Combining Fuzzy Clustering and Fuzzy Inferencing in Information Retrieval

Donald H. Kraft, Jianhua Chen and Andreja Mikulcic
Department of Computer Science
Louisiana State University
Baton Rouge, LA 70803-4020
E-mail: {kraft, jianhua}@bit.csc.lsu.edu

Abstract

We present an integrated approach to information retrieval which combines some techniques of fuzzy clustering and fuzzy inference in order to achieve optimal retrieval performance. To capture the relationships among index terms, fuzzy logic rules are used. We adapt several fuzzy clustering methods (such as fuzzy c-means and fuzzy hierarchical clustering) to the task of clustering documents with respect to the index terms. The clusters generated provide a basis for building the fuzzy logic rules. The clusters can also be used to form hyperlinks between documents. The fuzzy logic rules are applied with fuzzy inference to derive useful modifications of the initial query, which will guide the search for relevant documents. Alternative ways to use the fuzzy clusters are explored in this work as well. Our method combines fuzzy clustering and fuzzy inference with traditional relevance feedback approach for retrieval. The advantage of this approach is the emphasis on semantic information which relates the terms through the fuzzy clusters and fuzzy rules. A series of experiments have been conducted in order to validate this approach; a description of those experiments along with the results are presented.

I. INTRODUCTION

The focus of this paper is on the application of fuzzy clustering and fuzzy inferencing techniques in information retrieval. This is motivated by two factors. First, there is a great need to develop intelligent information retrieval systems in this information age, when the users are faced with the increasingly difficult task of searching through huge amounts of data for useful information. The explosive growth of information technology, on-line services, and the use of world wide web has certainly flooded ever more users with even more information. Therefore we face a greater need than ever for powerful, automated information retrieval systems. Second, the task of textual information retrieval naturally involves the handling of fuzziness and uncertainty, thus calling for the application of fuzzy techniques. In textual information retrieval, users submit queries to the retrieval engine, describing the kind of documents that are desired. The retrieval engine matches the queries to the documents in the text database, and returns to the user a list of the documents which are "best matches". Note that there is quite amount of uncertainty in both the document description (namely, what a document is "about") and the query specification (for what kinds of documents is the user, based on the query, looking). Thus, the application of fuzzy set theory in information retrieval is quite natural [12].

In this paper, we present an integrated approach to textual information retrieval (IR). We combine the strength of fuzzy sets theory and conventional IR techniques in order to achieve optimal retrieval performance. Some of the conventional IR techniques that are often employed are inverted document frequency (IDF) measures as a basis for document vector representations, and the cosine measure for query-document similarity [18]. In order to capture the semantic connection between index terms, fuzzy logic rules are constructed. The discovery of fuzzy rules uses two fuzzy clustering techniques: the fuzzy c-means algorithm [1, 2] and the hierarchical clustering algorithm [17, 18].

Interestingly, in our experiments using documents from the US Air Force *Engineering Data Compendium* (EDC) database[3], both algorithms give similar clusters. From the fuzzy clusters and the prototypes (centers) of clusters, fuzzy logic rules are obtained. The clusters can also be used to build hyperlinks between relevant documents. We use the fuzzy logic system generated by Chen and Kundu [6], which is sound and complete, for fuzzy inferencing in order to derive useful modifications of the initial query. We then use the modified query to guide the search for relevant documents. The advantage of our approach is that semantic information embedded in the rules has been utilized, which should lead to superior retrieval performance. To validate our method, we carry out several experiments using the Air Force EDC database. We have also gotten evaluations for the clustering results from experts [14]; these results confirm the validity of our method.

The paper is organized as follows. In Section II, we present a preliminary background for the vector space approach to information retrieval and a brief description of the fuzzy clustering techniques used in our experiments. In Section III, we describe the application of the clustering algorithms to document clustering and fuzzy rule discovery. The fuzzy logic inference method for deriving new queries is presented in Section IV. Section V contains our conclusions.

0-7803-5877-5/00/$10.00 © 2000 IEEE

II. PRELIMINARIES

We briefly describe the (traditional) vector space approach to IR and the two fuzzy clustering methods used in our experiments. Throughout the paper, we consider a finite set of textual documents, $D = \{D_1, D_2, \cdots, D_N\}$, and a finite set of index terms, $T = \{t_1, t_2, \cdots, t_s\}$.

