distance from feature vector $\mathbf{x}_j$ to the prototype $\beta_i$, $u_{ij}$ represents the degree of membership of feature point $\mathbf{x}_j$ in cluster $\beta_i$, and $\mathbf{U} = [u_{ij}]$ is a $c \times n$ constrained fuzzy $c$-partition matrix [9]. The number of clusters $c$ in (5) is dynamically updated in the CA algorithm. The memberships $u_{st}$ that minimize (5) with respect to $\mathbf{U}$ are given by [10]

$$u_{st} = u_{st}^{\mathrm{FCM}} + u_{st}^{\mathrm{Bias}}. \quad (6)$$

In (6), $u_{st}^{\mathrm{FCM}} = \frac{1/d_{st}^2}{\sum_{k=1}^{c} 1/d_{kt}^2}$, and $u_{st}^{\mathrm{Bias}} = \frac{\alpha}{d_{st}^2}\left(n_s - \overline{n}_t\right)$, where $n_s$ is the cardinality of cluster $s$,

$$n_s = \sum_{j=1}^{n} u_{sj}, \quad (7)$$

and $\overline{n}_t = \frac{\sum_{k=1}^{c} 1/d_{kt}^2 n_k}{\sum_{k=1}^{c} 1/d_{kt}^2}$. For clusters with cardinality higher (lower) than average, the bias term is positive (negative), thus appreciating (depreciating) the membership value. This leads to a gradual erosion of the cardinality of spurious clusters. When the cardinality of a cluster drops below a threshold,the cluster is discarded and $c$ is updated. Since the initial partition has an overspecified number of clusters, $c_{max}$, each cluster is approximated by many small clusters in the beginning. As the algorithm proceeds, the second term in (5) causes each cluster to expand, and at the same time, the constraint on the memberships causes adjacent clusters to compete. As a result, only a few clusters will survive, while others will shrink and eventually become extinct.

## 3.2 Extension to Non-Euclidean Relational Data

For our application, the CA must be extended so that it can work on relational data. In order to formulate the relational dual of the Fuzzy C Means (FCM) algorithm, Hathaway et al. [12] proved that the squared Euclidean distance, $d_{ik}^2 = \|\mathbf{x}_j - \mathbf{c}_i\|^2$, from feature vector $\mathbf{x}_j$ to the center of the $i^{th}$ cluster, $\mathbf{c}_i$, can be written in terms of the relation matrix $\mathbf{R}$ as follows:

$$d_{ik}^2 = (\mathbf{R}\mathbf{v}_i)_k - \mathbf{v}_i \mathbf{R} \mathbf{v}_i / 2. \quad (8)$$

where $\mathbf{v}_i$ is the membership vector defined by

$$\mathbf{v}_i = \frac{(u_{i1}^m, \ldots, u_{iN}^m)^t}{\sum_{j=1}^{N} u_{ij}^m}. \quad (9)$$

The value of $m$ is 2 in our application. Equation (8) 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 CA exists for the special case where the object data and relational data satisfy

$$\mathbf{R} = [R_{ij}] = \|\mathbf{x}_j - \mathbf{c}_i\|^2. \quad (10)$$

When a realization satisfying (10) does not exist for the relation matrix, $\mathbf{R}$, the relational dual of the CA may fail mainly because some of the distances computed using (8) may be negative [14]. To overcome this problem, we use the $\beta$-spread transform [13] to convert a non-Euclidean matrix $\mathbf{R}$ into an Euclidean Matrix $\mathbf{R}_\beta$ as follows:

$$\mathbf{R}_\beta = \mathbf{R} + \beta\,(\mathbf{M} - \mathbf{I}) \quad (11)$$

where $\beta$ 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 $1 \le i, j \le n$. It was suggested in [13] that the distances $d_{ik}^2$ be checked in every iteration for negativity, which indicates a non-Euclidean relation matrix. In that case, the $\beta$-spread transform should be applied with a suitable value of $\beta$ to make the $d_{ik}^2$ positive again. An underestimate for the lower bound on $\beta$ was derived [13] and related to the necessary shift that is needed to make the distances positive. This result can be summarized as

$$\Delta\beta = \max_{i,k}\{-2 d_{ik}^2 / \|\mathbf{v}_j - \mathbf{e}_k\|^2\}, \quad (12)$$

where $\mathbf{e}_k$ denotes the $k^{th}$ column of the identity matrix. An annealing schedule is used for $\alpha$ as a function of the iteration number $k$ using the exponential decay defined by

$$\alpha(k) = \eta_0 e^{-k/\tau}, \quad (13)$$

where $\eta_0$ is the initial value and $\tau$ is the time constant.

