

One Step Evolutionary Mining of Context Sensitive Associations and Web Navigation Patterns *

O. Nasraoui[†], and *R. Krishnapuram*[‡]

1 Introduction

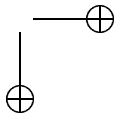
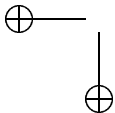
In addition to its ever-expanding size and lack of structure, the World Wide Web has not been responsive to user preferences and interests. Personalization deals with tailoring a user's interaction with the Web information space based on information about him/her, in the same way that a reference librarian uses background knowledge about a person in order to help them better. For example, the phrase "theory of groups" has completely different meanings for a sociologist and a mathematician. In this case, the phrase is the same, while the contexts are different. The concept of *contexts* can be mapped to distinct user *profiles*. Mass profiling is based on general trends of usage patterns (thus protecting privacy) compiled from all users on a site, and can be achieved by mining user profiles from the historical data stored in server access logs.

Recently, data mining techniques have been applied to extract usage patterns from Web log data [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. Of relevance to this paper is our previous work [9, 10, 11] where we have proposed new robust and fuzzy rela-

*Partial support of this work was provided by the National Science Foundation Grant IIS 9800899 to Raghu Krishnapuram and National Science Foundation CAREER Award IIS 0133948 to Olfa Nasraoui.

[†]Department of Electrical and Computer Engineering, 206 Engineering Science Bldg., The University of Memphis, Memphis, TN 38152-3180. E-mail: onasraou@memphis.edu.

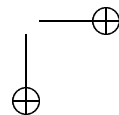
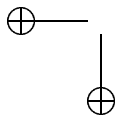
[‡]IBM India Research Lab, Block 1, Indian Institute of Technology, Hauz Khas, New Delhi 110016, India. E-mail: kraghura@in.ibm.com.



tional clustering techniques that allow Web usage clusters to overlap and that can detect and handle outliers in the data set, together with a new subjective similarity measure between two Web sessions, that captures the organization of a Web site, was presented and a mathematical model for “robust” Web user profiles and quantitative evaluation means for their validation. Unfortunately, the computation of a huge relation matrix added a heavy computational and storage burden to the clustering process.

In order to meet Web mining challenges, it is desired for a clustering technique to have the following attributes: *(i) Robustness to noise:* Web data is noisy by nature and a single outlier can completely derail a traditional clustering method. *(ii) Ability to determine the number of clusters/categories automatically:* also known as unsupervised clustering, a notoriously difficult problem known for its high computational costs and sensitivity to noise. *(iii) Ability to yield a multi-resolution categorization of the data:* A hierarchical approach offers a richer description of the data in contrast to the flat view of single-level clustering, and can accelerate the clustering process. *(iv) Insensitivity to initial conditions:* The most computationally efficient clustering techniques such as prototype-based techniques find the prototypes or the partition by local analytical optimization of a criterion function. Hence, they are sensitive to initialization. *(v) Ability to mine only good clusters:* Classical clustering techniques tend to force a structure on all regions of the data space, even where no structure exists. This attribute is referred to as *cluster mining* by Etzioni and Perkowski [5] and is closely intertwined with *robust* clustering as will be seen in this paper. *(vi) Ability to deal with atypical data sets and arbitrary similarity measures:* Current approaches avoid the feature representation dilemma of Web data by resorting to relational clustering or association rule discovery, both carrying a high computational and/or storage burden. A classical non-relational approach requires a differentiable dissimilarity measure. However, for DM problems, a domain specific similarity measure should be designed free of any constraints. *(vii) Efficiency:* Current approaches require the computation of all pairwise similarities (quadratic complexity) or the discovery of all association rules [12] *prior* to discovering user profiles, hence relying on *two relatively expensive* data mining steps. Of particular interest, is the discovery of frequent sets of URLs. Since URL associations tend to occur with very low support in Web log files, this step can become prohibitively expensive. In this paper, we present a *quasi-linear* complexity technique for mining both user profile clusters and URL associations in a *single* step.

Recently [13], we have presented a new evolutionary approach to clustering based on the Unsupervised Niche Clustering algorithm (UNC). Our clustering technique exploits the symbiosis between clusters in feature space and genetic biological niches in nature. UNC seeks dense areas in feature space and determines their number by converting the clustering problem into a multimodal function optimization problem within the context of genetic niching. Genetic Optimization makes UNC much less prone to suboptimal solutions than other objective function based approaches. Additionally, the use of robust weights makes UNC robust in the presence of noise and outliers. However, UNC was formulated for the 2-D case, based on a Euclidean metric space representation of the data.

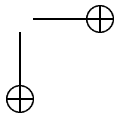
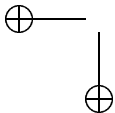


In this paper, we propose a Hierarchical modification of UNC, called H-UNC, that departs from the traditional limited flat view of the data, and generates instead, a hierarchy of clusters which give more insight to the Web mining process, and speeds it up considerably. We use H-UNC as part of a complete system of knowledge discovery in Web usage data. Our new approach does not necessitate fixing the number of clusters in advance, can provide profiles to match any desired level of detail or resolution, and requires no analytical derivation of the prototypes. Thus, it can handle a vast array of general subjective, even non-metric dissimilarities, making it suitable for many applications, particularly in data and Web mining. Our web mining approach also discovers associations between different Web pages based only on the user access patterns or profiles, and not on the Web site page content. These associations are meaningful only within well defined distinct profiles/contexts (*context-sensitive*) as opposed to all or none of the data (context-blind). This approach of discovering context-sensitive associations via clustering can be generalized to other transactional data. In this paper, we also derive interesting quantitative goodness measures for the discovered associations and their relation to profile based URL recommendations.

