2009

Comparison of Fuzzy Clustering Methods and Their Applications to Geophysics Data

David J. Miller
University of Nebraska-Lincoln

Carl A. Nelson
University of Nebraska-Lincoln, cnelson5@unl.edu

Molly Boeka Cannon
University of Nebraska-Lincoln

Kenneth P. Cannon
Utah State University

Hindawi Publishing Corporation
Applied Computational Intelligence and Soft Computing
Volume 2009, Article ID 876361, 16 pages
doi:10.1155/2009/876361

Research Article

Comparison of Fuzzy Clustering Methods and Their Applications to Geophysics Data

David J. Miller,[1] Carl A. Nelson,[1,2] Molly Boeka Cannon,[3] and Kenneth P. Cannon[4]

[1] *Department of Mechanical Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588-0656, USA*
[2] *Department of Surgery, University of Nebraska Medical Center, Omaha, NE 68198-4075, USA*
[3] *Geography and Geographic Information Science, School of Natural Resources, University of Nebraska-Lincoln, Lincoln, NE 68583-0961, USA*
[4] *USU Archaeological Services and Department of Sociology, Social Work, and Anthropology, Utah State University, Logan, UT 84322-0730, USA*

Correspondence should be addressed to Carl A. Nelson, cnelson5@unl.edu

Received 26 June 2009; Revised 9 December 2009; Accepted 24 December 2009

Recommended by Miin-Shen Yang

Fuzzy clustering algorithms are helpful when there exists a dataset with subgroupings of points having indistinct boundaries and overlap between the clusters. Traditional methods have been extensively studied and used on real-world data, but require users to have some knowledge of the outcome a priori in order to determine how many clusters to look for. Additionally, iterative algorithms choose the optimal number of clusters based on one of several performance measures. In this study, the authors compare the performance of three algorithms (fuzzy c-means, Gustafson-Kessel, and an iterative version of Gustafson-Kessel) when clustering a traditional data set as well as real-world geophysics data that were collected from an archaeological site in Wyoming. Areas of interest in the were identified using a crisp cutoff value as well as a fuzzy α-cut to determine which provided better elimination of noise and non-relevant points. Results indicate that the α-cut method eliminates more noise than the crisp cutoff values and that the iterative version of the fuzzy clustering algorithm is able to select an optimum number of subclusters within a point set (in both the traditional and real-world data), leading to proper indication of regions of interest for further expert analysis

1. Introduction

Archaeologists have employed various techniques of geophysical survey for over a half century with much of this pioneering work completed in Europe, particularly on Roman sites in England (see [1, 2] for review of this work). Today geophysical survey is widely used in European archaeology and is gaining popularity in North America [3]. The current work at the Goetz site in northwestern Wyoming (Figure 1) utilizes a magnetic gradiometer survey in order to detect subsurface features or areas of past human activity. Gradiometer survey tools like those shown in Figure 2 operate by directing a magnetic field into the matrix (soil) and reading the strength of the magnetic field that is returned from the matrix. An archaeological feature may be detected by the instrument if it has contrast to the matrix in which it resides. Such features may include hearths, house pits, storage pits, and other ground-disturbing activities left by inhabitants of the region.

An ideal setting would be one in which the matrix has little or no magnetic signature (and is uniform), and the archaeological feature has a significant magnetic signature. This would create high contrast between matrix and features and allow for high visibility of the feature, allowing an area of interest to emerge in the data analysis. However, this is rarely the case, and it is the challenge of geophysics researchers to recognize and sort out the features or areas of interest from the noise present in the matrix. Noise is also contributed by the presence of foreign materials (plants, rocks, refuse, etc.) in the scan region or irregularities in the terrain of the scan region (Figure 3). It is for these reasons that the investigators of this current work employed fuzzy clustering techniques to the collected gradiometer data.

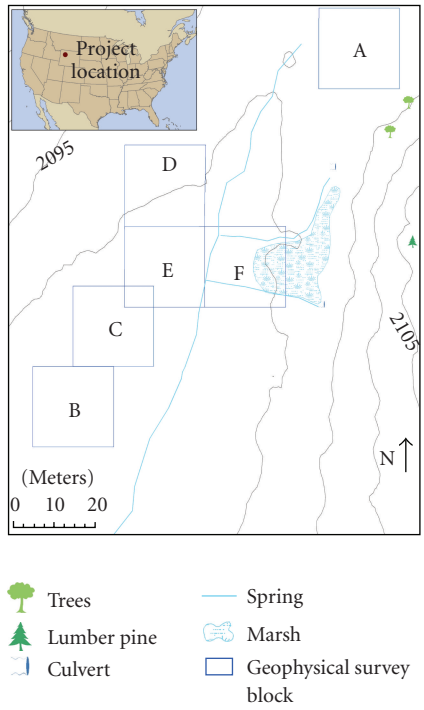

Figure 1: Map of the Goetz site near Jackson Hole, WY. Square areas A–F represent scanned regions discussed throughout this paper.

