A Fuzzy Relative of the k-Medoids Algorithm with Application to Web Document and Snippet Clustering

Raghu Krishnapuram[1], Anupam Joshi[2], and Liyu Yi[1]
[1]Department of Mathematical and Computer Sciences
Colorado School of Mines
Golden, Colorado 80401
{rkrishna, lyi}@mines.edu

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

Abstract

This paper presents new algorithms (Fuzzy c-Medoids FCMdd and Fuzzy c Trimmed Medoids or FCTMdd) for fuzzy clustering of relational data. The objective functions are based on selecting c representative objects (medoids) from the data set in such a way that the total dissimilarity within each cluster is minimized. A comparison of FCMdd with the Relational Fuzzy c-Means algorithm (RFCM) shows that FCMdd is much faster. We present examples of applications of these algorithms to Web document and snippet clustering.

1.Introduction

Object data refers to the the situation where the objects to be clustered are represented by vectors $\mathbf{x}_i \in \Re^p$. Relational data refers to the situation where we have only numerical values representing the degrees to which pairs of objects in the data set are related. Algorithms that generate partitions of relational data are usually referred to as relational (or sometimes pair-wise) clustering algorithms. Relational clustering is more general in the sense that it is applicable to situations in which the objects to be clustered cannot be represented by numerical features, but rather, only dissimilarities between pairs of objects can be measured. For example, we can use relational clustering algorithms to cluster URLs (Universal Resource Locators) if we can come up with a dissimilarity measure (or equivalently a similarity measure) to quantify the degree of resemblance between pairs of URLs. The pair-wise dissimilarities are usually stored in the form of a matrix called the dissimilarity matrix.

There are several well-known relational clustering algorithms in the literature. One of the most popular is the SAHN (Sequential Agglomerative Hierarchical Non-overlapping) model [1] which is a bottom-up approach that generates crisp clusters by sequentially merging pairs of clusters that are closest to each other in each step. Depending on how "closeness" between clusters is defined, the SAHN model gives rise to single, complete and average linkage algorithms. A variation of this algorithm can be found in [2]. Another well-known relational clustering algorithm is PAM (Partitioning Around Medoids) due to Kaufman and Rousseeuw [3]. This algorithm is based on finding k representative objects (also known as *medoids* [4]) from the data set in such a way that the sum of the within cluster dissimilarities is minimized. A modified version of PAM called CLARA (Clustering LARge Applications) to handle large data sets was also proposed by Kaufman and Rousseeuw [3]. Ng and Han [5] propose another variation of CLARA called CLARANS. This algorithm tries to make the search for the k representative objects (medoids) more efficient by considering candidate sets of k mediods in the neighborhood of the current set of k medoids. However, CLARANS is not designed for relational data. Finally, it is also interesting to note that Fu [6] suggested a technique very similar to the k medoid technique in the context of clustering string patterns generated by grammars in syntactic pattern recognition. Some of the more recent algorithms for relational clustering include [7] [8] [9] [10].

SAHN, PAM, CLARA and CLARANS generate crisp clusters. When the clusters are not well defined (i.e., when they overlap) we may desire fuzzy clusters. Two of the early fuzzy relational clustering algorithms are the ones due to Ruspini [11] and Diday [12]. Other notable algorithms include Roubens' Fuzzy Non Metric Model or FNM [13], Windham's Association Prototype Model or AP [14], Hathaway & Bezdek's Relational Fuzzy c-Means or RFCM [15], and Kaufman & Rousseeuw's Fuzzy Analysis or FANNY [3]. FANNY is in fact very closely related to RFCM, and is essentially equivalent to RFCM when the fuzzifier m, is equal to 2. In our experience, RFCM and FANNY are the most reliable. Some improvements on this algorithm can also be found in the literature. For example, the NERFCM model [16] extends RFCM to ease the restrictions that RFCM imposes on the dissimilarity matrix. More recently, Sen and Davé [17] generalize this approach further, including an extension to handle data sets containing noise and outliers.

