Robust Fuzzy Clustering Methods to Support Web Mining*

Anupam Joshi
Department of Computer Engineering and Computer Science
University of Missouri
Columbia, MO 65211
joshi@cecs.missouri.edu

Raghu Krishnapuram
Department of Mathematical and Computer Sciences
Colorado School of Mines
Golden, CO 80403
rkrishna@mines.edu

1. Introduction

The evolution of the Internet into the Global Information Infrastructure has lead to an explosion in the amount of available information. The Web, then, is becoming the apocryphal Vox Populi, and represents an instance of a semi-structured Distributed Knowledge Environment [27]. Realizing the vision of distributed knowledge access in this scenario and its future evolution will need tools to "personalize" the information space. If one were to look at the Web as a distributed, heterogeneous information base, Web personalization amounts to creating a system which responds to user queries in a manner which is dependent on who the user is. As a trivial example - a biologist querying on cricket in all likelihood wants something other than a sports enthusiast would.

An important component of personalization is Web Mining. Web mining can be viewed as the extraction of structure from an unlabeled, semi-structured data set containing the characteristics of users/information respectively. The logs kept by web servers provide a classic example of such data. Other more active schemes exist, where the users are asked to explicitly rate pages, and these ratings (encryption) are stored or their viewing time is monitored. Web mining, much like data mining, can be said to have three operations of interests – clustering (e.g. finding natural groupings of users, pages etc.), associations (e.g. which URLs tend to be requested together), and sequential analysis (the order in which URLs tend to be accessed). The first two are perhaps of greater interest, and form the focus of our ongoing work. The object of this paper is to describe some of the challenges of web mining, point out why existing techniques of data mining may be inadequate, outline a possible direction for research, and present our preliminary work.

Like Data mining, Web mining needs to deal with problems of scale (extremely large data sets). However, there are several new challenges that are raised by Web mining that make the straightforward use of data mining techniques not particularly useful. For one, the clusters and associations in Web mining do not have crisp boundaries. They overlap considerably and are best described by fuzzy sets. In addition, bad exemplars (outliers) and incomplete data can easily occur in the data set, due to a wide variety of reasons inherent to web browsing and logging. Thus, Web Mining requires modeling of an unknown number of overlapping sets in the presence of significant noise and outliers. Further, the appropriate "metrics" or (dis)similarity measures between entities are not clear. For example, what is the distance between two URLs? Data mining techniques have typically developed in structured domains where these issues are not significant. New applications of mining to semi structured data will require that these issues be addressed.

2. Background and Rationale

The process of identifying structure in an unlabeled data set (i.e., model identification) in terms of categories or components plays a central role in web mining. Consider for example the problem of

---

* This work was supported in part by an IBM Faculty development award to Joshi and an ONR grant (N00014-96-1-0439) to Krishnapuram. We also thank Lycos for their support in making their log data available.

determining categories of users with “similar” interests, or the problem of grouping together a set of pages with similar content, and so on. In order for a model identification technique to be useful in an application such as Web mining, it needs to satisfy four basic requirements. (i) The technique should be able to handle overlapping components. (ii) It needs to be robust. By robustness, we mean that the model identification process (and hence the performance of a system) should not be affected drastically due to outliers (bad observations), provided there is enough “good data” to support the assumed model. (iii) The technique should be able to determine the appropriate number of components automatically, since a priori knowledge about the number of components is rarely available. (iv) Finally, it needs to be scalable to high dimensions and very large data sets.