Figure 2: FM36 Magnetic Gradiometer by Geoscan Research (Bradford, West Yorkshire, UK). This handheld device is used to scan for subsurface archaeological features.

By partitioning a collection of data points into smaller subgroups, one is performing a clustering technique. Hard clustering is when each data point is uniquely assigned to one and only one cluster, while fuzzy clustering assigns a membership value to each point in each possible cluster and assigns the point to the cluster in which it has the highest "membership." Fuzzy clustering can be thought of as a precursor to hard clustering, since the end result is often partitioning of the data points into sets for control or categorization purposes. The problem with hard clustering is that it is assumed that the boundaries between groups are well defined, while this is not the case with many, in fact most, natural systems.

Fuzzy clustering is based on the notion of fuzzy sets as proposed by Zadeh in 1965 [4], which uses analogs to traditional set theory to combine and compare points in various groups with imprecision in the boundaries between the sets. This inherent imprecision makes fuzzy clustering ideal for emerging fields such as clustering and classification of geophysics data, in which the boundaries between locations of interest and the surrounding material are imprecise at best.

Two traditional methods of fuzzy clustering are the so-called fuzzy c-means (FCM) and Gustafson-Kessel (G-K) algorithms. FCM was originally proposed by Dunn [7] and was further refined by Bezdek [8], while G-K was developed in 1978 by Gustafson and Kessel [9]. The two methods are similar in that they use a distance measure to compute the cluster partitions and assign points to clusters; however, while FCM uses a norm-inducing identity matrix to compute the distances, G-K uses a cluster covariance matrix in the distance calculation, making it a subclass of FCM [5].

A third clustering method, similar to work done by Gath and Geva [6], is based on G-K, but does not assume any prior knowledge about the number or nature of subclusters present, which is not always available when analyzing real-world data. Rather than guessing at a number of clusters, the user defines a maximum number of clusters

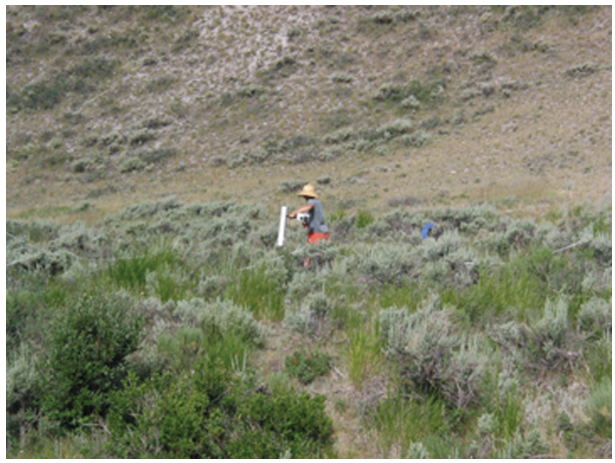

Figure 3: Using the handheld gradiometer in the field introduces a wide range of nonlinearity. Flora, terrain, and user inconsistencies all add to uncertainty inherent in the measurements.

(K) and the algorithm iteratively progresses from 2 to K, and outputs the cluster centers and partition matrix of the cluster with the best performance based on one of a number of validation methods (discussed more in the next section).

The purpose of this investigation was to compare these three similar, yet unique clustering algorithms, first on a standard data set, and then on geophysics data for detection of archaeological features. Specifically, the standard data set is the well-known Iris data initially gathered by Anderson in 1935 [10] and published by Fisher the following year [11]. This data set is used as a benchmark for comparison.

There are two reasons for this analysis. First, since custom codes for the standard and iterative G-K algorithms were written or modified in MATLAB (The MathWorks, Inc., Natick, MA) for this study, some measurement of their effectiveness was needed (the MATLAB function **fcm** was used as a baseline so validation of that code was deemed unnecessary), and the Iris data was used as a "debugging," validation and benchmarking tool. The second reason for validation was that in 1999 Bezdek published a correspondence indicating that there were multiple distinctly different versions of the Iris data that have been used as data sets in various published reports [12]. For this study, the original Iris data from [11] were carefully transcribed from the original work and independently checked for errors by a number of individuals to ensure that the correct data were, in fact, being used. This is intended to provide a uniform comparison of the three clustering methods.