One problem with RFCM is that it is computationally rather expensive. Thus, it is unsuitable for Web mining applications where n is extremely large. In this paper we present an objective function for fuzzy relational clustering based on the idea of identifying k medoids, and propose a heuristic algorithm to minimize it. We call this algorithm Fuzzy c-Medoids and abbreviate it as FCMdd rather than FCM. (FCM is usually associated with the Fuzzy C Means algorithm in the clustering community). The worst case complexity of FCMdd is $O(n^2)$, and it can be reduced further in practice. We compare the performance of FCMed with RFCM, and propose objective functions for robust versions of FCMdd.

2.The Fuzzy c Medoids Algorithm (FCMdd)

Let $X = \{\mathbf{x}_i | i = 1, \ldots, n\}$ be a set of n objects. Each object may or may not be represeted by a feature vector. Let $r(\mathbf{x}_i, \mathbf{x}_j)$ denote the dissimilarity between object $\mathbf{x}_i$ and object $\mathbf{x}_j$. Let $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c\}, \mathbf{v}_i \in X$ represent a subset of X with cardinality c, i.e., $\mathbf{V}$ is a c-subset of X. Let X_c represent the set of all c-subsets $\mathbf{V}$ of X. The Fuzzy Medoids Algorithm (FCMdd) minimizes:

$$J_m(\mathbf{V}; X) = \sum_{i=1}^{n} \sum_{i=1}^{c} u_{ij}^m \, r(\mathbf{x}_j, \mathbf{v}_i), \qquad (1)$$

where the minimization is performed over all $\mathbf{V}$ in X_c. In (1), u_{ij} represents the fuzzy [18], or possibilistic [19] [20] membership of $\mathbf{x}_j$ in cluster i. The membership u_{ij} can be defined heuristically in many different ways. For example, we can use the FCM [18] membership model given by:

$$u_{ij} = \frac{\left(\frac{1}{r(\mathbf{x}_j, \mathbf{v}_i)}\right)^{1/(m-1)}}{\sum_{k=1}^{c} \left(\frac{1}{r(\mathbf{x}_j, \mathbf{v}_k)}\right)^{1/(m-1)}}, \qquad (2)$$

where $m \in [1, \infty)$ is the "fuzzifier". Another possibility is:

$$u_{ij}^m = \frac{\exp\{-\beta r(\mathbf{x}_j, \mathbf{v}_i)\}}{\sum_{k=1}^{c} \exp\{-\beta r(\mathbf{x}_j, \mathbf{v}_k)\}}. \qquad (3)$$

Above equations generate a fuzzy partition of X in the sense that the sum of the memberships of an object $\mathbf{x}_j$ across classes is equal to 1. If we desire possibilistic memberships [19], we could use functions of the following type: [21]:

$$u_{ij} = \left[1 + \frac{r(\mathbf{x}_j, \mathbf{v}_k)}{\eta_i}\right]^{-1} \qquad (4)$$

$$u_{ij} = \exp\left(-\frac{r(\mathbf{x}_j, \mathbf{v}_i)}{\eta_i}\right) \qquad (5)$$

Since u_{ij} is a function of the dissimilarities $r(\mathbf{x}_j, \mathbf{v}_k)$, it can be eliminated from (1). This is the reason J_m is shown as a function of $\mathbf{V}$ alone. When (1) is minimized, the $\mathbf{V}$ corresponding to the solution generates a fuzzy or possibilistic partition via an equation such as (2). However, (1) cannot be minimized via the alternating optimization technique, because the necessary conditions cannot be derived by differentiating it with respect to the medoids. (Note that the solution space is discrete). Thus, strictly speaking, an exhaustive search over X_c needs to be used. However, following Fu's [6] heuristic algorithm for a crisp version of (1), we describe a fuzzy algorithm (FCMdd) that minimizes (1).

The Fuzzy c-Medoids Algorithm (FCMdd)

```
Fix the number of clusters c; Set iter = 0;
Randomly pick the initial set of medoids:
  V = {v_1, v_1, ..., v_c} from X_c;
Repeat
    for i = 1 to c do  /*Compute memberships:*/
        for j = 1 to n do
            Compute u_ij by using (2), (3),(4) or (5).
        endfor
    endfor
    Store the current medoids:    V^old = V;
    Compute the new medoids:
    for i = 1 to c do
        q = argmin_{1<=k<=n} sum_{j=1}^{n} u_ij^m r(x_k, x_j)
        v_i = x_q;
    endfor
    iter = iter + 1;
Until (V^old = V or iter = MAX_ITER).
```