Present State in Robust Model Identification: A variety of methods have been proposed for parameter estimation and model identification. For example, Gaussian mixture modeling has been studied extensively in the literature, and the Expectation-Maximization (EM) method [9] has been one of the most widely used approaches. Most current algorithms do not consider robustness aspects and assume that the number of components is known. Some use validity measures or other criteria such as Minimum Description Length (MDL) or Akaike's Information Criterion [10] to determine the number of components. However, they are computationally expensive, since the parameter estimation procedure needs to be repeated for several values of C. The use of "hard rejection" of outliers is not satisfactory in the "region of doubt", i. e., the region between the good points and the outliers. In such cases, what one needs is "soft" or "fuzzy" rejection. The breakdown point of an estimator is the smallest fraction of outliers that will cause the estimate to be arbitrarily wrong. When only a single cluster is present, a breakdown of close to 50% can be achieved by using techniques such as the Minimum Volume Ellipsoid (MVE) [8]. When there are multiple clusters, the theoretical breakdown point is much lower [4]. Moreover, when components overlap, algorithms that identify and remove one component at a time [11,12] will not work. Finally, none of these algorithms can handle the case where the components cannot be represented by prototypes. Traditional relational clustering techniques [13,14] are not robust.

Present State in Web Personalization and Mining: The notion of web mining and personalization has only recently been articulated, and that too mostly informally. However, scattered attempts to make web information access easier have been going on for quite a while. One of the earliest such attempts was the Firefly system [22], which operated in the domain of music choices, and attempted to provide CDs that best match a user’s professed interests. It accepts user feedback on its choices. In the Webwatcher project [18] at CMU, the notion of a tour guide is used. However, the underlying learning algorithms are not robust, which can pose a significant problem. As the authors observe, several users end up traversing paths that do not correspond to their stated interests. This can generate significant bad data points and outliers. Also, the users are presented with a binary choice in terms of expressing their interest using keywords – there is no notion of degree of interest. A somewhat similar approach is taken in the Avanti [21] project. It uses an initial interview to gather user interests, as well as possibly classify them into known “stereotype” users. Perkowitz and Etzioni [19] try to formalize the problem as being one of an “adaptive” web site. They define operations such as promotions/demotions, highlighting and linking that could be done on static pages to dynamically create content tailored for a specific user. While interesting, their formalism seems to demand a robust clustering technique in order to work successfully. Moreover, factors they propose such as correlations between two pages seem to be inherently fuzzy and best tackled using a fuzzy approach. Another approach to observing path traversal and clustering based on that data is advanced in [20]. The basic approach there is to define a path similarity measure for a given web site. Then, the logged data about user’s paths is clustered using a simple K-means algorithm to aggregate users into groups. However, it is not clear how the similarity metric is devised, and whether it can produce meaningful clusters. There has recently been some work in the area of recommender systems that also relates to our work. PHOAKS (People Helping One Another Know Stuff)[24] and referral web[25] are some examples.

In summary, the state of the art in web mining is quite primitive. There has been no systematic effort to identify the features needed, to study the effect of their “non-numericalness”, and to see whether converting them into numerical features is worthwhile. Most of the efforts have relied on existing clustering techniques that rely on crisp data and membership, even though most groups (of similar documents, of similar users etc.) that are needed for web mining are inherently fuzzy. Moreover, the algorithms used by most initial efforts into this area are not robust, and therefore highly susceptible to failure when operating on real data which tends to be quite noisy in this case. Finally, scalability is an issue that has often been ignored, but that is quite critical to these applications. For example, Lycos (which has made its data available to us for analysis) generates about 4GB of log data every day, and experiences about 8000 hits/minute.

3. Research Directions

In recent work, we have focused on robust and possibilistic clustering [2-5], as well as cluster validity and automatic determination of number of clusters [8]. We have also created intelligent middleware to enable adaptive web access in bandwidth and resource constrained scenarios such as mobile systems[16]. Our future work will focus on basic research into computationally efficient, fuzzy and robust model identification techniques and tools for web. More specifically, we are conducting research in the following areas.