The remainder of this paper is organized as follows. Section 2 will cover the three different clustering algorithms and experimental setup, while the validation using the Iris data is presented in Section 3. Section 4 shows the results of applying the clustering algorithms to real-world geophysics data to determine the presence and location of archaeological "anomalies." Section 5 will provide a summary and statement of conclusions as well as highlight future directions for this research.

2. Clustering Methods

Each of the three algorithms presented in the following section follow a similar structure: (1) select initial cluster centers, (2) calculate the distances between all points and all cluster centers, (3) update the partition matrix until some termination threshold is met. The differences lie in the way the algorithms perform steps (1) and (2), where they each derive their strengths. These differences will be discussed below. FCM and G-K have been extensively studied in the literature, so only a brief review of these two methods is presented below.

2.1. Fuzzy c-Means. The FCM algorithm is shown in Figure 4 as adapted from [5]. The purpose of the algorithm is to satisfy (1). Cluster centers (prototypes) are calculated during each iteration as the mean of the points in each sub-cluster, and the initial partition matrix, U, is randomly assigned at the beginning of the algorithm. The algorithm repeats until the difference between partition matrices $U^{(l)}$ and $U^{(l-1)}$ (the lth and $l-1$st iteration, resp.) is less than ε. As the weighting exponent, $m \to \infty$, the fuzziness of the function increases, causing feature vectors with low membership, μ, to contribute less to the overall weighting of the partition [13]; m is typically set equal to 2, as was the case in this study. If the error never reached below ε (10^{-5} in this study), the maximum number of iterations was set to 100 as a second termination criterion

$$\text{Minimize } J_m(U, v) = \sum_{k=1}^{N}\sum_{i=1}^{c}(\mu_{ik})^m D_{ikA}^2. \tag{1}$$

2.2. Gustafson-Kessel Algorithms. When comparing Figures 4 and 5, representing the FCM and Gustafson-Kessel algorithms, respectively [5], one should immediately notice many similarities. The parameters m, ε, μ, and J are all the same between the two algorithms. The major difference is in the calculation of the distances between the points in the data set and the cluster centers. Whereas in FCM the

Given the data set **Z**, choose the number of clusters $1 < c < N$, the weighting exponent $m > 1$, the termination tolerance $\varepsilon > 0$ and the norm-inducing matrix **A**.

Repeat for $l = 1, 2, \ldots$

Step 1: **Compute the cluster prototypes (means):**

$$v_i^{(l)} = \frac{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m Z_k}{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m}, \quad 1 \le i \le c.$$

Step 2: **Compute the distances:**

$$D_{ikA}^2 = \left(Z_k - v_i^{(l)}\right)^T A \left(Z_k - v_i^{(l)}\right), \quad 1 \le i \le c,\ 1 \le k \le N.$$

Step 3: **Update the partition matrix:**

if $D_{ikA} > 0$ for $1 = i = c, 1 = k = N$,

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} \left(D_{ikA}/D_{jkA}\right)^{2/(m-1)}}$$

otherwise

$$\mu_{ik}^{(l)} = 0 \text{ if } D_{ikA} > 0, \text{ and } \mu_{ik}^{(l)} = \in [0, 1] \text{ with } \sum_{i=1}^{c} \mu_{ik}^{(l)} = 1$$

until $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$.

Figure 4: Fuzzy c-means algorithm from [5].

distance measure uses a norm-inducing (identity) matrix, G-K uses a parameter based on the covariance matrix of each cluster, allowing the distance norm to adapt to the shape of the subclusters to best suit the data [5]. According to [9], the term M_j (equivalent to $\rho_i \det(F_i)^{1/n} F_i^{-1}$ from Figure 5) is symmetric and positive-definite, allowing the algorithm to adjust to conditions when feature dimensions are scaled differently; thus the algorithm can adjust to variations in the shape of each sub-cluster.

2.3. Iterative Gustafson-Kessel. The Unsupervised Fuzzy Partition-Optimal Number of Classes (UFP-ONC) algorithm by Gath and Geva [6] is an attempt to further optimize G-K clustering (and by extension FCM). In both G-K and FCM, one must know something about the inherent divisions among the data in order to provide the algorithm with the number of subclusters present in the data. To augment a traditional clustering algorithm, Gath and Geva added an iterative loop to the algorithm that, rather than a fixed number of subclusters, uses a maximum number of clusters and one of several performance measures to determine the optimal number of subclusters within the data.