The remainder of this paper is organized as follows. In Section 2 we review genetic niching methods. In Section 3, we explain our Knowledge Discovery in Web Log files process for web mining, and present quantitative goodness measures for the discovered profiles, associations, and subsequent URL recommendations. In Section 4, we present the Unsupervised Niche Clustering algorithm (UNC). In Section 5, we present the Hierarchical Unsupervised Niche Clustering algorithm (H-UNC), and adapt it to clustering Web sessions. In Section 6, we present our experimental results. Finally, we present our conclusions in Section 7.

2 Genetic Niching

The traditional GA [14] has proved effective in exploring complicated fitness landscapes and converging populations of candidate solutions to a single global optimum. However, some optimization problems require the identification of global as well as local optima in a multimodal domain. As a result, several population diversity mechanisms have been proposed to counteract the convergence of the population to a single solution by maintaining a diverse population of members throughout its search. These methods, known as niching methods [14, 15, 16, 17], were designed to identify multiple optima within multimodal domains. Each peak in a multimodal domain can be thought of as a niche. In nature, niches correspond to different subspaces of the environment that can support different types of life such as species or organisms. The fertility of the niche as well as the efficiency of each organism at exploiting that fertility is what determines the number of organisms that can be contained in a niche. This principle is at the base of how GAs should maintain the population diversity of its members in a multimodal domain. Thus, the niches should be populated in proportion to their fitness relative to other peaks. Mahfoud [17] proposed an improved crowding mechanism, called “deterministic crowding” (DC). After the mating of 2 parents, DC replaces each parent by the most similar



child only if the latter has higher fitness.

3 The Knowledge Discovery Process of Web Session Profiling

3.1 Extracting Web User 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) URL of the page accessed, \dots , etc. A user session consists of accesses originating from the same IP address within a predefined time period. Each URL in the site is assigned a unique number $j \in \{1, \dots, N_U\}$, 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 \mathcal{S} .

3.2 Assessing Web User Session Similarity

The similarity measure between two user-sessions: $\mathbf{s}^{(k)}$ and $\mathbf{s}^{(l)}$ relies on two sub-measures [9, 11]. The first measure which ignores the site structure is given by

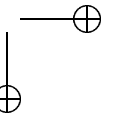
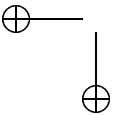
$$S_{1,kl} = \frac{\sum_{i=1}^{N_U} s_i^{(k)} s_i^{(l)}}{\sqrt{\sum_{i=1}^{N_U} s_i^{(k)}} \sqrt{\sum_{i=1}^{N_U} s_i^{(l)}}}. \text{ The second similarity measure requires the pre-}$$

computation of the similarities at the structural URL level that will be used in the computation of the similarity at the session level.

The entire Web site is modeled as a tree with the nodes representing different URL’s. The tree is similar to that of a directory where an edge connects one node to another if the URL corresponding to the latter is hierarchically located under that of the former. The “syntactic” similarity between the i^{th} and j^{th} URLs is defined as $S_u(i, j) = \min\left(1, \frac{|p_i \cap p_j|}{\max(1, \max(|p_i|, |p_j|) - 1)}\right)$, where p_i denotes the path traversed from the root node (main page) to the node corresponding to the i^{th} URL, and $|p_i|$ indicates the length of this path. Note that this similarity which lies in $[0, 1]$ basically measures the amount of overlap between the paths of the two URLs. This overlap is inferred directly from the URL address string by exploiting the one-to-one mapping between the address and the site topology. The pairwise URL similarities should be computed only once offline for a particular Web site prior to any clustering. Now the similarity on the session level which incorporates the syntactic URL similarities

is computed by $S_{2,kl} = \frac{\sum_{i=1}^{N_U} \sum_{j=1}^{N_U} s_i^{(k)} s_j^{(l)} S_u(i, j)}{\sum_{i=1}^{N_U} s_i^{(k)} \sum_{j=1}^{N_U} s_j^{(l)}}. \text{ The final similarity given by a max-}$

imally optimisitic aggregation of $S_{1,kl}$ and $S_{2,kl}$ is $S_{kl} = \max(S_{1,kl}, S_{2,kl})$. Finally, this similarity is mapped to the dissimilarity measure $d_s^2(k, l) = (1 - S_{kl})^2$. One of the desirable properties of this Web session dissimilarity is that it becomes more stringent as the accessed URLs get farther from the root because the amount of



specificity in user accesses increases correspondingly. Our syntactic similarity offers an implicit way to capture the concept hierarchy of the URLs of a Web site while mining the clusters and associations, and can be generalized to other transactional databases.

3.3 Clustering Web User Sessions

The extracted sessions can be clustered using either relational or non-relational clustering. However, the former requires the precomputation of a huge similarity matrix, and the latter requires a method that can handle non-differentiable similarity measures. In Section 5, we will present an evolutionary clustering algorithm that can handle such arbitrary similarity measures.

3.4 Interpretation and Evaluation of the Results

The results of clustering the user session data are interpreted using the following quantitative measures [9]. First, the user sessions are assigned to the closest clusters based on the computed distances, d_{ik} , from the i^{th} cluster to the k^{th} session. 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 $1 \leq i \leq C$.

The sessions in cluster \mathcal{X}_i are summarized by a typical session “profile” vector [9] $\mathbf{P}_i = (P_{i1}, \dots, P_{i_{NU}})^t$. The components of \mathbf{P}_i are URL relevance weights, estimated by the probability of access of each URL during the sessions of \mathcal{X}_i as follows

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

where $\mathcal{X}_{ij} = \left\{ \mathbf{s}^{(k)} \in \mathcal{X}_i \mid s_j^{(k)} > 0 \right\}$. 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.