A. The Vector Space Approach To Information Retrieval

The vector space model is a representative of the ranked, "best-match" retrieval models. In this model, each document D_i is represented as a vector of dimension s, the number of terms:

$$D_i = \langle w_{i1}, w_{i2}, \cdots, w_{is} \rangle. \qquad (2.1)$$

Here, each w_{ij} is a real number (typically positive), characterizing the *weight* of the term t_j in D_i. These weights, called indexing weights, can be computed from the frequencies of occurrence of the terms as follows:

$$w_{ij} = f_{ij} * \log[N / N_j], \qquad (2.2)$$

where f_{ij} is the frequency with which term t_j occurs in document D_i, N is the number of documents in the collection, and N_j is the number of documents in which term t_j occurs at least once. Equation (2.2) is called the inverted document frequency (IDF) model [18]. Moreover, terms can be generated from the text itself as keywords; one can remove words that are too common and non-content bearing (e.g., "a", "the", "however") from the natural language of the texts, and can then stem the remaining words (e.g., "work", "worker", "worked", "working"), all before doing the frequency analysis [18].

A query q is represented in the same way as an s-dimension vector:

$$q = \langle w_{q1}, w_{q2}, \ldots, w_{qs} \rangle. \qquad (2.3)$$

Here, the w_{qj} weights are called query weights.

The degrees of match between a query and the documents are obtained by comparing the vectors and computing similarity levels. For a given query, a ranked collection of "best match" documents according to similarity measures will be returned to the user. Salton suggests using the cosine measure as the criterion for document and query similarity [18]. Given a document D_i and a query q, as represented above in (2.1) and (2.3), the cosine similarity measure $SIM(D_i, q)$ is defined to be:

$$SIM(D_i, q) = \frac{\sum_{j=1}^{s} w_{ij} * w_{qj}}{\sqrt{\sum_{j=1}^{s} w_{ij}^2}\sqrt{\sum_{j=1}^{s} w_{qj}^2}} \qquad (2.4)$$

B. Fuzzy Hierarchical Clustering

By fuzzy hierarchical clustering, we mean agglomerative hierarchical clustering (AHC) [17, 18] based on a weighted similarity measure. The idea behind AHC is fairly simple. We start with the set of objects to be clustered and a similarity measure $SIM(O_i, O_j)$ for any pair of objects (O_i, O_j) in the data set. The AHC algorithm will initially make every object a cluster. Then, the algorithm will repeatedly merge the two "most similar" clusters into one cluster until the similarity between any two clusters falls below some heuristic threshold. The measurement of similarity between two clusters can be done in a number of ways. For example, one can take the minimum of the similarities between any pair of objects, each from one cluster. This is the so-called *complete link clustering* (CLC) [18]. One can also use the maximum, or the average, pair-wise similarity measures. In the experiments done in this work, we use the CLC approach.

C. Fuzzy Clustering by Fuzzy C-means Algorithm

The fuzzy c-means algorithm by Bezdek [1, 2] is a family of algorithms which form fuzzy clusters iteratively through optimizing an objective function. Given a set of n sample data points $p_i = \langle x_{i1}, x_{i2}, \ldots, x_{is} \rangle$: $1 \le i \le n$, and the desired number of clusters C (≥ 2), the fuzzy c-means algorithm produces C fuzzy clusters A_k, $1 \le k \le C$, by finding cluster centers v_k and the membership values $\mu_{ki} = \mu_k(p_i)$ for each point p_i and cluster A_k. The algorithm chooses the μ_{ki} and v_k so that the following objective function (where $m > 1$ is a fixed constant) is minimized:

$$J_m = \sum_{k=1}^{C} \sum_{i=1}^{n} (\mu_{ki})^m \|p_i - v_k\|^2 \qquad (2.5)$$

This is subject to the constraints that $\sum_k \mu_{ki} = 1$ for each i, and that every $\mu_{ki} \ge 0$. Here, v_k is visualized as the center of the cluster A_k. Moreover, $\|p_i - v_k\|$ denotes the distance between the points p_i and v_k, which is taken to be the Euclidian distance in this work.