The algorithm (Figure 6) has many similarities to FCM and G-K: centroids (v) are calculated as the mean of points within a cluster, points are assigned to a cluster using a partition matrix, U, termination is determined using a criterion, ε, and a distance measure is employed that makes use of the covariance of the cluster members. Rather than the traditional Euclidian distance measure, Gath and Geva employed the so-called "exponential" distance measure, d_e^2, which is intended to accommodate hyperellipsoidal clusters with variable densities. However, rather than automatically subdividing the data into K subclusters, UFP-ONC starts at $k = 2$ clusters ($k = 1$ can be ignored since it represents a sub-cluster consisting of the entire universe of discourse) and proceeds up to some user-defined maximum, $K < N$. Termination of the algorithm for a given maximum number of clusters occurs when the maximum difference between U^i and U^{i-1} is less than ε or when a maximum number of iterations is reached.

While the exponential distance measure is definitely worthy of further study and could provide better separation between the nonspherical, variable density anomaly clusters present in geophysics data, the algorithm was not used here in favor of an iterative version of the standard G-K algorithm. The first reason was that, despite its possible benefits, using the exponential distance measure would introduce another level of complexity and possible source of error to this current study (since comparative analysis of the three methods is more meaningful if a similar distance measure is used). Secondly, the functionality of the UFP-ONC algorithm is very similar to G-K (as seen in Figures 5 and 6), so very little extra information would be gained by using the exponential distance measure. Finally, since the partition and covariance matrices naturally produced by G-K can be used to calculate the various performance measures as used in [6] there is little reason not to simply add an iterative

Given the data set **Z**, choose the number of clusters $1 < c < N$, the weighting exponent $m > 1$, the termination tolerance $\varepsilon > 0$.

Repeat for $l = 1, 2, \ldots$

Step 1: **Compute the cluster prototypes (means):**

$$v_i^{(l)} = \frac{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m Z_k}{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m}, \quad 1 \le i \le c.$$

Step 2: **Compute the cluster covariance matrices:**

$$F_i = \frac{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m \left(Z_k - v_i^{(l)}\right)\left(Z_k - v_i^{(l)}\right)^T}{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)}\right)^m}, \quad 1 \le i \le c.$$

Step 3: **Compute the distances:**

$$D_{ikA_i}^2 = \left(Z_k - v_i^{(l)}\right)^T \left[\rho_i \det\left(F_i\right)^{1/n} F_i^{-1}\right] \left(Z_k - v_i^{(l)}\right), \quad 1 \le i \le c,\ 1 \le k \le N.$$

Step 4: **Update the partition matrix:**

if $D_{ikA_i} > 0$ for $1 = i = c$, $1 = k = N$,

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} \left(D_{ikA_i}/D_{ikA_i}\right)^{2/(m-1)}}$$

otherwise

$$\mu_{ik}^{(l)} = 0 \text{ if } D_{ikA_i} > 0, \text{ and } \mu_{ik}^{(l)} \in [0,1] \text{ with } \sum_{i=1}^{c} \mu_{ik}^{(l)} = 1$$

until $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$.

Figure 5: Gustafson-Kessel algorithm from [5].

step to G-K. For this study, $m = 2$ and $\varepsilon = 10^{-5}$ for both the iterative and noniterative versions of G-K.

During each iteration, one of three different performance measures is calculated; the best fuzzy partition matrix is the one which optimizes the functions shown in (2)–(4). All three measures take into account sub-cluster hypervolume, and two of them accommodate point density [6]. Equation (2) is the "fuzzy hypervolume," (3) is the "average partition density," and (4) is the "partition density," where F_i is the covariance matrix. Optimal volumetric measures are minima while density measures are maxima. Gath and Geva showed that the FHV criterion exhibited a clear minimum for most cases they studied; however, as the clusters began to overlap more and more or as the compactness of clusters began to vary, the density criteria would provide a better measure of performance. Such a result was expected when analyzing Fisher's Iris data

$$F_{\text{HV}} = \sum_{i=1}^{K} [\det(F_i)]^{1/2}, \tag{2}$$

$$D_{\text{PA}} = \frac{1}{K} \sum_{i=1}^{K} \frac{\sum_{j=1}^{N} \mu_{ij}}{[\det(F_i)]^{1/2}}, \tag{3}$$