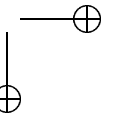
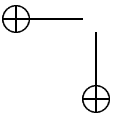
The final prototypes resulting from UNC can be evaluated based on the mean squared error or average dissimilarity, which for the i^{th} cluster, is given by

$$\sigma_i^{*2} = \frac{\sum_{\mathbf{s}^{(k)} \in \mathcal{X}_i} d_{ik}^2}{|\mathcal{X}_i|}. \quad (2)$$

Another measure is the robust cardinality given by

$$N_i^* = \sum_{\mathbf{s}^{(k)} \in \mathcal{X}_i} w_{ik}, \quad (3)$$

where $w_{ik} = \exp \left(-\frac{d_{ik}^2}{2\sigma_i^{*2}} \right)$ is a robust weight (that is high for inliers/good data and low for outliers/noise). Note that the robust cardinality and robust weights can only be exploited when a “robust” clustering method is used to produce the final profiles.



Measures of Goodness of Discovered Itemsets

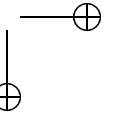
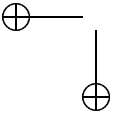
An interesting observation about the weights, w_{ik} , is that they can be used to compute a soft count of the number of transactions that are very similar to a certain profile. As such, the robust cardinalities N_i^* can be viewed as *soft support measures* for the profiles and their corresponding URL associations [12]. That is, the URLs relevant to a particular profile can be considered to form a soft *large itemset* within that *particular context*. Note that the concept hierarchy of the URLs (items) is taken into account via our syntactic Web session similarity. This approach can be generalized to other transactional data. Similarly, the scale or average dissimilarity in each cluster σ_i^{*2} represents a measure of compactness which is inversely related to the *strength* of association between the items in the corresponding large itemset. Therefore, the above measures are quantitative measures of goodness of the clusters/profiles, *as well as* their corresponding large itemsets or inferred URL associations.

A Direct Relation Between the Itemsets Goodness Measures and the Quality of Subsequent Recommendations

It is easy to show, for the simpler case when $S_{kl} = S_{1,kl}$ and $d_s^2(k, l) = (1 - S_{kl})$, that $1 - \sigma_i^{*2}$ simplifies to

$$1 - \sigma_i^{*2} = \frac{\sum_{\mathbf{s}^{(k)} \in \mathcal{X}_i} S_{1,ik}}{|\mathcal{X}_i|}. \quad (4)$$

The simplest way to use the profiles to make recommendations for new Web sessions is to simply recommend the URLs that are significant in the profile nearest to the new session. This can be seen as implementing the rule $\{S_{new} \Rightarrow P_i\}$. The expression for $1 - \sigma_i^{*2}$ in (4) can be seen as the average of the strengths of associations between all sessions assigned to the i^{th} cluster and profile P_i . In fact since the similarity is correlation based, $1 - \sigma_i^{*2}$ is an aggregate measure of the *lift* of all association rules associating sessions in the i^{th} cluster to the *pseudo-session* consisting of the significant URLs in profile P_i . This is a desirable measure to have because the *lift* is not prone to the weaknesses of the *confidence* measure, and because it justifies and provides goodness measures of the subsequent profile based URL recommendations. Most importantly, we should note that the above measures are *not global* to the entire data set. Instead, they are *specific* to one particular cluster or profile of Web sessions, hence enforcing the *context-sensitive* nature of the discovered associations. Though the preceding discussion was from an association rule point of view, we can see that σ_i^{*2} is also a measure of coding error associated with the i^{th} profile when this is seen from a vector quantization point of view (the i^{th} profile is used as a code vector for encoding all vectors in the i^{th} cluster). Similarly, from a 1-NN classifier point of view, σ_i^{*2} can be seen as a measure of classification error.



4 The Unsupervised Niche Clustering Algorithm (UNC)

4.1 Representation and Initialization

The solution space for possible cluster centers consists of n -dimensional prototype vectors. These are represented by concatenating the binary codes of the individual features for one cluster center into a binary string with 8 bits per feature value. The initial centers are selected randomly from the set of feature vectors. This results in a population of N_P individuals, $P_{(i)}$, $i = 1, \dots, N_P$.

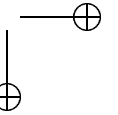
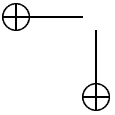
4.2 A Robust Multi-modal Fitness Function

Since in general, we identify dense areas of a feature space as clusters, the fitness value, f_i , for the i th candidate center location, \mathbf{c}_i , is defined as the density of a hypothetical cluster at that location. For the case of 2-dimensional clusters, the density can be defined as

$$f_i = \frac{\sum_{j=1}^N w_{ij}}{\sigma_i^2}, \quad (5)$$

where $w_{ij} = \exp\left(-\frac{d_{ij}^2}{2\sigma_i^2}\right)$ is a robust weight that measures how typical data point \mathbf{x}_j is in the i th cluster, σ_i^2 is a robust measure of scale (dispersion) for the i th cluster, d_{ij}^2 is the distance from data point \mathbf{x}_j to cluster center \mathbf{c}_i , and N is the number of data points. It can easily be seen that as a variance measure, σ_i^2 is also related to the radius of the niche, since in this particular optimization problem each cluster in the data set will generate a niche in the fitness landscape. For the case of n -dimensional Gaussian clusters with variance σ_i^2 , the normalized distance $\frac{d_{ij}^2}{2\sigma_i^2}$ follows a χ^2 distribution with n degrees of freedom. In particular, the probability that a data point lies within a normalized distance of $\chi_{2,\alpha}^2$ from the center is α . The niche radius is defined as that distance, d_{ij}^2 , from the center that encloses a high percentage of the points in that cluster (such as $\alpha = 0.995$) Hence, the niche radius is close to $K\sigma_i^2$, where K is approximately $\chi_{2,0.995}^2$. Note that with reliable estimates of the cluster scales, σ_i^2 , the robust weights w_{ij} approaches 1 when d_{ij}^2 approaches zero (for points close to the cluster center), while asymptotically approaching zero when d_{ij}^2 approaches infinity (for outliers), hence offering a means of distinguishing between good and bad data with respect to every cluster. Moreover, by taking $\frac{\partial f_i}{\partial \mathbf{c}_i} = 0$, we