The resulting CARD algorithm is summarized below:

---

*Competitive Agglomeration For Relational Data (CARD)*
*Fix the maximum number of clusters* $c = c_{max}$;
*Initialize* $k = 0$; $\beta = 0$; $\mathbf{U}^{(0)}$; $n_i$, $0 \le i \le c$ *using (7);*
**Repeat**
  *Compute membership vectors* $\mathbf{v}_i$
    *for* $1 \le i \le c$ *using (9);*
  *Compute* $d_{ik}^2 = (\mathbf{R}_\beta \mathbf{v}_i)_k - \mathbf{v}_i^t \mathbf{R}_\beta \mathbf{v}_i / 2$
    *for* $1 \le i \le c$ *and* $1 \le k \le N_S$;
  *If* ($d_{ik}^2 < 0$ *for any* $i$ *and* $k$) *then* $\{$

*Compute* $\Delta\beta$ *by using (12);*
*Update* $d_{ik}^2 \leftarrow d_{ik}^2 + (\Delta\beta/2) * \|\mathbf{v}_j - \mathbf{e}_k\|^2$
*for* $1 \le i \le c$ *and* $1 \le k \le N_S$*;*
*Update* $\beta = \beta + \Delta\beta$*;*
}
*Update* $\alpha(k)$ *using (13); Update* $\mathbf{U}^{(k)}$ *using (6);*
*Compute* $n_i$ *using (7);*
*If* $(n_i < \epsilon_1)$ *Discard* $i^{th}$ *cluster and update c;*
$k = k + 1$ *;*
**Until** *(memberships stabilize).*

## 4 Interpretation of the Results

The results of applying CARD on the user session relational data are interpreted using the following quantitative measures. First, the user sessions are assigned to the closest clusters based on the distances computed in (8). This creates $c$ clusters $\mathcal{X}_i = \left\{\mathbf{s}^{(k)} \in \mathcal{S} \mid d_{ik} < d_{jk}\ \forall j \ne i\right\}$, for $1 \le i \le c$. The sessions in cluster $\mathcal{X}_i$ are then summarized in a typical session "profile" vector $\mathbf{P}_i = (P_{i1}, \ldots, P_{iN_U})^t$. 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:

$$P_{ij} = p\left(\mathbf{s}_j^{(k)} = 1 | \mathbf{s}_j^{(k)} \in \mathcal{X}_i\right) = \frac{|\mathcal{X}_{i_j}|}{|\mathcal{X}_i|}, \tag{14}$$

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 $\overline{D}_{Wi} = \frac{\sum_{\mathbf{s}^{(k)} \in \mathcal{X}_i} \sum_{\mathbf{s}^{(l)} \in \mathcal{X}_i, l \ne k} d_{kl}^2}{|\mathcal{X}_i|(|\mathcal{X}_i| - 1)}$. 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 $\overline{D}_{Bij} = \frac{\sum_{\mathbf{s}^{(k)} \in \mathcal{X}_i} \sum_{\mathbf{s}^{(l)} \in \mathcal{X}_j, l \ne k} d_{kl}^2}{|\mathcal{X}_i||\mathcal{X}_j|}$. For a good partition, the inter-cluster distances should be high because they measure the separation between clusters.

## 5 Experimental results

The web mining procedure described in Section 2 was used to extract typical user session profiles from the log data of the Web site for the Department of Computer Engineering and Computer Sciences at the University of Missouri-Columbia. Data during a period of 12 days was used. After filtering out irrelevant 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. While applying CARD, a cluster was discarded if its cardinality $(n_i)$ was less than 5. The initial value of $\eta$ $(\eta_0)$ was set to 0.0002, the time constant $\tau$ was set to 10, and $c_{max}$ was chosen to be 50. The initial distance values $d_{ik}^2$ were obtained by randomly choosing $c_{max}$ rows from the relation matrix and computing the fuzzy memberships.

After clustering the relational data with CARD, the final number of clusters was 20. Table 1 illustrates four profiles computed using (14), where only the significant URLs $(P_{ij} > 0.15)$ 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 and are summarized in Table 2 which also lists the cardinality and the intra-cluster distance for some of the clusters.

The results show that CARD succeeded in delineating many different profiles in the user sessions. Except for the 14th cluster, all clusters correspond to real profiles reflecting distinct user interests. The profiles followed the access patterns on typical users – the general "outside visitor" is captured in profiles 1 and 16, prospective students in profile 7, students in CECS438/CECS352 (taught by the same faculty member) in profile 6 etc. A listing of all 20 profiles is not presented here due to paucity of space. The goodness of these clusters is recognizable through their low intra-cluster distances (considerably lower than the total pairwise session distance of 0.88), and their high inter-cluster distances (the majority between .9 and 1). Note that all the remaining sessions that do not belong to any profile are lumped in the $14^{th}$ profile which is easily recognized 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 was very high (0.95), which is even higher than the total average pairwise distance of all sessions.