$$P_D = \frac{\sum_{i=1}^{K} \sum_{j=1}^{N} \mu_{ij}}{F_{\text{HV}}}. \tag{4}$$

2.4. Experimental Setup. In addition to the settings for the three algorithms and the different algorithms themselves, as discussed above, there were several other parameters that were varied throughout the study. The first of these parameters was the method of segregating points of interest from background and noise (induced from irregularities in the soil, terrain effects and vegetation; see Figure 3) in the scans. From an in situ scan, geophysical magnetometer data ranges from approximately −180 to 2000 nT, with features of interest falling somewhere between −3 and −5 nT, with some variability in the ranges of interesting features. Due to this variation, a fuzzy segregation method was used to determine which data points were of interest and was compared to a crisp cutoff at −3 and −5 nT. The membership function used as a cutoff is shown in Figure 7. A membership value of 0.98 (similar to an α- or λ-cut [14]) was used to identify the anomaly points for this study.

Given the data set **X**, choose the number of clusters $1 < K < N$, the weighting exponent $m > 1$, the termination tolerance $\varepsilon > 0$.

Repeat for $k = 1, 2, \ldots, K$

Step 1: **Choose primary centroids** V_i**:**

Step 2: **Compute membership of all feature vectors in each cluster:**

$$\mu_{ij} \approx h\left(i \mid X_j\right) = \frac{1/d_s^2\left(X_j, V_j\right)}{\sum_{k=1}^{k} 1/d_s^2\left(X_j, V_j\right)}, \quad 1 \le i \le N,\ 1 \le j \le K$$

$$d_s^2\left(X_j, V_j\right) = \frac{\left[\det\left(F_i\right)\right]^{1/2}}{P_i} \exp\left[\frac{\left(X_j - V_i\right)^T F_i^{-1}\left(X_j - V_i\right)}{2}\right]$$

$$P_i = \frac{1}{N}\sum_{j=1}^{N} h\left(i \mid X_j\right)$$

$$F_i = \frac{\sum_{j=1}^{N} h\left(i \mid X_j\right)\left(X_j - V_i\right)\left(X_j - V_i\right)^T}{\sum_{j=1}^{N} h\left(i \mid X_j\right)}$$

While $\max_{ij}[|\mu_{ij} - \hat{\mu}_{ij}|] > \varepsilon$

Step 3: **Compute new centroids,** $\hat{V}_i$**, and update memberships,** μ_{ij} **to** $\hat{\mu}_{ij}$**.**

$$\hat{V}_i = \frac{\sum_{j=1}^{N}\left(\mu_{ij}\right)^m X_j}{\sum_{j=1}^{N}\left(\mu_{ij}\right)^m}$$

Figure 6: Unsupervised optimal fuzzy clustering algorithm from [6].

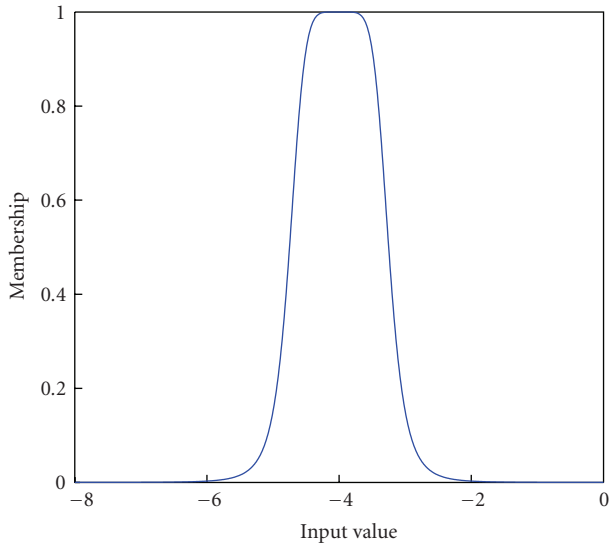

Figure 7: Fuzzy membership function used to determine points of interest from raw geophysics data. MATLAB function: gbellmf(x, $[0.75, 3, -4]$).

The parameters used for the **gbellmf** MATLAB function include the dataset and a vector describing the position and size of the membership function being described. The descriptive parameters used were $[0.75, 3, -4]$. Various combinations were tried to find the optimum combination. These changes are detailed in the appendix.

The second parameter that was changed was the maximum number of clusters to be considered. While G-K and FCM each use the full number of clusters indicated by the user, the iterative G-K algorithm can use any number of clusters, from 2 to K, as possible best partitions. Each magnetometer data file was processed for three clusters and for a maximum of 10 clusters, since these were the values used for the Iris validation in the following section. G-K and FCM results are compared to Iterative G-K results to see if certain clusters are always represented, regardless of how many subclusters are present.