obtain $\mathbf{c}_i = \frac{\sum_{j=1}^N w_{ij} \mathbf{x}_j}{\sum_{j=1}^N w_{ij}}$. The above two observations lead us to conclude that the objective or fitness function in (5) is expected to be optimal only at the centroid of the cluster, even in the presence of noise, outliers, or more generally any data that does not follow the distribution of the majority of the data in the i th cluster. This means that the fitness measure is robust. Similarly, when a data set contains several clusters, with reliable estimates of the different cluster scales, σ_i^2 , the robust weights w_{ij} will only be high for points that are within the boundaries of the i th



cluster. This means that the landscape of the fitness function in (5) is expected to reach several suboptimal peaks (multiple modes) located at the centroids of these clusters, and their identification is a multi-modal optimization problem.

The scale parameter that maximizes the fitness value for the i^{th} cluster can be found by setting $\frac{\partial f_i}{\partial \sigma_i^2} = 0$ to obtain

$$\sigma_i^2 = \frac{\sum_{j=1}^N w_{ij} d_{ij}^2}{\sum_{j=1}^N w_{ij}}. \quad (6)$$

Therefore, σ_i^2 will be updated using an iterative hill-climbing procedure, using the previous values of σ_i^2 to compute the weights w_{ij} in (6). The scale estimation by Hill climbing makes the entire *hybrid* genetic optimization process converge much faster (typically 10 generations) than a purely genetic search. When mating takes place, each child should inherit the scale parameter, σ_i^2 , of the closest parent as its initial scale before updating.

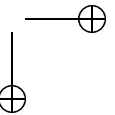
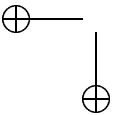
After convergence of the population, the best individual from each good niche is extracted using a greedy approach [13] to obtain the set of final cluster centers, \mathcal{C} .

4.3 Computational Complexity

In each generation, the most extensive computational requirement for UNC consists of computing the residuals, fitness and scale, for each of the N_P individuals in the population, and exactly $N_P/2$ inter-niche distances, resulting in $\mathcal{O}(N_P \cdot N)$ computations. Since the population size tends to be a small fraction of the size of the data set, the complexity is close to linear, and can be further reduced if clustering is performed hierarchically.

5 Hierarchical Unsupervised Niche Clustering and its adaptation to Web Usage Mining

We retain the principal structure of UNC presented in Section 4, except for a few differences that result from the distinct nature of the session data. The solution space for possible session prototypes consists of binary chromosome strings which are defined to be the binary session attribute vectors \mathbf{s}_i defined in Section 3.1. The fitness function remains the same as in (5), except that the Web session dissimilarity measure, defined in Section 3.2, is used instead of the Euclidean distance to take the Web site topology in account. The computational time of genetic optimization can be significantly reduced if we perform clustering in a hierarchical mode. In other words, we could cluster smaller subsets of the data using a smaller population size at multiple levels, instead of clustering the entire data set on a single level which would necessitate a larger population size. The computational complexity of UNC is $\mathcal{O}(N_P \cdot N)$, where N_P is the population size and N is the number of data points to be clustered. Since, in the hierarchical mode, N_P can usually be a very small fraction of N (typical example from our experiments: $\frac{1}{10000}$ to $\frac{1}{1000}$, this complexity



is much lower than that of relational clustering techniques such as Agglomerative Hierarchical Clustering (AHC) [18], $\mathcal{O}(N^2 \log N)$ and the closely related graph theoretic based Minimum Spanning Tree (MST) [18], $\mathcal{O}(N^2)$.

The hierarchical clustering is performed recursively starting from the top level (lowest resolution) until a termination criterion, based on the minimum acceptable size of a cluster, N_{split} , and its maximum allowable mean squared error, σ_{split}^2 , is met. Let l denote the current level. Let $\mathcal{X}_{(l-1)} = \mathcal{X}_{(l-1)_1} \cup \dots \cup \mathcal{X}_{(l-1)_{|C_{(l-1)}|}}$ denote the data set partitioned at level $l-1$, where $\mathcal{X}_{(l-1)_i}$ is the i th cluster found at level $l-1$. Let $C_{(l-1)}$ denote the list of prototypes inducing the above partition, and let $\Sigma_{(l-1)} = \{\sigma_{(l-1)_1}^{*2}, \dots, \sigma_{(l-1)_{|C_{(l-1)}|}}^{*2}\}$ denote the set of mean squared errors computed for each subset of the above partition, using (2). The hierarchical clustering procedure using UNC for Web mining is given below.