The Hard c-Medoids Algorithm (HCMdd)

```
Fix the number of clusters c; Set iter = 0;
Randomly pick the initial set of medoids:
  V = {v_1, v_1, ..., v_c} from X_c;
Repeat
    for j = 1 to n do  /*Assign objects to clusters:*/
        Assign x_j to β_1;  r_min = r(x_j, v_1);
        for i = 2 to c do
            If (r(x_j, v_i) < r_min) assign x_j to β_i;
        endfor
    endfor
    Store the current medoids:    V^old = V;
    Compute the new medoids:
    for i = 1 to c do
        q = argmin_{x_k ∈ β_i} sum_{x_j ∈ β_i} r(x_k, x_j)
        v_i = x_q;
    endfor
    iter = iter + 1;
Until (V^old = V or iter = MAX_ITER).
```

For the sake of completeness, we have presented the crisp version of FCMdd above, which we call the Hard c Medoids (HcMdd) algorithm. In this algorithm, cluster i is denoted by β_i. The above algorithms fall in the category of Alternating Cluster Estimation [22] paradigm, and are not *guaranteed* to find the global minimum. It is advisable to try many random initializations to increase the reliability of the results.

3.Robust versions of FCMdd

It is well-known that algorithms that minimize a Least-Squares type objective function are not robust [23][24]. In other words, a single outlier object could lead to a very unintuitive clustering result. To overcome this problem, we present a variation of of FCMdd that is is based on the Least Trimmed Squares idea [25].

To design an objective function for a robust version of FCMdd based on the Least Trimmed Squares idea, we use the membership function in (2). Substituting the expression for u_{ij} in (2) into (1), we obtain:

$$J_m(\mathbf{V};\mathbf{X}) = \sum_{i=1}^{n}\left(\sum_{i=1}^{c}(r(\mathbf{x}_j,\mathbf{v}_i))^{1/(1-m)}\right)^{1-m} = \sum_{j=1}^{n} h_j, \quad (6)$$

where

$$h_j = \left(\sum_{i=1}^{c}(r(\mathbf{x}_j,\mathbf{v}_i))^{1/(1-m)}\right)^{1-m} \quad (7)$$

is $1/c$ times the harmonic mean of the dissimilarities $\{r(\mathbf{x}_j,\mathbf{v}_i)) : i = 1,\ldots,c\}$ when $m = 2$. The objective function for the Fuzzy c Trimmed Medoids (FCTMdd) algorithm is obtained by modifying (6) as follows:

$$J_m^T(\mathbf{V};\mathbf{X}) = \sum_{k=1}^{s} h_{k:n}. \quad (8)$$

The Fuzzy c Trimmed Medoids Algorithm (FCTMdd)

Fix the number of clusters c, and the fuzzifier m;
Randomly pick the initial set of medoids:
$\mathbf{V} = \{\mathbf{v}_1,\mathbf{v}_1,\ldots,\mathbf{v}_c\}$ from X_c;
$iter$ = 0;
Repeat
Compute the harmonic dissimilarities
h_j for $j = 1,\ldots,n$ using (7);
Sort $h_j, j = 1,\ldots,n$ to create $h_{j:n}$;
Keep the objects corresponding to the first s $h_{j:n}$;
Compute memberships for s objects:
for $j = 1$ **to** s **do**
for $i = 1$ **to** c **do**
Compute $u_{ij:n}$ by using (2);
endfor
endfor
Store the current medoids: $\mathbf{V}^{old} = \mathbf{V}$;
Compute the new medoids:
for $i = 1$ **to** c **do**
$q = \underset{1\le k\le s}{\text{argmin}} \sum_{j=1}^{s} u_{ij:n}^m \, r(\mathbf{x}_{k:n},\mathbf{x}_{j:n})$
$\mathbf{v}_i = \mathbf{x}_q$;
endfor
$iter = iter + 1$;
Until $\left(\mathbf{V}^{old} = \mathbf{V} \text{ or } iter = MAX\_ITER\right)$.

In (8) $h_{k:n}$ represents the k-th item when $h_j, j = 1,\ldots,n$, are arranged in ascending order, and $s < n$. The value of s is chosen depending on how many objects we would like to disregard in the clustering process. This allows the clustering algorithm to ignore outlier objects while minimizing the objective function. For example, when $s = n/2$, 50% of the objects are not considered in the clustering process, and the objective function is minimized when we pick c medoids in such a way that the sum of the harmonic-mean dissimilarities of 50% of the objects is as small as possible.