(i) Establishing connections between fuzzy and robust statistical concepts:. Fuzzy clustering provides a better description tool when the clusters are not well-separated [1], as is the case with situations that arise in Web mining. Moreover, continuous membership values make the resulting algorithms are analytically more tractable, and have a lesser tendency to get stuck in local minima. The theory of robust statistics was developed independently of fuzzy sets. Recent work [2-5] clearly establishes that these two fields have much in common, and together they provide mutually supplementary and complementary ideas. For instance, fuzzy clustering techniques are capable of addressing the problem of multiple clusters. On the other hand, robust-statistical techniques emphasize robustness aspects, but typically can only deal with a single component.

(ii) Determining the optimum number of clusters: This is a largely unsolved problem. If we denote the number of clusters by C, traditional techniques perform clustering for a range of C values and evaluate the validity of the resulting partition to determine the optimum value. This approach is too tedious to be practical. This problem is particularly serious when noise and outliers are present, because the noise points may well be identified as a component. We propose to use a robust competitive agglomeration technique [5,7] to address this issue. In the worst case, the proposed algorithms are $O(N \log NC)$ in complexity, where N is the number of data points. The efficiency will be improved by combining Alternating Optimization (AO) [6] methods with Monte Carlo/bootstrapping techniques [8].

(iii) Extending the techniques discussed above to relational clustering: Relational clustering methods do not use prototypes so that they can handle non-numeric features via the idea of "similarity". Therefore, they are very useful for clustering non-numeric data such as URL's. We are developing robust relational clustering algorithms based on the techniques outlined above. In this case, the complexity increases to $O(N^2C)$. Again, the complexity can be reduced by the use of Monte Carlo techniques and bootstrapping methods

(iv) Developing techniques to impose similarity metrics on the data from the web: Metrics are required by the clustering algorithms. This involves converting alpha-numeric features such as protocol names or document locations into numerical values for prototype-based clustering, and defining similarity measures on them for relational clustering. Often, these metrics will be non-Euclidean in nature.

4. Robust Clustering Methods

Let $X = \{\mathbf{x}_j \mid j = 1...N\}$ be a set of n-dimensional feature vectors, and let $\mathbf{B} = (\beta_1,...,\beta_C)$ represent a C-tuple of prototypes each of which characterizes one of the C clusters. Each β_i consists of a set of parameters. Let u_{ij} represent the grade of membership of feature point $\mathbf{x}_j$ in β_i. The $C\times N$ matrix $U=[u_{ij}]$ is called a constrained fuzzy C-partition matrix if it satisfies [6]

$$u_{ij} \in [0,1] \text{ for all } i, j, \quad 0 < \sum_{j=1}^{N} u_{ij} < N \text{ for all } i, \quad \text{and } \sum_{i=1}^{C} u_{ij} = 1 \text{ for all } j. \tag{1}$$

The Fuzzy c-Means (FCM) algorithm partitions the feature vectors into C clusters based on an objective function $J(\mathbf{B},U;X)$ of the form

$$J(\mathbf{B},U;X) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m} d_{ij}^{2}. \tag{2}$$

In the above equation, $m \in [1,\infty)$ is a weighting exponent called the fuzzifier, and d_{ij}^2 is the distance from feature point $\mathbf{x}_j$ to prototype β_i. Minimization of (2) is usually achieved by a Picard iteration technique which updates memberships and prototypes in an alternating fashion until convergence. However, since $J(\mathbf{B},U;X)$ is essentially a sum of squares criterion, it is not robust. More Recently, there have been several attempts to robustify FCM [28-32]. Most of these algorithms have the potential to achieve the theoretical breakdown point of $N_{\min}/N$, where $N_{\min}$ is the cardinality of the smallest cluster. Many of them are based on the reformulated objective function for FCM [6]:

$$R(\mathbf{B};X) = \sum_{k=1}^{n}\left(\sum_{i=1}^{c} D_{ik}^{1/1-m}\right)^{1-m} = \sum_{k=1}^{n} H_k, \tag{3}$$