The equations for determining the μ_{ki} that minimize J_m are given by:

$$\mu_{ki} = \frac{[\|p_i - v_k\|^2]^{-1/(m-1)}}{\sum_{j=1}^{C} [\|p_i - v_j\|^2]^{-1/(m-1)}}, \quad 1 \le k \le C, \; 1 \le i \le n. \qquad (2.6)$$

together with the following equations for v_k (which are to be considered coordinate-wise for p_i and v_k):

$$v_k = \frac{\sum_{i=1}^{n} (\mu_{ki})^m p_i}{\sum_{i=1}^{n} (\mu_{ki})^m}. \qquad (2.7)$$

The actual computation of μ_{ki} begins by initializing the μ_{ki} values randomly, subject to $\mu_{ki} \ge 0$ and $\sum_k \mu_{ki} = 1$ for each i. One then iteratively uses (2.7) to first compute the v_k values, and then uses those values in (2.6) to update the μ_{ki} values. The process continues until the maximum of the absolute difference in the membership values (and the centers) in

the current iteration and those in the previous iteration falls below some convergence threshold $\delta > 0$. The convergence proofs of the c-means algorithm are presented in [1, 2]. Extensions of fuzzy c-means algorithms have been applied for learning fuzzy control rules [13, 15].

III. APPLICATION OF FUZZY CLUSTERING FOR DOCUMENT CLASSIFICATION

In this section, we describe the fuzzy clustering approach to document classification and fuzzy rule discovery. We will first briefly describe the AirForce EDC data set, and then present the results of clustering on this data.

A. The Air Force EDC Data Set

The Air Force EDC data set [3] is a text database, which is a part of the US Air Force's multimedia ergonomics database system, CASHE:PVS (Computer Aided Systems Human Engineering: Performance Visualization System). The CASHE:PVS system consists of the complete Engineering Data Compendium (EDC) data set [3], the military standard (MIL-STD-1472D) Human Engineering Design Criteria for Military Systems, Equipment, and Facilities [7], and a unique visualization tool, the Perception and Performance Prototyper (P^3). More information about the CASHE:PVS system can be obtained from the web site [19].

CASHE:PVS has been produced to define new approaches to communicate human factors data and to provide access to technical information relevant to human performance design problems. The goal is to enable ergonomics to be supported as a full partner among other design disciplines within a computer-aided environment [4, 5]. For example, a designer interested in the intelligibility of speech in a noisy environment, such as the cockpit of an airplane, can look up the appropriate data in CASHE:PVS and peruse them. However, to gain a deeper understanding of what the data really means, that designer can also use the P^3 visualization tool to experience the data. Sample speech signals can be heard in varying amounts of background noise, different noises can be used, and techniques to improve speech intelligibility are demonstrated. The reference data, coupled with interactive visualization, provide the designer with a synthesis and analysis capability for working with other designers.

The EDC data set consists of 1136 documents containing engineering design and human factor data. A subset of the EDC related to audio topics has been selected in order to keep the dataset to a manageable size for this study. Table 1 illustrates the portion of the EDC which was used for this study. This represents N = 114 entries out of the 1136 in the entire EDC. These N entries yield s = 2857 keywords after using a stoplist and stemming.

Table 1. Portions of the EDC Used in This Study

Sec. 2(Auditory Acquisition of Info.)	2.1 (Measurement of Sound)
Sec. 2(Auditory Acquisition of Info.)	2.2 (Physiology of the Ear)
Sec. 2(Auditory Acquisition of Info.)	2.3 (Detection)
Sec. 2(Auditory Acquisition of Info.)	2.4 (Discrimination)
Sec. 2(Auditory Acquisition of Info.)	2.5 (Temporal Resolution)
Sec. 2(Auditory Acquisition of Info.)	2.6 (Loudness)
Sec. 2(Auditory Acquisition of Info.)	2.7 (Pitch)
Sec. 2(Auditory Acquisition of Info.)	2.8 (Localization)
Sec. 6 (Perceptual Organization)	6.4 (Auditory Perceptual Organ.)
Sec. 8 (Human Language Processing)	8.3 (Intelligibility of Speech)
Sec. 8 (Human Language Processing)	8.4 (Intelligib. of Alt. Speech)
Sec. 10 (Effects of Envir. Stressors)	10.3 (Noise)