The objective function in (8) cannot be minimized easily. However, we can design the following heuristic algorithm. Again, we caution the reader that the above algorithms can converge to a local minimum. It is good to try many random initializations to increase the reliability of the results. Interestingly the worst-case complexity of this algorithm still remains O(n^2). This is a good result, considering that robust algorithms are very expensive. It is quite trivial to design a robust version based on the Least Median of Squares idea as well [26]. In this case we simply replace the summation in (8) by the median.

4.Experimental Results

In this section, we compare HCMdd and FCMdd with RFCM on two data sets. We then show an example of snippet clustering with FCMdd and FCTMdd.

The first data set is a collection of 1042 abstracts obtained from the Cambridge Scientific Abstract Web site. The abstracts correspond to 10 topics (distance education, filament, health care, intermetallic, laminate, nuclear, aeronautics, plastic, trade, furnace, and recycling). There are about 100 abstracts per topic, but since the abstracts were not carefully chosen, some are outliers. In addition, we deliberately added 20 outliers. The second data set is a collection of 59 HTML documents compiled by 6 students at the Colorado School of Mines. Each student was asked to collect about 10 Web pages related to a particular topic (fish, astronomy, ray tracing, chatroom conversation, neurobiology and sports).

The procedure we used to generate the feature vectors was as follows. We first used a "stop-word elimination and stemming" algorithm obtained from Louisiana State University [28] to filter out insignificant words, remove certain types of word-endings, and select 500 keywords out of the collection by using the inverted document frequency (IDF) method. Then we generated the feature vector corresponding to each document by recording the relative frequencies of occurrence of the keywords in the document. Thus, we generated a 500-dimensional feature vector for each document. To reduce the dimensionality, we applied principle component analysis and selected the eigenvectors corresponding to the top 10 eigenvalues as the new features. Each 500-dimensional vector representing a document was projected onto the 10 eigenvectors to form a 10-dimensional feature vector that represents the particular document. We perform the clustering on the dissimilarity matrix generated from normalized versions of these 10-dimensional vectors. In the case of RFCM, the dissimilarity matrix was generated using the square of the Euclidean distance between the vectors. In the case of FCMdd, several dissimilarity measures were tried.

Table 1 shows the results obtained for the first two data sets. The column with the heading FCMdd-cos corresponds to the case when FCMdd was run with the cosine dissimilarity measure (i.e., $1-$cosine of angle between vectors), and the columns with the headings FCMdd-E, FCMdd-E^2, and FCMdd-L_1 correspond to the case when FCMdd was run with the Euclidean distance, squared Euclidean distance, and L_1-norm respectively. The first row results are the average of 20 different runs of the algorithms on the Cambridge Scientific Abstract data set. In each run, 120 abstracts were randomly selected from the collection, and only these 120 abstracts were used to generate the keywords and the 10 eigenvectors. The 500-dimensional feature vectors were constructed for all the remaining abstracts, and these feature vectors were then projected onto the 10 eigenvectors to generate the object data for a particular run of the algorithms. This process was repeated 20 times. The second row of the table corresponds to the Web-page data set.

The results show that FCMdd compares favorably with RFCM in terms of classification rates, and at the same time is an order of magnitude faster. The CPU times (in seconds) were recorded on a Pentium II, 400 MHz processor, and do not include the time to compute the dissimilarity matrix.

We also have results on a collection of snippets corresponding to Web documents that were retrieved by a search engine [27]. However, we will present only one example due to space constraints. The data set we used was a set of snippets corresponding to 200 Web documents retrieved by a search engine in response to the query "salsa". We used the "stop-word elimination and stemming" algorithm as described above, followed by the IDF method to generate $p = 500$ keywords. Each snippet was then represented by a p-dimensional vector, where the $i-$th component represents the (normalized) frequency of occurrence of the $i-$th keyword in the snippet. To compute the dissimilarities between snippets, we used the Jacard Index dissimilarity measure given by:

$$d_J(\mathbf{s_1}, \mathbf{s_2}) = \frac{\sum_{i=1}^{p} min(s_{1i}, s_{2i})}{\sum_{i=1}^{p} max(s_{1i}, s_{2i})}, \tag{9}$$