where $H_k = \left(\sum_{i=1}^{c} D_{ik}^{1/1-m}\right)^{1-m}$. Since the H_k values corresponding to outliers are large, the idea is to design the objective function so that its global minimum is achieved when large H_k are discounted or ignored. The objective function of the Robust FCM algorithm [32] is:

$$J_{RFCM} = \sum_{k=1}^{n} \rho\left(H_k\right).$$

This objective function applies a loss function $\rho(.)$ [8] to reduce the effect of outliers. The loss function is typically linear for small distances and then saturates for larger ones. The membership update equation for this formulation remains the same as that of the original FCM. However, the center update equation becomes:

$$\mathbf{c}_i = \frac{\sum_{j=1}^{N} w_j u_{ij}^2 \mathbf{x}_j}{\sum_{j=1}^{N} w_j u_{ij}^2},$$

where $w_j = w(H_j) = d\rho(H_j)/dH_j$ can be interpreted as the degree of "goodness" of point $\mathbf{x}_j$. Ideally, for noise points, this value should be as low as possible. The objective function of the Fuzzy Trimmed C Prototype (FTCP) algorithm (Kim et al., 1996) is:

$$J_{FTCP} = \sum_{k=1}^{r} H_{[k]} \tag{4}$$

where $H_{[k]}$ is the k-th item when the quantities H_i, $i=1,\ldots,n$ are arranged in ascending order, and r is a value less than N. If the value of r is set equal to $N-N_{\min}+1$, FTCP will achieve the theoretical breakdown point. The Fuzzy C Least Median of Squares (FCLMedS) algorithm [28] replaces the summation in (3) with the median. FTCP and FCLMedS can achieve a high breakdown point at the expense of computational complexity, since in theory they both require a random search procedure. Kim et al. [30] give a heuristic iterative technique to minimize (4). A genetic search is used for FCLMedS in [28].

Recently, Frigui and Krishnapuram [5] have introduced an algorithm based on competitive agglomeration (CA). This approach to determination of number of components combines the advantages of agglomerative clustering, partitional clustering, and robust statistics, and achieves a high breakdown point by initially approximating the data set by a large number of small clusters. The objective function of CA is:

$$J_R(B,U;X) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^2 \rho(d_{ij}^2) - \alpha \sum_{i=1}^{C}\left[\sum_{j=1}^{N} w_{ij} u_{ij}\right]^2 \quad \text{subject to} \quad \sum_{i=1}^{C} u_{ij} = 1 \text{ for all } j. \qquad (5)$$

The loss function $\rho(.)$ can be chosen as the identity function when there are no outliers. The first term allows us to obtain compact clusters and is minimized when there is one data point per cluster. The robust weights $w_{ij} \in [0,1]$ in the second term are related to the loss function by $w_{ij} = \partial\rho(d_{ij}^2) / \partial d_{ij}^2$. The weights are high for good points and low for noise points. Therefore, the second term is minimized when all good data points are lumped into one cluster. Thus, (5) tries to partition data set into the smallest possible number of clusters. It can be shown that the membership update equation for (5) is given by

$$u_{ij} = u_{ij}^{\mathrm{RR}} + u_{ij}^{\mathrm{Bias}}, \qquad (6)$$

where u_{ij}^{RR} is the robust relative membership and u_{ij}^{Bias} is the bias membership given by

$$u_{ij} = \left[\sum_{k=1}^{C}\left(\frac{\rho(d_{ij}^2)}{\rho(d_{ik}^2)}\right)^{1/(m-1)}\right]^{-1} \quad \text{and} \quad u_{ij}^{\mathrm{Bias}} = \frac{\alpha}{\rho(d_{ij}^2)}(N_i - \overline{N}_j). \qquad (7)$$

In (7), N_i is the cardinality of cluster i and $\overline{N}_j$ is the weighted average of cardinalities of all clusters, where the weight associated with the cardinality of cluster k is $1/\rho(d_{kj}^2)$. The second term in (6) can be either positive or negative, and it allows good clusters to agglomerate and spurious clusters to disintegrate. Initially, the value of C is large, and it is continually updated as clusters become extinct. The value of α, which relates to scale, needs to be initially large to encourage agglomeration. It can then reduced according to an "annealing schedule". This technique can potentially find clusters of various types if we use appropriate prototypes and distance measures. Since the algorithm starts with a large value of C, it insensitive to initialization effects.