Four data files were obtained from a geophysics study of the Goetz site in northwestern Wyoming using a Fluxgate FM/36 magnetic gradiometer, a standard geologic scanning tool (Figure 2). Scans from Figures 10(a)–10(c) were of 20 × 20 meter grids A, B, and C indicated in Figure 1 while the scans represented in Figure 10(d) were from 20 × 20 m grids D, E, and F. At each site, readings were taken at 0.125 m intervals with a transverse interval of 0.25 m. After the storage capacity of the Fluxgate FM/36 was met, the data were downloaded to a laptop computer and converted to comma-delimited files using Geoplot 3.00 software (Geoscan Research, Bradford, West Yorkshire, UK). These raw data files

were then input to the MATLAB program containing the algorithms described above.

3. Fisher's Iris Data

In 1999, Bezdek et al. published a correspondence called "Will the *Real* Iris Data Please Stand Up?" [12]. This correspondence pointed to a series of minor errors in reported values of data collected by Anderson in 1935 [10] and reported first by Fisher in 1936 [11]. These data points have been used extensively throughout the literature to provide a baseline for clustering algorithms, but if errors were present in the data, the results might not be as strong as they would otherwise have been. In an effort to validate the current work and attempt to eliminate the confusion in the community about which data set is "authentic," the Iris data were copied directly from Fisher's original work and then compared to that reported by Bezdek, both digitally and by hand. After a number of independent verifications, it was found that the results agree, so there is high probability that the data originally reported by Fisher in 1936 are being used here. The data are reported here as Table 1, for the sake of completeness and disclosure.

With the data accuracy question resolved, the validation of the algorithms was the next task. The MATLAB function, **fcm**, was used as a benchmark for the other two algorithms. Since prior information was known about the data set, it was assumed that there were three clusters for both G-K and FCM, since each uses a fixed number of clusters. Maxima of 3 and 10 subclusters were assumed for the iterative G-K algorithm to test for accuracy of clustering and efficacy of performance measures. The results are shown in Table 2. To get an idea of the magnitude of errors to be expected: Bezdek reported that for unsupervised algorithms, as these are, one can expect anywhere from 10 to 15 errors in the cluster assignments of the Iris data [15], which were used as a basis for current comparisons. It can be seen that, with the single exception of the iterative G-K algorithm with FHV performance measure and maximum of 10 subclusters, all of the algorithms found partitions that group points optimally, within the range of errors reported by Bezdek.

In all but one case from Table 2 (Iterative G-K, FHV, max = 10), the algorithms found all three clusters in almost identical locations (see Tables 3 and 4), which are similar to the results reported in [11]. The one difference, as one should notice when considering the last column of Table 2, is that the FHV performance measure resulted in 9 subclusters when the maximum number of clusters was set to 10. A summary of the performance of this measure with respect to increasing maximum number of clusters to be considered is shown in Table 4. The nine cluster centers obtained using Iterative G-K, FHV, max = 10 are shown in Table 5.

Noticing that there is a pattern to the resultant cluster centers shown in Table 5, this begs the question, "What would happen if the results were reclustered?" Keeping this in mind, the 9 cluster centers from Table 5 were run through the same clustering algorithm that produced them (Iter. G-K, FHV, max = 10) to produce the resultant cluster centers shown in the last four columns of Table 3 (there is no way to compare the results of 9 subclusters with the correct values obtained with 3 subclusters). Though these reclustered results do not perform as well as the rest of the algorithms represented in Table 2, the outcome is improved since there are three subclusters with similar centers to those shown throughout Table 3. These results, as well as the poor overall FHV performance measure, are exactly as predicted by Gath and Geva [6] for overlapping clusters as discussed in the previous section. Given these results and the similarity to other outcomes reported in the literature, the authors believe that the algorithms perform satisfactorily and are ready for use in clustering real-world data.

4. Clustering Geophysics Data

Data gathered at the Goetz site in northwestern Wyoming were gathered during the 2002 and 2003 summer field seasons using the Fluxgate FM/36 magnetic gradiometer. A graphical representation of the data is shown in Figure 8 along with an expert's opinion about what regions represent areas of interest that the fuzzy system should identify. Black areas of the figures represent unscanned areas or foreign objects (nonartifact metal) in the field of view of the scanner.