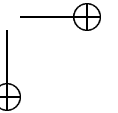
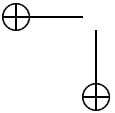
Hierarchical clustering using UNC (H-UNC algorithm)

Fix population size, number of generations, maximum number of levels (L);
Set starting level $l = 1$;
Set initial number of clusters $|C_{(l-1)}| = 1$;
Set initial data set to be clustered $\mathcal{X}_{(l-1)} = \mathcal{X}_{(l-1)_1} = \mathcal{X}$;
Set initial set of prototypes $C_{(l-1)} = \emptyset$;
Initialize final list of prototypes $\mathcal{P} = \emptyset$;
Initialize the set of mean squared errors $\Sigma_{(l-1)} = \{\sigma_{(l-1)_1}^{*2} = 1\}$;
Cluster_Recursively ($\mathcal{X}_{(l-1)}$, $C_{(l-1)}$, $\Sigma_{(l-1)}$, l);
Assign all data points in \mathcal{X} to closest prototype $P_i \in \mathcal{P}$;
Recompute σ_i^{*2} using (2) and N_i^* using (3);

The procedure *Cluster_Recursively* () is given below.

Cluster_Recursively (\mathcal{X}_{l-1} , C_{l-1} , Σ_{l-1} , l)

FOR $i = 1$ **TO** $|C_{l-1}|$ **DO** { /* Each prototype in C_l */
 IF ($l = 1$) or ($|\mathcal{X}_{(l-1)_i}| > N_{split}$ and $\sigma_{(l-1)_i}^{*2} > \sigma_{split}^2$ and $l \leq L$)
 THEN
 {
 Perform UNC clustering on data subset $\mathcal{X}_{(l-1)_i}$;
 /* will result in extracted prototypes set, $C_{(l)_i}$,
 partitioned data set $\mathcal{X}_{l_i} = \mathcal{X}_{l_1} \cup \dots \cup \mathcal{X}_{l_{|C_l|}}$,
 and $\Sigma_{l_i} = \{\sigma_{l_1}^{*2}, \dots, \sigma_{l_{|C_l|}}^{*2}\}$ computed for
 each subset of this partition */
 Cluster_Recursively (\mathcal{X}_{l_i} , C_{l_i} , Σ_{l_i} , $l + 1$);
 }
 ELSE {
 Add i^{th} prototype to final list of prototypes, \mathcal{P} ;
 }
}



5.1 Ease of Setting the Parameters

For the case of Web session clustering, all session dissimilarities are confined in $[0, 1]$. Hence, it is easy to set the values of parameters σ_{split}^2 , and N_{split} , especially in an interactive mode. As a rule of thumb, σ_{split}^2 should be the largest tolerable dissimilarity between sessions considered to be in the same cluster, and N_{split} should be the minimum size of an acceptable cluster or profile. Even though the above parameters will eventually determine the number of clusters at the last level of the partitioning, they are not crucial to the performance of H-UNC because first of all if the partitioning is done at too many levels of the hierarchy, the final clusters will still be good (only exhibiting higher specificity or resolution). And even with too few levels in the hierarchy, H-UNC is expected to identify as many of the good (maximally dense as per the fitness measure) clusters as possible at that level unlike other approaches that will simply link different clusters, thus inducing erroneous prototypes.

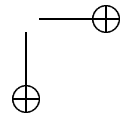
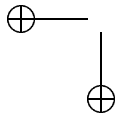
5.2 Comparison with Conventional Hierarchical Clustering

Our approach is substantially different from classical divisive hierarchical clustering techniques, where more clusters are created at increasing levels of a cluster hierarchy. This is because our approach relies on robust weights to suppress the influence of outliers and data belonging to other clusters, and on a multimodal optimization approach where multiple clusters are sought in parallel at each level. This means that at any given level of recursive clustering, even if the population size is too small, H-UNC is expected to identify as many good clusters as the population size, while classical hierarchical approaches are expected to yield the optimal cluster prototypes only at the optimal level of the partition that corresponds to the known correct number of clusters. In fact this is why H-UNC is able to perform well even with anomalously small population sizes that would have never been appropriate with other genetic based approaches. Finally, we note that another difference between H-UNC and classical hierarchical techniques is that the data is re-partitioned at the very end of clustering (the last level of the hierarchy). This means that there is no final commitment of the data at each level. This avoids one of the well known pitfalls of hierarchical clustering techniques, and also allows H-UNC to yield better partitions, and hence better and more accurate profiles for Web mining.

6 Experimental Results

6.1 Synthetic Simulation Results

First, we illustrate UNC (or H-UNC with $L=1$)'s performance on 2-D data sets because the results can be inspected visually and easily. We also compare UNC's results with K-Means [19] and the Possibilistic C-Means algorithm (PCM) [20], a robust clustering algorithm. All three algorithms are initialized with randomly selected centers (for PCM, this is followed by applying the Fuzzy C-Means algorithm and computing the fuzzy average distance to initialize the scale parameters, η). Fig.



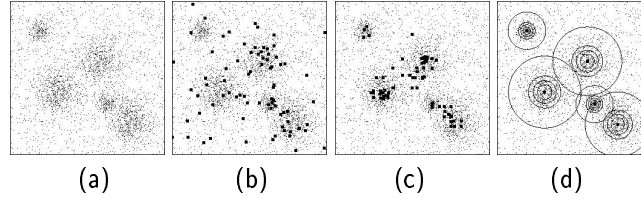


Figure 1. *Evolution of the population: (a) original data (b) Initial population, (c) population after 30 generations, (d) final extracted centers*

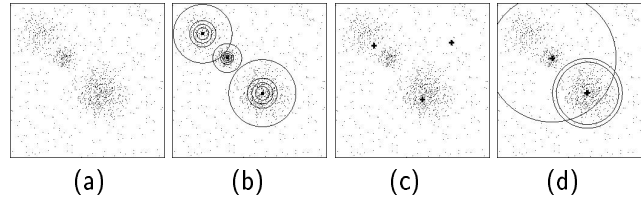
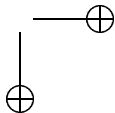
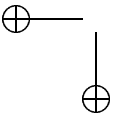