The above prototype-based algorithms cannot be used if the features are not numerical. On the other hand, there are fuzzy relational clustering algorithms which do not use the idea of prototypes. These algorithms require that we compute a $N \times N$ similarity matrix $S = [s_{\mathrm{ij}}]$ in which s_{ij} represents the similarity of $\mathbf{x}_i$ and $\mathbf{x}_j$. The similarity matrix can also be viewed as a fuzzy relation matrix. The relational clustering algorithms are generally computationally more complex. Examples include the SAHN models [33] which are $O(N^2 \log N)$ and hierarchical in nature and methods that use objective functions [13,14] which are $O(N^2)$. In order for these models to be useful, we need to robustify them, find methods to determine the number of clusters C, and reduce the computational complexity. For example, the Relational Fuzzy c-Means (RFCM) [14] uses the objective fuction:

$$K_{RFCM}(U;X) = \sum_{i=1}^{C} \frac{\sum_{j=1}^{N}\sum_{k=1}^{N} u_{ij}^m u_{ik}^m r_{jk}}{2\sum_{t=1}^{N} u_{it}^m} \qquad (8)$$

The robust version of this objective function may be:

$$K_{RFCM}(U;X) = \sum_{i=1}^{C} \frac{\sum_{j=1}^{N}\sum_{k=1}^{N} u_{ij}^m u_{ik}^m \rho(r_{jk})}{2\sum_{t=1}^{N} u_{it}^m}$$

When the number of clusters is not known, we can use the competitive agglomeration idea, and modify the objective function in (8) as follows:

$$K_{RFCM}(U;X) = \sum_{i=1}^{C} \frac{\sum_{j=1}^{N}\sum_{k=1}^{N} u_{ij}^{m} u_{ik}^{m} r_{jk}}{2\sum_{t=1}^{N} u_{it}^{m}} - \alpha \sum_{i=1}^{C} \left[\sum_{j=1}^{N} u_{ij}^{2} \right] \tag{9}$$

We refer to the resulting algorithm as the Competitive Agglomeration for Relational Data (CARD) algorithm [34].

5. Preliminary Work

Recently, we have created a prototype web mining systems using our initial work into developing robust fuzzy clustering methods [34]. This system analyzes the web access logs from a server, and tries to mine "typical" user access patterns. Since access logs do not typically have user identification, we processed the logs to separate them into temporally compact "sessions". Extracting user profile categories from unstructured log data involves the following major steps:

<u>Preprocessing and Segmentation of the access log data into sessions:</u> 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", etc), (iv) URL of the page accessed, (v) Data transmission prototcol (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, etc), which are typically embedded in other pages and are only transmitted to the user's machine as a byproduct of access to a certain web page which has already been logged.

Next, analogous to WebMiner, 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 {\em 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 between 1 and N, the total number of URLs. Thus, the ith user session is encoded as an N-dimensional binary attribute vector $s(i)$ whose entries are 1 if the user accessed the corresponding URL during this session, and zero otherwise.

The ensemble of all sessions extracted from the server log file is 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 enable 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.

<u>Adaptation of Session Data to Clustering - Computing The Relation Matrix:</u> In the absence of any a priori knowledge, an unsupervised classification or clustering method seems to be ideally suited to partition the user sessions. Given the nature of the web mining problem, relational clustering methods need to be used. 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. The web sessions are too complex to convert to simple numerical features, partly because the organization of the web site (and not just the URLs themselves) must be taken into account when defining similarity of the web sessions. In fact, the URLs in a site have a hierarchical or tree-like (more precisely, forest like) structural composition. Therefore, we have defined a suitable

similarity measure between two sessions that incorporates both the structure of the site, as well as the URLs involved.