where $\mathbf{s_1} = [s_{11}, ..., s_{1n}]^T$ and $\mathbf{s_1} = [s_{11}, ..., s_{1n}]^T$ are the two p-dimensional vectors representing to two snippets. FCMdd was applied to this data with $c = 7$, and the snippets were crisply assigned to the clusters after the algorithm converged. There were three main clusters with 151, 19, and 19 snippets. The remaining clusters had 1 to 6 snippets in them and therefore we did not consider them as significant. We computed average keyword frequencies for each individual cluster. The top 10 most frequent keywords (along with their frequencies in parentheses) are shown in Table 2. These keywords give us a "profile" of the cluster. As can be seen the first cluster is mostly about hot sauces, the second one is about salsa music, and the third one is about salsa dancing. Since FCMdd is not robust, the first cluster also contains many irrelevant (outlier) snippets. Table 3 shows randomly sampled snippets from each cluster. It can be seen that the last two snippets in cluster 1 are outliers.

We applied FCTMdd to the same data with 25% trimming, and found that most of the irrelevant snippets were identified correctly as outliers by the algorithm. For example, FCTMdd eliminates the last two snippets shown in Table 3 from cluster 1. Cluster 1 now contains only 65 snippets, and the other two clusters are unchanged. Table 4 summarizes the results of FTCMdd. If we compare the profile vectors produced by FTCMdd with those of FCMdd, we can see that the profile of cluster 1 is significantly strengthened by FTCMdd. The remaining clusters are virtually unchanged by FCTMdd, since their already have fairly strong profiles.

5.Conclusions

In this paper, we have presented a new relational fuzzy clustering algorithm based on the idea of medoids. The worst-case complexity of the algorithm is O(n^2), which happens while updating the medoids in each iteration. The complexity compares very favorably with other fuzzy algorithms for relational clustering, such as RFCM. Moreover, in our experience, the algorithms converge very quickly, in 5 or 6 iterations. Our preliminary results show that the algorithm gives good results on Web documents and snippets. We intend to use this algorithm in Web mining applications, where the objects (such as URLs) may not have a numerical representation. Since users' access patterns are are buried in data with significant noise components, robust methods are needed. Predictive models (including statistical tools) cannot produce reliable results unless the data represents a significant percentage of the total number of possible (traversal) patterns, which is astronomically large. Thus, methods for reducing the dimensionality of the problem are required. Moreover groups, i.e. clusters (of users, of web pages, of URLs) are not crisp, the same entity can belong to different groups to different degrees. Therefore, fuzzy approaches are more suitable for Web mining. Our approach to this problem is to simplify it by categorizing the user space as well as the document space by applying clustering techniques. By keeping track of the memberships of the users, web objects, and traversal patterns in the categories, we hope to achieve a significant reduction in dimensionality and increase the reliability of the results. The clustering methods we employ for categorization have to be necessarily robust and fuzzy, to be able to handle large percentages of outliers and overlaps. They also need to be of low complexity to deal with extremely large data sets.

Acknowledgements

Partial support of this work by the National Science Foundation Grants IIS 9800899 and IIS 9801711 is gratefully acknowledged.

References

[1] P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy - The Principles and Practice of Numerical Classification*, W. H. Freeman, San Francisco, 1973.

[2] S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient algorithm for large databases," in *Proceedings of SIGMOD '98*, Seattle, June 1998, pp. 73–84.

[3] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data, An Itroduction to Cluster Analysis*, John Wiley & Sons, Brussels, Belgium, 1990.

[4] L. Kaufman and P. J. Rousseeuw, "Clustering by means of medoids," in *Statistical Data Analysis Based on the L_1 Norm*, Y. Dodge, Ed., pp. 405–416. North Holland/Elsevier, Amsterdam, 1987.

[5] R. T. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," in *Proceedings of the 20th VLDB Conference*, Santiago, Chile, Sept. 1994, pp. 144–155.

[6] K. S. Fu, *Syntactic Pattern Recognition and Applications*, Academic Press, San Diego, CA, 1982.

[7] K. C. Gowda and E. Diday, "Symbolic clustering using a new similarity measure," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, pp. 368–377, 1992.