Figure 2. *Three noisy clusters varying in size and density: (a) original data set (b) Results of UNC, (c) Results of K-Means with $C = 3$ (2 clusters are missed), (d) Results of PCM with $C = 3$ (2 clusters are identical, 1 cluster is missed; grossly over-estimated inlier bound indicates lack of robustness)*

1 shows the evolution of the population (denoted by square symbols) using UNC for a noisy data set. The initial population is chosen randomly from the set of feature vectors. This explains the higher concentration of solutions in the densest areas, which converge toward the correct centers in subsequent generations. The cluster centers found using UNC, K-Means, and PCM are shown in Figs. 2 - 3. In these figures, the circular contours around each cluster center depict the normalized distance, $\frac{d_{ij}^2}{\sigma_i^2}$, corresponding to $\chi_{0.995}^2$. The outermost contours are called *inlier bounds*, and reflect the accuracy of the final scale estimates which in turn reflect the *robustness* of clustering. This is because data that falls within the outer contour of a cluster is generally considered to be good/inlier data, while data falling beyond it is considered to be noise/outlier data. No contours are shown for K-Means since it does not estimate scale, and is not robust. Note that in addition to requiring a prespecified number of clusters, C , K-Means and PCM are not as robust as UNC.

6.2 Web Usage Mining Experimental Results

The one day access record Web log was collected in 1998 on the main site at the University of Missouri-Columbia with $N_S = 29,876$ sessions and $N_U = 17,665$ URLs. The parameters for the robust hierarchical UNC were fixed to the following values: The crossover and mutation probabilities are $P_c = 0.9$ and $P_m = 5 \times 10^{-6}$, respectively. UNC used 10 generations per clustering with a population size, $N_P = 20$ and $N_{min} = 10$. Since all session dissimilarities are confined in $[0,1]$, it is reasonable to choose $\sigma_{max}^2 = 0.95$, $\sigma_{split}^2 = 0.3$, and $N_{split} = 50$. Clusters with



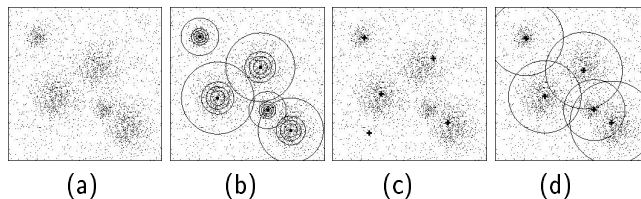


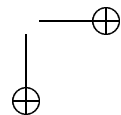
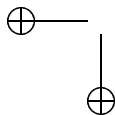
Figure 3. *Five noisy clusters: (a) original data set (b) Results of UNC, (c) Results of K-Means with $C = 5$ (1 cluster missed, another incorrect), (d) Results of PCM with $C = 5$ (over-estimated inlier bounds indicate lack of robustness)*

cardinalities > 20 were considered sufficiently strong profiles. For the profiles, we list the cardinality, $|\mathcal{X}_i|$, core cardinality, $|\mathcal{X}_i^*|$, robust cardinality, N_i^* , and average dissimilarity, σ_i^{*2} (inversely related to aggregate association lift). The symbol “-” in the $|\mathcal{X}_i^*|$ column means that the core of the i^{th} profile contains fewer than 20 sessions. H-UNC succeeded in delineating many real profiles reflecting typical access patterns as seen in Tables 1 – 3 – prospective students in profile 3 at $L=1$, job hunters in profile 9 at $L=2$, game players in profile 11 at $L=2$, humor seekers in profile 18 at $L=3$, in addition to students accessing their course pages, etc. The quality of these clusters (or lift of the associations as discussed in Section 3) is confirmed by their low average dissimilarity compared to the maximal value of 1.

(i) **Robust profiling** is obtained by retaining profile members whose robust weights, w_{ij} , exceed a given threshold, w_{min} , equal to 0.6 in our experiments. This allows us to concentrate on the core of each profile by filtering out the noise sessions which end up being assigned to the closest profile. The w_{min} -core of the i^{th} profile is defined as $\mathcal{X}_i^* = \{\mathbf{s}^{(k)} \in \mathcal{X}_i | w_{ik} > w_{min}\}$. When only sessions with weights exceeding 0.6 are considered, some profiles (e.g. Nos. 11 at level 1) end up having less than 20 members, hence making weak profiles. Also, the core of some clusters, with irrelevant sessions assigned, were discovered to contain sessions with a specific interest. The results will be examined from three different perspectives:

(ii) **Multiresolution profiling:** is done by examining profiles obtained with a varying number of levels, L , in the hierarchy. As L increases, more real profiles emerge. At level 3 (Table 3), most real profiles show a strong attraction to certain students’ homepages, motivated by specific interests: Profiles No. 17 and 24 show interest in popular music bands. Profile No. 26 show interest in actor “Antonio Banderas”. Clearly, these user interests are of a different nature compared to those of profile No. 11 (the *Euler number “e”*) or profile No. 16 (American literature page). The increase in resolution can also be illustrated by the single profile (No. 9) at level 1 (Table 1) which, at level 2 (Table 2) gets split into profiles No. 10, 11, 12, and 13, showing distinct interests in four different sets of pages designed by the same student: GIF animation, lottery games, Euler number “e”, and color blending programs.