Clustering the Data Using CARD: The CARD algorithm described in the previous section was applied to the relational data to find the number of clusters and the partition automatically.

The procedure described above 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 at Columbia. After filtering out irrelevant entries, the number of distinct URLs accessed was 369. While applying CARD to the relational data, a cluster was discarded if its cardinality was less than 5. After clustering the relational data with CARD, the final number of clusters was 20. The results show that CARD succeeded in delineating many different profiles in the user sessions. Except for one cluster, all clusters correspond to real profiles reflecting distinct user interests. The profiles followed the access patters on typical users. A listing of all 20 profiles is not presented here due to a paucity of space, but is detailed in a technical report [34] available from our web site (http://www.cecs.missouri.edu/~joshi/web-mine) . The goodness of these clusters is recognizable through their low intra-cluster distances (considerably lower than the total pairwise session distance), and their high inter-cluster distances (the majority between .9 and 1). We are now experimenting with running this algorithm on data from our campus' main web server, as well as on log data that Lycos has made available to us. We are also creating an "instrumented" version of the departmental web site so that we can actively log user ids and time spent on browsing particular pages etc. We also ask the user to rate the web pages and monitor the user's viewing time.

6. Discussion

It is clear that for the new class of applications of data mining (e.g. web mining), several new challenges exist that will demand the development of new techniques for robust fuzzy clustering that can handle noisy, uncertain, vague, and incomplete information. As mentioned earlier, given our ability to mine user's interest from such data, we could create adaptive sites that would appear different to different users. In our preliminary work, we are using server parsed HTML to achieve this. For example when a user first comes to our site, we present to her a page with links to the top level page of the CECS department, the page where she spent the most time, and the page which we expect she wants based on the profile clustering.

References

1. J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
2. R. Krishnapuram and J. Keller, "A Possibilistic Approach to Clustering," IEEE Transactions on Fuzzy Systems, Vol. 1, No. 2, May 1993, pp. 98-110.
3. R. Krishnapuram, H. Frigui, and O. Nasraoui, "Fuzzy and Possibilistic Shell Clustering Algorithms and Their Application to Boundary Detection and Surface Approximation: Parts I and II," IEEE Transactions on Fuzzy Systems, Vol. 3, No. 1, Feb. 1993, pp. 29-60.
4. R. Davé and R. Krishnapuram, "Robust Clustering Methods: A Unified View," to appear in the IEEE Transactions on Fuzzy Systems.
5. H. Frigui and R. Krishnapuram, "A Robust Clustering Algorithm Based on Competitive Agglomeration and Soft Rejection of Outliers," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, June 1996, pp. 550-555.
6. R. J. Hathaway and J. C. Bezdek, "Switching regression models and fuzzy clustering", IEEE Transactions on Fuzzy Systems, vol. 1, no. 3, August 1993, pp. 195-204.
8. P. J. Rousseeuw and A. M. Leroy, Robust regression and outlier detection, John Wiley, New York, 1987.
7. S. Medasani and R. Krishnapuram, "Determination of the Number of Components in Gaussian Mixtures Based on Agglomerative Clustering" Proceedings of the IEEE International Joint Conference on Neural Networks, Houston, June 1997, pp. 1412-1417.