B. Experimental Fuzzy Clustering Results

We have performed several experiments using the fuzzy hierarchical clustering and fuzzy c-means algorithms described in Section II on the chosen subset of the EDC database. Before applying these clustering algorithms, some pre-processing has to be performed to extract the vector space representations. First, as noted above, stop words have to be eliminated, as they provide no useful characterization of the document. Subsequently, we need to apply a stemming algorithm to find the root form of the words. We use the stemming algorithm in [8] for this task. Newly encountered words are added to a global word list, and the word frequency count is calculated for each word in each document. As mentioned previously, out of the 114 documents, we obtained 2857 keywords (index terms) after stop word elimination and stemming. Finally, we need to reduce this list of terms to an even smaller size in order to be amenable to clustering. This is done by choosing the top s maximal-weighted keywords from the data set. The maximal weight w_j of term t_j is obtained by taking the maximal weight of t_j over all 114 documents, i.e., $w_j = \max_{i=1}^{114} w_{ij}$. After finding w_j for each of the 2857 terms, we choose the top s terms from the term list when sorted in descending order of the w_j values. Here, for our experiment, s is set to be 100. Thus, after pre-processing, each of the 114 documents is represented as a vector of dimensionality of 100.

We conduct two experiments on fuzzy hierarchical clustering: one uses word frequency counts as weights in the vector space representation, and the other uses inverted document frequencies. The heuristic threshold for the minimum similarity (above which to merge two clusters) was heuristically set to 0.01 for the inverted document frequency case,

and it was set to 0.1 for the word frequency case. In both cases, the algorithms formed 12 clusters out of the 114 documents and the clusters obtained by the two experiments differ only slightly. According to the expert evaluations conducted by people involved with the US Air Force database [14], the experiments using the inverted document frequencies yield a slightly better result.

In applying the fuzzy c-means algorithm, we set the convergence threshold δ to 0.001, and the number of clusters, C, to 12. This choice of C reflects our intention to show that fuzzy c-means algorithm can also find natural clusters in documents, just like the hierarchical clustering method can. We note that the fuzzy c-means algorithm forms clusters by optimizing the objective function (2.5), which is pretty much based on a Euclidean distance measure in this work (although any other distance measure or dis-similarity measure can be used), whereas hierarchical clustering is based on the cosine similarity measure. Therefore, the theoretical underpinnings of the two approaches are not quite the same. However we would like to note that both will produce reasonably meaningful clusters.

The fuzzy c-means algorithm will produce "fuzzy" clusters in the sense that μ_{ki}, the membership of document D_i in cluster A_k, is a value in the interval [0,1]. This is in contrast to the case with hierarchical clustering, in which each document belongs to exactly one cluster. Since we would like to compare the performance between hierarchical and fuzzy c-means clustering, "hardening" is performed to the fuzzy clusters obtained by the fuzzy c-means algorithm. That is, for each document D_i, we find the cluster index k (for cluster A_k) such that μ_{ki} is maximal over the μ_{ji} for all clusters A_j; we then set μ_{ki} to 1 and the other μ_{ji} values to 0 for $j \neq k$.

We have performed several experiments with the c-means algorithm, varying the values of the parameter $m > 1$ in (2.7). We found that for the subset of EDC database, when $m = 2.0$ or $m \geq 1.5$, the cluster centers take on the same, or nearly the same, value, so that the clusters look identical, and each document belongs to each of the 12 clusters with essentially the same membership value. Therefore, after "hardening", the clusters obtained are chaotic. On the other hand, for $m \leq 1.4$, the crisp clusters obtained after hardening seem to be more correct. This was verified by the experts [14], when they were given the results of two fuzzy C-means clusterings, one with m = 2.0 and the other with m = 1.1. Of course, the experts were also given the two hierarchical clustering results.

The experts indicate [14] that in evaluating clusters, they considered both:

> (1) whether the entries within a given cluster were related enough to make a valid cluster, and (2) whether other entries that are equally related were missing from the cluster.

They also comment [14] that

> "... the original fuzzy clustering method (fuzzy c-means with m = 2.0) did a poor job of clustering ... The clusters as a whole don't really describe anything."