(iii) **Inferring Associations between different URLs:** Associations [12] between different URLs can be directly inferred by simple inspection of the robust profile vector components. In general, the relevant URLs can be considered to form



a soft (fuzzy) large itemset. Profile No. 12 in Table 3 shows an association between the URLs (/~c639692/blend.html) and (/~wwwtools/colormaker). Their contents revealed that they shared the same subject (color making tools). Profile No. 15 also shows an association between Web pages belonging to professors in different departments. It turned out that both pages are dedicated to the history of the sinking of the “Titanic”. *Note how the above association were discovered based only on the user access patterns or profiles and not their content.* A lot can be learned about user interests from their access patterns and profile associations. For example, by further examining the individual significant components of the 5th profile vector obtained at $L = 1$ (career interests game based on Dr. Holland’s theory of 6 classes of work environments that fit different people’s personalities) which shows the relevance of pages representing *social*, *entreprising*, *artistic*, *realistic*, and *investigative* work environments respectively, we can deduce that more people tend to identify themselves as *social* (described as *helpers*) than as *entrepreneurial* (described as *persuaders*). Also, the URL corresponding to *conventional* work receives an insignificant weight in this profile. It is not surprising that few people would rather identify themselves with this trait, described in this game page as “People who, carry out tasks in detail or follow through on others’ instructions”.

Table 1. *Some of the profiles discovered by H-UNC at $L = 1$*

i	$ \mathcal{X}_i $	$ \mathcal{X}_i^* $	N_i^*	description	σ_i^{*2}
1	8959	6139	7002.6	main page	0.07
3	104	47	64.1	main page, general information about applying to and living in MU, departments list, admission policy, ... etc	0.71
8	94	42	60.6	Human Resources Services site at MU	0.33
9	158	93	97.5	Accesses to /~c639692 pages: student offering GIF animation archive, lotto games, etc	0.49

7 Conclusion

For real life data mining, the dissimilarity measure may not be a true distance metric, and dealing with relational data is impractical given the huge dimension of the data sets. Therefore, we presented a *quasi-linear* complexity Hierarchical Unsupervised Niche Clustering algorithm (H-UNC), that exploits the symbiosis between clusters in feature space and genetic biological niches in nature. H-UNC was successfully used to cluster the sessions extracted from real server access logs into *multi-resolution* user session profiles, and even to identify the noisy sessions and profiles. We have illustrated that our clustering process results in the discovery of *context-sensitive* associations between different URL addresses on a given site, with no additional cost. 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. We have proposed *qualitative aggregate profile association “lift” measures*, and examined them from the

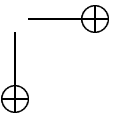
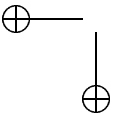
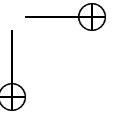
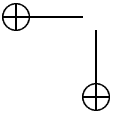


Table 2. *Some of the 15 Profiles discovered by H-UNC at $L = 2$*

i	$ \mathcal{X}_i $	$ \mathcal{X}_i^* $	N_i^*	description	σ_i^{*2}
8	39	24	28.1	Human Resources Services site at MU	0.15
9	389	200	244.9	Accesses to Human Resources Services site (job opportunities, applications, etc) and MU's Employee and Administrative info site	0.31
10	46	29	28.9	Accesses to /~c639692/concise pages: student offering GIF-89 animation archive and info for Web pages)	0.26
11	42	-	27.1	Accesses to /~c639692/lotto.html: same student offering lotto playing games	0.1
12	74	51	50.2	Accesses to /~c639692/exp pages (about the euler number "e")	0.32
13	32	21	22.3	Accesses to /~c639692/blend.html: student offering color blending program and to /~wwwtools/colormaker (MU site containing color making tools in Java)	0.1

Table 3. *A sample of the 34 profiles discovered by H-UNC at $L = 3$*

i	$ \mathcal{X}_i $	$ \mathcal{X}_i^* $	N_i^*	description	σ_i^{*2}
12	32	21	22.3	Accesses to /~c639692/blend.html (student offering color blending program) and to /~wwwtools/colormaker (MU site containing color making tools in Java)	0.1
15	33	21	21.9	Accesses to /~journsww/alex/titanic.htm and /~socbrent/titanic.htm (pages dedicated to the Titanic by journalism and sociology professors)	0.2
16	949	542	611.5	Accesses to /~engmo/amlit.html: English professor's American literature page	0.28
17	226	86	140.4	Accesses to /~c641644 pages: student dedicating page to music group Nirvana	0.29
18	153	112	122.7	Accesses to /~c717733/funnies (student offering jokes' page)	0.08
24	286	154	195.4	Accesses to /~c690403/dmb pages (student dedicating page to music band)	0.14
26	222	172	150.5	Accesses to /~c617756 pages (student dedicating page to actor Antonio Banderas)	0.1

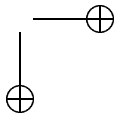
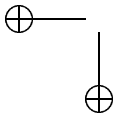


points of view of coding theory, classification, and association rules. We have also related the measures to expected future URL recommendation accuracy. Knowledge about associations between different URLs on a given Web site can be used to improve the design of that Web site and to better understand users' behaviors and their access patterns. Our web mining approach discovers associations between different Web pages based *only* on the user access patterns or profiles, and not on the page content. Also, the associations are meaningful only within well defined distinct profiles/contexts (context-sensitive) as opposed to all or none of the data (context-blind). This approach of discovering context-sensitive associations via clustering can be generalized to other transactional data.