## 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 Competitive Agglomeration for Relational Data (CARD) algorithm can deal with complex and subjective dissimilarity/similarity measures which are not restricted to be Euclidean. The resulting clusters are evaluated subjectively, as well as based on standard statistical criteria.

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 looking into robust profiling methods which have the advantage of being more resistant to noise, and a multi-resolution profiling approach where clustering is applied recursively on the profiles found in previous runs.


## Acknowledgments

Partial support of this work by an Office of Naval Research Research grant(N00014-96-1-0439) to R. Krishnapuram, and by an IBM faculty development award N00014-96-1-0439) to A. Joshi, is gratefully acknowledged.


## References


[1] Firefly, "http://www.firefly.com"

[2] A. Joshi, S. Weerawarana, and E. Houstis, "On disconnected browsing of distributed information," *Proc. Seventh IEEE Intl. Workshop on Research Issues in Data Engineering (RIDE)*, pp. 101-108, 1997.

[3] L. Terveen, W. Hill, and B. Amento, "PHOAKS - A system for sharing recommendations," *Comm. ACM*, **40**:3, 1997.

[4] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell, "WebWatcher: A learning apprentice for the world wide web," *AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, March, 1995.

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

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

[7] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," *Proc. of the 20th VLDB Conference*, pp. 487-499, Santiago, Chile, 1994.

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

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

[10] H. Frigui and R. Krishnapuram, "Clustering by competitive agglomeration," *Pattern Recognition*, vol. 30, No. 7, pp. 1109-1119, 1997.

[11] K. S. Fu, *Syntactic Methods in Pattern Recognition,* Academic Press, New York, 1974.

[12] R. J. Hathaway, J. W. Davenport and J. C. Bezdek, "Relational duals of the c-means algorithms," *Pattern Recognition*, vol. 22, pp. 205-212, 1989.

[13] R. J. Hathaway and J. C. Bezdek, "NERF c-Means: Non-Euclidean relational fuzzy clustering," *Pattern Recognition*, vol. 27, No. 3, pp. 429-437, 1994.

[14] S. Sen and R. N. Davé, "Clustering of Relational Data Containing Noise and Outliers," *Proc. of FUZZIEEE*, Anchorage, Alaska, May 1998, pp. 1411-1416.


Table 1. Profile Examples

| $i$ | $\mathbf{P}_i$ |
|---|---|
| 5 | {.75 - /cecs_computer.class} {.94 - /courses.html} {.98 - /courses_index.html} {.95 - /courses10.html} {.34 - /courses30.html} {.20 - /courses_webpg.html} {.28 - /courses20.html} {.85 - /} |
| 6 | {.58 - /̃joshi/courses/cecs352} {.27 - /̃joshi/courses/cecs352/slides-index.html} {.20 - /̃joshi/courses/cecs352/text.html} {.18 - /̃joshi/courses/cecs352/handout.html} {.20 - /̃joshi/courses/cecs352/outline.html} {.15 - /̃joshi/courses/cecs438} {.18 - /̃joshi/courses/cecs352/environment.html} {.16 - /̃joshi/courses/cecs352/proj} |
| 15 | {1.0 - /̃lan/cecs353} {.43 - /̃lan/cecs353/assign1.html} {.25 - /̃lan/cecs353/syl.html} {.25 - /̃lan/cecs353/outline.html} {.75 - /̃lan/cecs353/assign2.html} {.18 - /̃lan} |
| 16 | {.15 - /faculty.html} {.17 - /people.html} {.15 - /people_index.html} {1.0 - /} |

Table 2. A subset of user sessions clusters

| $i$ | $\lvert\mathcal{X}_i\rvert$ | description | $\overline{D}_{Wi}$ |
|---|---|---|---|
| 1 | 114 | main page, faculty list, individual faculty pages, research, people and class list | 0.56 |
| 2 | 75 | D1's pages | 0.15 |
| 3 | 32 | access statistics pages | 0.066 |
| 5 | 207 | general course inquiries | 0.18 |
| 6 | 165 | D2's courses cecs352 and cecs 438 | 0.18 |
| 7 | 64 | inquiries about undergraduate degrees and courses | 0.28 |
| 14 | 114 | mixture of unrelated accesses that don't make a strong profile | 0.95 |
| 16 | 45 | main page, faculty list and people | 0.24 |