They further comment that

> "Both of the hierarchical methods did a much better job of clustering the entries than did the original fuzzy clustering method. Clustering using inverted frequency may have been a little bit better than clustering that considered the number of occurrences in each entry, but the difference was very slight ... Modified fuzzy clustering (c-means with m = 1.1) was at least as good as hierarchical clustering. It tended to generate fewer large, heterogeneous clusters (though it did have one extremely mixed, 39-member cluster!) ... The hierarchical/inverted-frequency method was the best at avoiding outliers (items off the topic of the other cluster entries), but the modified fuzzy method also did a pretty good job at this. Both hierarchical methods (but not the fuzzy methods) also managed to create at least one "perfect" cluster (no misses and no outliers)."

IV. COMBINING FUZZY CLUSTERING AND FUZZY INFERENCE FOR IMPROVING RETRIEVAL PERFORMANCE

In this section, we present our approach of using fuzzy clusters and fuzzy inference to improve retrieval performance. We explore two methods that utilize the fuzzy clusters generated from the EDC data set: (1) Use the fuzzy clusters to build fuzzy logic rules which capture the semantical connection between terms; and then use these rules under a fuzzy logic system (Chen and Kundu [6]) to derive useful modifications of the user's original query. (2) Use the fuzzy clusters directly in retrieval. Our experiments seem to suggest that the use of inferred query and fuzzy clustering does improve retrieval performance.

A. Rule Construction from Fuzzy Clusters

After finding the document clusters by the fuzzy c-means algorithm (with hardening), we can construct fuzzy rules of the form

$$[t_i \geq w_i] \rightarrow [t_j \geq w_j]$$

from the clusters and their centers obtained by the fuzzy c-means algorithm. Here, t_i and t_j are terms, and w_i and w_j are positive real weights in the interval (0,1]. The intuitive meaning of the rule is that whenever term t_i's weight (in a document or query) is at least w_i, the related term t_j's weight (in the same document or query) should be at least w_j. These rules can be applied to derive useful modifications of the user's original query.

The current implementation of our method uses the the *centers* obtained by the fuzzy c-means algorithm to construct

the fuzzy logic rules. Our method proceeds as follows: First, we normalize the vectors representing the cluster centers. Then for each cluster center, sort the terms in descending order of term weights and focus on the first M (≥ 2) terms in this sorted list. Subsequently, build term pairs from the chosen terms in each cluster center in the form of $\langle[t_i, w_i], [t_j, w_j]\rangle$. Moreover, multiple occurrence of the same pairs with different weights (obtained from different cluster centers) will be merged by selecting the minimal weight for each term over all pair occurrences. Finally, from the pair of the form $\langle[t_i, w_i], [t_j, w_j]\rangle$, we build two rules:

$$[t_i \geq w_i] \rightarrow [t_j \geq w_j]$$

and

$$[t_j \geq w_j] \rightarrow [t_i \geq w_i]$$

Note the symmetry of the above pair of rules. The intuitive idea behind the pair of rules is that the occurrence of the term t_i with a weight at least w_i should be always accompanied by the term t_j with a weight at least w_j, and vice versa.

B. Use of Fuzzy Inference for New Query Derivation

The fuzzy logic rules obtained by fuzzy clustering and rule discovery can be used to modify a user's original query, using the fuzzy logic system developed by Chen and Kundu in [6]. We note that each fuzzy rule derived from the fuzzy clusters is of the form

$$r: [t_i \geq w_i] \rightarrow [t_j \geq w_j], \qquad (4.1)$$

which is a well-formed formula in the fuzzy logic defined in [6], where formulas are formed by using logical connectives $\{\wedge, \vee, \neg, \rightarrow\}$, the logical constant $\perp$ (= false), and propositions of the form $[A \leq \alpha]$, or of the form $[A \geq \alpha]$, where A is an ordinary atom in a propositional logic and $\alpha \in [0, 1]$. The above rule is used to modify a user's query q as follows. Given the query q in the form

$$q = \langle w_{q1}, w_{q2}, \ldots, w_{qs} \rangle. \qquad (4.2)$$

the above rule (4.1) is applicable to q if $w_{qi} \geq w_i$ and $w_{qj} \leq w_j$. The application of this rule to q will yield q′, which coincides with q on each dimension except $w_{q'j} = w_j$. Note that this application step precisely corresponds to the modus-ponens inference in the fuzzy logic in [6]. Let $R = \{r_1, r_2, ..., r_z\}$ be the set of all fuzzy rules learned, and let q be the user's initial query. The modified query q′ is obtained from q by repeatedly applying the rules in R in order to derive new weights for each term t_j ($1 \leq j \leq s$) until no more applicable rules can be found. Note that the final modified query q′ is independent of the order of rules applied. This modified query q′ will be used to search for relevant documents.