9. J. H. Wolfe, "Pattern clustering by multivariate mixture analysis," Multivariate Behavioral Research, Vol. 5, July 1970, pp. 329-350.
10. J. Zhang and J.W. Modestino, "A model-fitting approach to cluster validation with application to stochastic model-based image segmentation," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 12, No. 10, October 1990, pp. 1009-1017.
11. J.-M. Jolion, P. Meer and S. Bataouche, "Robust clustering with applications in computer vision", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 8, August 1991, pp. 791-801.
12. C. V. Stuart, 'MINPRAN: A new robust estimator for computer vision," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 17, no. 10, October 1995, pp. 925-938.
13. M. P. Windham, "Numerical Classification of Proximity Data with Assignment Measures," Journal of Classification, Vol. 2, pp. 157-172, 1985.
14. 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.
15. P. H. A. Sneath and R. R. Sokal, Numerical Taxonomy - The Principles and Practice of Numerical Classification, W. H. Freeman and Co., San Francisco, 1973.
16. Anupam Joshi, S. Weerawarana, and E. Houstis, “On Disconnected Browsing of Distributed Information”, Proc. Seventh IEEE Intl. Workshop in Research Issues in Data Engineering, 1997, pp. 101-108.
17. Joshi, Anupam, Ramakrishnan, N., Rice, J.R. and Houstis, E., “On Neurobiological, Neuro--Fuzzy, Machine Learning and Statistical Pattern Recognition Techniques”, IEEE Trans. Neural Networks 8:1, pp 18-31, 1997
18. 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.
19. M. Perkowitz and O. Etzioni, “Adaptive Sites: Automatically Learning from User Access Patterns”, UW-CS Technical report at http://www.cs.washington.edu/homes/map/adaptive/
20. C. Shahabi, 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), 1997, pp. 20-29.
21. J. Fink, A. Kobsa, and J. Schreck, “Personalized Hypermedia Information Provision through Adaptive and Adaptable System Features”, http://zeus.gmd.de/hci/projects/avanti/publications/ISandN97/ISandN97.html
22. Firefly, http://www.firefly.com
23. Joshi, Anupam, “To Learn or not to Learn...” IJCAI '95 Workshop on Adaptation and Learning in Multiagent Systems (G. Weiss & S. Sen eds.), Lecture Notes in AI v. 1042, Springer Verlag.
24. L. Terveen, W. Hill, B.Amento, “PHOKAS- A system for sharing recommendations”, Comm. ACM, 40:3, 1997
25. H. Kautz, B. Selman, M. Shah, “Referal Web: Customizing Social Networks and Collaborative Filtering”, Comm. ACM, 40:3, 1997
26. T. Imielinski, B. R. Badrinath, “Mobile Wireless Computing: Challenges in Data Management”, Comm. ACM 37:10, pp. 18-28, 1994.
27. NSF, Planning Workshop on Distributed Knowledge Work Environment: Digital Libraries, Santa-Fe, March 1997.
28. Nasraoui, O. and Krishnapuram, R. (1997). A Genetic Algorithm for Robust Clustering Based on a Fuzzy Least Median of Squares Criterion, *Proceedings of NAFIPS*, Syracuse NY, 217-221.
29. Davé, R. N. (1991). Characterization and Detection of Noise in Clustering, *Pattern Recognition Letters* ,12 (11) 657-664.
30. Kim, J., Krishnapuram, R. and Davé, R. (1996). Application of the Least Trimmed Squares Technique to Prototype-Based Clustering, Pattern *Recognition Letters*, 17, 633-641.
31. Frigui, H. and Krishnapuram, R. (1995). A Robust Clustering Algorithm Based on the M-Estimator, *Proceedings of the First International Conference on Neural, Parallel and Scientific Computations*, Atlanta, (1), 163-166.

32. Choi, Y. and Krishnapuram, R. (1996). Fuzzy and robust formulations of maximum-likelihood-based Gaussian mixture decomposition, *Proceedings of the IEEE International Conference on Fuzzy Systems*, 1899-1905.
33. P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy - The Principles and Practice of Numerical Classification*, W. H. Freeman and Co., San Francisco, 1973.
34. O. Nasraoui, H. Frigui, A. Joshi, R. Krishnapuram, "Mining Web Access Logs Using Relational Competitive Fuzzy Clustering", Technical report 98-01, CECS Department, University of Missouri, 1998.