Because of its hierarchical nature and very low population size requirement, H-UNC is significantly faster than UNC for large data sets. H-UNC inherits from Genetic Algorithms their implicit parallelism which makes it a relatively easy candidate for parallelization efforts that can make it even faster. For more general data mining applications, our approach to genetic clustering has the following advantages over previous methods: (i) It is insensitive to initialization and *robust* in the presence of outliers and noise. Hence, it can cope with missing/corrupted data and preprocessing errors more gracefully than non-robust techniques. ; (ii) it can *automatically determine the number of clusters*; (iii) because of the single cluster representation scheme used, *the size of the the search space does not increase with the number of clusters or the number of data*. (iv) It is *generic* enough that it can handle any type of distance/dissimilarity measure and any type of input data regardless of the type of preprocessing (*crucial for data and Web mining*). (v) it offers the advantage of *multi-resolution* clustering/profiling. (vi) It can easily be made scalable by continuously mining portions of the data instead of loading the entire data set in memory. The structure of the algorithm does not change. Only the extraction procedure, after each mining step, has to take into account all the cluster prototypes/representatives discovered so far when extracting the niche peaks. This will automatically merge re-discovered profiles/clusters with old ones that are similar, and add newly discovered profiles/clusters to the final list. Unlike *Lamarckian learning* [21], our dynamic approach to estimate the scale mathematically during genetic optimization of the cluster representatives *does not disrupt the genotype* of the candidate solutions. However, it improves *individual learning* in the evolutionary process by dynamically modifying the *fitness landscape* in a way that will make it easier to maintain diversity and to converge closer to the niche peaks. This can be seen as introducing a *Baldwin effect* [22] to hybridize evolutionary process. We are currently investigating different approaches to make our approach scalable to large data sets, using it for the unsupervised categorization of large text corpuses, as well as experimenting with different recommendation approaches to achieve evolutionary Web personalization.

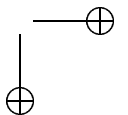
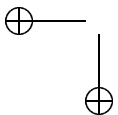
Acknowledgment

Partial support of this work by the National Science Foundation Grant IIS 9800899 to Raghu Krishnapuram and National Science Foundation CAREER Award IIS 0133948 to Olfa Nasraoui is gratefully acknowledged.



Bibliography

- [1] M. Spiliopoulou and L. C. Faulstich, "Wum: A web utilization miner," in *Proceedings of EDBT workshop WebDB98*, Valencia, Spain, 1999.
- [2] R. Cooley, B. Mobasher, and J. Srivastava, "Data preparation for mining world wide web browsing patterns," *Journal of knowledge and information systems*, vol. 1, no. 1, 1999.
- [3] A. Buchner and M. D. Mulvenna, "Discovering internet marketing intelligence through online analytical web usage mining," *SIGMOD Record*, vol. 4, no. 27, 1999.
- [4] O. Zaiane, M. Xin, and J. Han, "Discovering web access patterns and trends by applying olap and data mining technology on web logs," in *Advances in Digital Libraries*, Santa Barbara, CA, 1998, pp. 19–29.
- [5] M. Perkowitz and O. Etzioni, "Adaptive web sites: Automatically synthesizing web pages," in *AAAI 98*, 1998.
- [6] B. Mobasher, N. Jain, E-H. Han, and J. Srivastava, "Web mining: Pattern discovery from world wide web transactions," Tech. Rep. 96-050, University of Minnesota, Sep. 1996.
- [7] C. Shahabi, A. M. Zarkesh, J. Abidi, and V. Shah, "Knowledge discovery from users web-page navigation," in *Proceedings of workshop on research issues in Data engineering*, Birmingham, England, 1997.
- [8] T. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal, "From user access patterns to dynamic hypertext linking," in *Proceedings of the 5th International World Wide Web conference*, Paris, France, 1996.
- [9] O. Nasraoui, R. Krishnapuram, and A. Joshi, "Mining web access logs using a relational clustering algorithm based on a robust estimator," in *NAFIPS Conference*, New York, NY, Jun. 1999, pp. 705–709.
- [10] O. Nasraoui, H. Frigui, R. Krishnapuram, and A. Joshi, "Mining web access logs using relational competitive fuzzy clustering," in *Eighth International Fuzzy Systems Association Congress*, Hsinchu, Taiwan, Aug. 1999.



- [11] O. Nasraoui, R. Krishnapuram, H. Frigui, and Joshi A., “Extracting web user profiles using relational competitive fuzzy clustering,” *To appear in International Journal of Artificial Intelligence*, 2000.
- [12] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in *20th VLDB Conference*, Santiago, Chile, 1994, pp. 487–499.
- [13] O. Nasraoui and R. Krishnapuram, “A novel approach to unsupervised robust clustering using genetic niching,” in *Ninth IEEE International Conference on Fuzzy Systems*, San Antonio, TX, May 2000, pp. 170–175.
- [14] J. H. Holland, *Adaptation in natural and artificial systems*, MIT Press, 1975.
- [15] K. A. De Jong, “An analysis of the behavior of a class of genetic adaptive systems,” *Doct. Diss., U. of Michigan.*, vol. 36, no. 10-5140B, pp. 29–60, 1975.
- [16] D. E. Goldberg and J. J. Richardson, “Genetic algorithms with sharing for multimodal function optimization,” in *2nd Intl. Conf. Genetic Algorithms*, Cambridge, MA, Jul. 1987, pp. 41–49.
- [17] S. W. Mahfoud, “Crowding and preselection revisited,” in *2nd Conf. Parallel problem Solving from Nature, PPSN '92*, Brussels, Belgium, Sep. 1992.
- [18] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, Wiley Interscience, NY, 1973.
- [19] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Fifth Berkeley Symp. on Math. Statist. and Prob.*, Berkeley, California, 1967, pp. 281–297, University of California Press.
- [20] R. Krishnapuram and J. M. Keller, “A possibilistic approach to clustering,” *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 98–110, May 1993.
- [21] D. Whitley, S. Gordon, and K. Mathias, “Lamarckian evolution, the baldwin effect and function optimization,” in *Parallel Problem Solving From Nature-PPSN III*, Y. Davidor, H. Schwefel, and R. Manner, Eds., pp. 6–15. Springer Verlag, 1994.
- [22] G. Hinton and D. Whitley, “How learning can guide evolution,” *Complex Systems*, vol. 1, pp. 495–502, 1987.