We have implemented the query modification method and performed several experiments with it. The preliminary results obtained in the experiments suggest that the modified queries are helpful to improve precision in most of the cases. For example, suppose we want to get documents regarding the topics of "pitch" and "adaptation", with more emphasis on "pitch". This is modeled by the query q_1 with weight for the term "adapt" set to be 0.4, and the weight for "pitch" set to be 0.8, and weights for all other terms set to be 0. The intended target set of documents is for those in subsection 2.7 (with 11 documents), which is essentially captured by the cluster 5 obtained by fuzzy clustering with $m = 1.10$, which is judged by the experts as a "pretty good cluster". Starting with q_1, we apply the query modification method and get the modified query q_1', which has the same weights for "adapt" and "pitch" as in q_1 and several additional terms with positive weights: term "interrupt" got weight 0.2029, term "modul" (root of "modulate", "modulation", etc.) got weight 0.2399 and term "tone" got weight 0.6155. Using both queries q_1 and q_1' for the retrieval task, we have the following observations on the query results:

(1) If we compare the top M documents obtained by query q_1 (according to their similarity to q_1) v.s. the top M documents obtained by q_1', where M is a fixed number (say M = 10), then q_1' fares better (or comparable) in both precision and recall. For example, when M = 11, q_1 produces 9 documents out of the 11 documents cluster, giving rise to a 81.8 percent recall and precision; while q_1' captures exactly the 11 documents in the cluster, resulting in a 100 percent recall and precision.

(2) If we compare the documents obtained by q_1 with similarities above some threshold $0 < \delta < 1$ v.s. those obtained by q_1' with the same similarity threshold, then q_1' gives better precision with a comparable or slightly inferior recall. For example, for $\delta = 0.1$, q_1 produces 17 documents which contains all the 11 relevant documents, resulting in a 100 percent recall but a 68.7 percent precision; while q_1' produces 12 documents including all 11 relevant ones, thus giving rise to a 100 percent recall and 91 percent precision. When we take $\delta =$ 0.15, q_1 presents 15 documents with 100 percent recall and 73 percent precision; while q_1' presents 8 documents with 72 percent recall and 100 percent precision.

The experiments we have performed on using the query modification method are still quite limited and the nature of the results reported here on the performance of the modified query should be considered *preliminary*. Further studies are needed to validate the query modification method.

C. Alternative ways to utilize the fuzzy clusters

We have explored other alternative ways to use the fuzzy clusters obtained by fuzzy c-means method in retrieval. One way is to match the user's query (without modification by the fuzzy rules) to the cluster centers and return all the documents in the cluster which has the best match with the query. Another way is to combine the use of fuzzy rules for query modification and the direct use of clusters based on the

modified query.

V. CONCLUSIONS

In this paper, we present an integrated approach to information retrieval which combines the strength of fuzzy sets theory and traditional IR techniques to achieve optimal retrieval performance. Fuzzy clustering and hierarchical clustering methods are applied for document classification and for finding natural clusters in documents. From the fuzzy clusters, fuzzy logic rules are constructed in an attempt to capture semantic connections between index terms. The fuzzy rules are subsequently used in fuzzy inference within a fuzzy logic system to modify user's queries in retrieval. A series of experiments, in conjunction with expert evaluations, have been conducted to validate our method.

The experiments with clustering methods using the Air Force EDC data set show that both clustering methods can find reasonable natural clusters in documents. However, neither method is perfect, as judged by experts [14]. This is not surprising, considering the fact that only very primitive statistical information (word frequency or IDF) is used in the clustering methods, and no semantic information (e.g.word meaning and connections between words) is available. Our future work will address the issue of incorporating semantic information in the clustering process. The preliminary results obtained in fuzzy rule construction and fuzzy inference for query modification also show good promise for retrieval precision improvement. We will perform more extensive validation experiments on this in the near future.