4.1. Fuzzy versus Crisp Cutoff. When the various clustering algorithms were applied to actual geophysics data, there was a wide range of different results. The first difference had to do with the cutoff method used: fuzzy or crisp. Figure 9 shows the results of processing the data files using the crisp cutoff values -5 to -3, and Figure 10 shows the results of processing the data files with a fuzzy cutoff using the membership function shown in Figure 7 at $\alpha = 0.98$. Table 6 quantitatively shows the number of clusters present in each of the different variations of the parameters. Though the data in Table 6 appear to show little difference between fuzzy and crisp cutoff values, the major dissimilarity is in the number of data points remaining after cutoff (shown in Table 7).

Though it might be possible to further tune the crisp cutoff values to further limit the number of "noise" points in the data set, the data in Table 7 illustrate that increasing the α-cut does not add a significant improvement to the resulting data set. Additionally, the authors feel that the inherent ability of fuzzy membership functions to handle nonlinearity makes it an ideal choice for an investigation of this type. The archaeological experts recommended the cutoff values of "about" $[-5, -3]$, which should immediately indicate "fuzzy logic" to anyone familiar with the technique. Also, since the data is being collected in a noisy environment, operator error is unavoidable, and there will always be the possibility of outliers, a fuzzy membership cutoff methodology is ideal. Moreover, future versions of this system could incorporate more fuzzy membership functions to further eliminate outliers. For example, a fuzzy inference system could be set up to accommodate the position of possible anomalies within the area of interest, in essence "preclustering" data points based on magnetometer reading and location (similar to a nearest-neighbor algorithm). Attempting this kind of

Table 1: Fisher's Iris data [11] that were validated and compared to that reported by Bezdek et al. [12].