[8] G. D. Ramkumar and A. Swami, "Clustering data without distance functions," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, vol. 21, pp. 9–14, 1998.

[9] Y. El Sonbaty and M. A. Ismail, "Fuzzy clustering for symbolic data," *IEEE Transactions on Fuzzy Systems*, vol. 6, pp. 195–204, 1998.

[10] P. Bajcsy and N. Ahuja, "Location- and density-based hierarchical clustering using similarity analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1011–1015, 1998.

[11] E. H. Ruspini, "Numerical methods for fuzzy clustering," *Information Science*, vol. 2, pp. 319–350, 1970.

[12] E. Diday, "La methode des nuees dynamiques," *Rev. Stat. Appliquee*, vol. XIX(2), pp. 19–34, 1975.

[13] M. Roubens, "Pattern classification problems and fuzzy sets," *Fuzzy Sets and Systems*, vol. 1, pp. 239–253, 1978.

[14] M. P. Windham, "Numerical classification of proximity data with assignment measures," *Journal of Classification*, vol. 2, pp. 157–172, 1985.

[15] R.J. Hathaway, J.W. Devenport, and J.C. Bezdek, "Relational dual of the c-means clustering algorithms," *Pattern Recognition*, vol. 22, no. 2, pp. 205–212, 1989.

[16] R. J. Hathaway and J. C. Bezdek, "NERF c-means: Non-Euclidean relational fuzzy clustering," *Pattern Recognition*, vol. 27, pp. 429–437, 1994.

[17] S. Sen and R. N. Dave, "Clustering of relational data containing noise and outliers," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, Anchorage, May 1998, pp. 1411–1416.

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

[19] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 98–110, 1993.

[20] R. J. Hathaway and J. C. Bezdek, "Switching regression models and fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 3, pp. 195–204, 1993.

[21] R. Krishnapuram and J. M. Keller, "The possibilistic c-means algorithm: Insights and recommendations," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 385–393, 1996.

[22] T. A. Runkler and J. C. Bezdek, "Ace: A tool for clustering and rule extraction," *IEEE Transactions on Fuzzy Systems*, 1999.

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

[24] R. N. Davé and R. Krishnapuram, "Robust clustering methods: A unified view," *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 2, pp. 270–293, 1997.

[25] J. Kim, R. Krishnapuram, and R. N. Davé, "Application of the least trimmed squares technique to prototype-based clustering," *Pattern Recognition Letters*, vol. 17, pp. 633–641, 1996.

[26] O. Nasraoui and R. Krishnapuram, "A genetic algorithm for robust clustering based on a fuzzy least median of squares criterion," in *Proceedings of NAFIPS'97*, Syracuse, NY, Sept. 1997, pp. 217–221.

[27] O. Zamir and O. Etzioni, "Web document clustering: A feasibility demonstration," in *Proceedings of SIGIR'98*, Melbourne, Australia, Aug. 1998.

[28] J. Chen, A. Mikulcic, and D. H. Kraft, "An integrated approach to information retrieval with fuzzy clustering and fuzzy inferencing," in *Knowledge Management in Fuzzy Databases*, O. Pons, M. Ampara Vila, and J. Kacprzyk, Eds. Physica Verlag, Heidelberg, Germany, 1998.

Table 1: Results on Abstract and Web-Page Data Sets

Data Set	RFCM-E^2		FCMdd-cos		FCMdd-E		FCMdd-E^2		FCMdd-L_1	
	rate (%)	CPU (s)	rate (%)	CPU (s)	rate (%)	CPU (s)	rate (%)	CPU (s)	rate (%)	CPU (s)
Abstr	83.60	126.88	83.10	5.03	74.55	4.20	83.97	5.20	75.52	3.77
Web Pg	83.05	0.34	84.75	0.01	84.75	0.00	84.75	0.01	88.14	0.01

Table 2: FCMdd Results on Snippet Data Set

cluster	kw1	kw2	kw3	kw4	kw5	kw6	kw7	kw8	kw9	kw10
1	hot (.35)	sauc (.23)	thi (.21)	chile (.19)	recip (.15)	food (.13)	gourmet (.13)	garlic (.11)	spice (.10)	mexican (.09)
2	music (1.0)	danc (.63)	dj (.47)	latin (.42)	onli (.26)	todai (.21)	lesson (.16)	listen (.16)	live (.16)	floor (.16)
3	danc (1.4)	lesson (.31)	look (.32)	jeanni (.21)	gonzalez (.16)	jobi (.16)	london (.16)	lui (.16)	oxford (.16)	page (.16)