ACKNOWLEDGMENT

The authors are grateful to Janet Lincoln and Don Monk at the US Air Force for offering their expert evaluations on the experimental results in this work, and for allowing us to use the Air Force's EDC data set. We would also like to thank Sukhamay Kundu for allowing us to use his implementation of the fuzzy c-means algorithm.

REFERENCES

[1] J.C. Bezdek, A convergence theorem for the fuzzy ISODATA clustering algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2), 1980, pp. 1-8.

[2] J.C. Bezdek, R.J. Hathaway, M.J. Sabin, and W.T. Tucker, Convergence theory for fuzzy c-Means: counterexamples and repairs, *IEEE Transactions on Systems, Man, and Cybernetics* (17), 1987, pp. 873-877.

[3] K.R. Boff, J.E. Lincoln (Eds), *Engineering Data Compendium: Human Perception and Performance* v I, II, and III (Wright-Patterson Air Force Base, OH: Human Engineering Division, Harry G. Armstrong Medical Research Laboratory), 1988.

[4] K.R. Boff, D.L. Monk, W.J. Cody, Computer Aided Systems Human Engineering: A Hypermedia Tool, Space Operation Applications and Research (SOAR) 1991, July Houston: NASA.

[5] K.R. Boff, D.L. Monk, S.J. Swierenga, C.E. Brown, W.J. Cody, Computer-Aided Human Factors for Systems Designers, July 1991 (San Francisco: Human Factors Society annual meeting).

[6] J. Chen, S. Kundu, A sound and complete fuzzy logic system using Zadeh's implication operator, *Foundations of Intelligent Systems: Lecture Notes in Computer Science 1079*, 1996, pp. 233-242.

[7] Department of Defense, *Human Engineering Design Criteria for Military Systems, Equipment, and Facilities* (MIL-STD-1472D), Notice 3, Washington, DC, 1994.

[8] W. B. Frakes, Stemming algorithms, In: W. B. Frakes, R. Baeza-Yates (Eds), *Information Retrieval: Data Structures & Algorithms*, Prentice Hall, 1992.

[9] G.J. Klir, B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Upper Saddle Rive, NJ: Prentice-Hall, 1995.

[10] Kraft, D. H., Bordogna, G., and Pasi, G., An Extended Fuzzy Linguistic Approach to Generalize Boolean Information Retrieval, *Information Sciences*, (2), November, 1995, pp. 119-134.

[11] D.H. Kraft, B.R. Boyce, Approaches to Intelligent Information Retrieval, in F.E. Petry, M.L. Delcambre (Eds), *Advances in Databases and Artificial Intelligence*, volume 1: Intelligent Database Technology: Approaches and Applications, Greenwich, CT: JAI Press, 1995, pp. 243-261.

[12] D.H. Kraft and D.A. Buell, Fuzzy Sets and Generalized Boolean Retrieval Systems, *International Journal of Man-Machine Studies*, v. 19, 1983, pp. 45-56; reprinted in D. Dubois, H. Prade, and R. Yager, (Eds), *Readings in Fuzzy Sets for Intelligent Systems*, San Mateo, CA: Morgan Kaufmann Publishers, 1992.

[13] S. Kundu, J. Chen, Fuzzy linear invariant clustering for applications in fuzzy control, *Proceedings of NAFIPS/IFIS/NASA'94*, San Antonio, TX, 1994.

[14] J. Lincoln, D. Monk, private communications, 1997.

[15] A. Mikulcic, J. Chen, Experiments on using fuzzy linear clustering from fuzzy control system design, *Proceedings of IEEE/FUZZ'96*, New Orleans, September 1996.

[16] S. Miyamoto, *Fuzzy Sets in Information Retrieval and Cluster Analysis*, Boston, MA: Kluwer Academic Publishers, 1990.

[17] E. Rasmussen, Clustering Algorithms, In W.B. Frakes, R. Baeza-Yates (Eds), *Information Retrieval: Data Structures & Algorithms*, Englewood Cliffs, NJ: Prentice Hall, 1992.

[18] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Reading, MA: Addison-Wesley, 1989

[19] http://www.dtic.dla.mil/iac/cseriac/cashe/cashe.htm#edc

[20] L.A. Zadeh, Fuzzy sets, *Information and Control* (8), 1965, pp. 338-353.