Table 3: Sample Snippets in 3 Clusters Found by FCMdd

cluster #1

1) Vermont maple syrup, gift baskets, hot sauce, salsa, new England specialty products, gourmet foods, corporate, sweets]. Vermont maple syrup, gift baskets, hot sauce, salsa, new England specialty products, gourmet foods, corporate,

2) HOT SAUCE. #299 Devil's Dozen! $38.00 Click here to see a few of our hot sauces! #201 Batten Island Gourmet, Mild $2.00 closeout - only 1 left #211 Mountain Man Fire Roasted Habanero 3.95$MountainManisourbestsellinghabsauce!\#212PurpleHaze$3.95

3) Made with searing red savina chiles, hot habanero chiles and thai chiles makes this salsa hotter then you know where. The Paradise Pineapple Salsa was awarded first place in the 1996 Fiery Foods Festival-fruit salsa division.

4) Kaari Tiistai 17.00-18.00 SALSA Alkeet Miguel 18.00-19.00 PARISALSA Alkeet Miguel 19.00-20.00 OPISKELE ESPANJAA Salsaamalla Alkeet (tunti pidet espanjaksi) Miguel 20.00-21.00 RUEDA de CASINO & PARISALSA Jatko Miguel Keskiviikko 17.00-18.00 FLAMENCO Alkeet (sevillanas) Kaari 18.00-19.00 FLAMENCO Jatko (tangos) Kaari 19.00-20.00.

5) By combining a multi-faceted corporate marketing background with the latest in computer animation and presentation techniques, ROUTE 66 PRODUCTIONS, LLC has formed a division specializing in sophisticated digital media applications, called DIGITAL SALSA.

cluster #2

1) Edinburgh Latin/Hispanic Music and Dance Service: Tel-Aviv Yigal Korolevski has kindly send me a summary of what's going on in Tel-Aviv. Wednesday - 21:00-22:00 (beginners), 22:00-24:00 Brazilian party.

2) A collection of the best music played at Tropicana famous dancing floor. Compositions by Ignacio Pieiro, Ernesto Lecuona, Enrique Jorrn, the father of cha cha ch, Celina Gonzlez, Miguel Matamoros. They received the award "Disco de Oro" for their

3) DJ Jesus R. R. DJ Jesus plays all the hottest Latins tunes: salsa, cumbia, merengue, banda, quebradita, ranchera and romantica. Available to DJ latin techno and salsa hits for any occasion. El Tigre" DJ Manny - LATIN MUSIC PRODUCTIONS. Phone: 283-4213. 3213

cluster #3

1) The instructor, Gustavo Sr. has been teaching authentic Latin dances in the Seattle area for many years, and travels often to his native Panama, to constantly update his knowledge of the true international dance hall scene. The instructor, Gustavo Sr. has

2) What I really love, is watching great salsa dancing - I wish I could dance well but need a lot of lessons and practice (so does my novia who is Guatemalteca). What I really love, is watching great salsa dancing - I wish I could dance well but need a lot of lessons and practice (so does my novia who is Guatemalteca).

3) After a few lessons you'll have stepped into a whole new scene - you'll quickly make new friends, and before long you'll be dancing the nights away at one of Viva Salsa's events or at one of the many salsa clubs in Oxfordshire, the home counties, London, New

Table 4: FCTMdd Results on Snippet Data Set

cluster	kw1	kw2	kw3	kw4	kw5	kw6	kw7	kw8	kw9	kw10
1	hot (.82)	sauc (.52)	thi (.48)	chile (.43)	gourmet (.29)	garlic (.25)	food (.23)	recip (.23)	pepper (.19)	mexican (.17)
2	music (1.0)	danc (.63)	dj (.47)	latin (.42)	onli (.26)	todai (.21)	lesson (.16)	listen (.16)	live (.16)	floor (.16)
3	danc (1.4)	lesson (.31)	look (.32)	jeanni (.21)	gonzalez (.16)	jobi (.16)	london (.16)	lui (.16)	oxford (.16)	page (.16)