Iris Sestosa				Iris Versicolor				Iris Virginica			
Sepal lengh	Sepal width	Petal length	Petal width	Sepal lengh	Sepal width	Petal length	Petal width	Sepal lengh	Sepal width	Petal length	Petal width
5.1	3.5	1.4	0.2	7.0	3.2	4.7	1.4	6.3	3.3	6.0	2.5
4.9	3.0	1.4	0.2	6.4	3.2	4.5	1.5	5.8	2.7	5.1	1.9
4.7	3.2	1.3	0.2	6.9	3.1	4.9	1.5	7.1	3.0	5.9	2.1
4.6	3.1	1.5	0.2	5.5	2.3	4.0	1.3	6.3	2.9	5.6	1.8
5.0	3.6	1.4	0.2	6.5	2.8	4.6	1.5	6.5	3.0	5.8	2.2
5.4	3.9	1.7	0.4	5.7	2.8	4.5	1.3	7.6	3.0	6.6	2.1
4.6	3.4	1.4	0.3	6.3	3.3	4.7	1.6	4.9	2.5	4.5	1.7
5.0	3.4	1.5	0.2	4.9	2.4	3.3	1.0	7.3	2.9	6.3	1.8
4.4	2.9	1.4	0.2	6.6	2.9	4.6	1.3	6.7	2.5	5.8	1.8
4.9	3.1	1.5	0.1	5.2	2.7	3.9	1.4	7.2	3.6	6.1	2.5
5.4	3.7	1.5	0.2	5.0	2.0	3.5	1.0	6.5	3.2	5.1	2.0
4.8	3.4	1.6	0.2	5.9	3.0	4.2	1.5	6.4	2.7	5.3	1.9
4.8	3.0	1.4	0.1	6.0	2.2	4.0	1.0	6.8	3.0	5.5	2.1
4.3	3.0	1.1	0.1	6.1	2.9	4.7	1.4	5.7	2.5	5.0	2.0
5.8	4.0	1.2	0.2	5.6	2.9	3.6	1.3	5.8	2.8	5.1	2.4
5.7	4.4	1.5	0.4	6.7	3.1	4.4	1.4	6.4	3.2	5.3	2.3
5.4	3.9	1.3	0.4	5.6	3.0	4.5	1.5	6.5	3.0	5.5	1.8
5.1	3.5	1.4	0.3	5.8	2.7	4.1	1.0	7.7	3.8	6.7	2.2
5.7	3.8	1.7	0.3	6.2	2.2	4.5	1.5	7.7	2.6	6.9	2.3
5.1	3.8	1.5	0.3	5.6	2.5	3.9	1.1	6.0	2.2	5.0	1.5
5.4	3.4	1.7	0.2	5.9	3.2	4.8	1.8	6.9	3.2	5.7	2.3
5.1	3.7	1.5	0.4	6.1	2.8	4.0	1.3	5.6	2.8	4.9	2.0
4.6	3.6	1.0	0.2	6.3	2.5	4.9	1.5	7.7	2.8	6.7	2.0
5.1	3.3	1.7	0.5	6.1	2.8	4.7	1.2	6.3	2.7	4.9	1.8
4.8	3.4	1.9	0.2	6.4	2.9	4.3	1.3	6.7	3.3	5.7	2.1
5.0	3.0	1.6	0.2	6.6	3.0	4.4	1.4	7.2	3.2	6.0	1.8
5.0	3.4	1.6	0.4	6.8	2.8	4.8	1.4	6.2	2.8	4.8	1.8
5.2	3.5	1.5	0.2	6.7	3.0	5.0	1.7	6.1	3.0	4.9	1.8
5.2	3.4	1.4	0.2	6.0	2.9	4.5	1.5	6.4	2.8	5.6	2.1
4.7	3.2	1.6	0.2	5.7	2.6	3.5	1.0	7.2	3.0	5.8	1.6
4.8	3.1	1.6	0.2	5.5	2.4	3.8	1.1	7.4	2.8	6.1	1.9
5.4	3.4	1.5	0.4	5.5	2.4	3.7	1.0	7.9	3.8	6.4	2.0
5.2	4.1	1.5	0.1	5.8	2.7	3.9	1.2	6.4	2.8	5.6	2.2
5.5	4.2	1.4	0.2	6.0	2.7	5.1	1.6	6.3	2.8	5.1	1.5
4.9	3.1	1.5	0.2	5.4	3.0	4.5	1.5	6.1	2.6	5.6	1.4
5.0	3.2	1.2	0.2	6.0	3.4	4.5	1.6	7.7	3.0	6.1	2.3
5.5	3.5	1.3	0.2	6.7	3.1	4.7	1.5	6.3	3.4	5.6	2.4
4.9	3.6	1.4	0.1	6.3	2.3	4.4	1.3	6.4	3.1	5.5	1.8
4.4	3.0	1.3	0.2	5.6	3.0	4.1	1.3	6.0	3.0	4.8	1.8
5.1	3.4	1.5	0.2	5.5	2.5	4.0	1.3	6.9	3.1	5.4	2.1
5.0	3.5	1.3	0.3	5.5	2.6	4.4	1.2	6.7	3.1	5.6	2.4
4.5	2.3	1.3	0.3	6.1	3.0	4.6	1.4	6.9	3.1	5.1	2.3
4.4	3.2	1.3	0.2	5.8	2.6	4.0	1.2	5.8	2.7	5.1	1.9
5.0	3.5	1.6	0.6	5.0	2.3	3.3	1.0	6.8	3.2	5.9	2.3
5.1	3.8	1.9	0.4	5.6	2.7	4.2	1.3	6.7	3.3	5.7	2.5
4.8	3.0	1.4	0.3	5.7	3.0	4.2	1.2	6.7	3.0	5.2	2.3
5.1	3.8	1.6	0.2	5.7	2.9	4.2	1.3	6.3	2.5	5.0	1.9
4.6	3.2	1.4	0.2	6.2	2.9	4.3	1.3	6.5	3.0	5.2	2.0
5.3	3.7	1.5	0.2	5.1	2.5	3.0	1.1	6.2	3.4	5.4	2.3
5.0	3.3	1.4	0.2	5.7	2.8	4.1	1.3	5.9	3.0	5.1	1.8

Table 2: Number of resultant Iris clusters and classification errors from each of the three algorithms using two different maxima for the iterative algorithm. *Number of errors from clustering of 9 resultant subclusters using FHV, max = 3.

Algo	FCM (3 clust.)	G-K (3 clust.)	Iter. G-K, DPA (max = 3)	Iter. G-K, PD (max = 3)	Iter. G-K, FHV (max = 3)	Iter. G-K, DPA (max = 10)	Iter. G-K, PD (max = 10)	Iter. G-K, FHV (max = 10)*
#Clusters	3	3	3	3	3	3	3	9
Errors	16	14	14	14	14	14	14	24

Table 3: Iris cluster centers reported by various algorithms and settings. *Cluster centers obtained from clustering of 9 resultant subclusters using FHV, max = 3.

Algo		FCM (3 clust.)				G-K (3 clust.)							
Center	1	5.004	3.4141	1.4828	0.25354	5.0147	3.4383	1.4663	0.24451				
	2	5.8888	2.761	4.3637	1.3972	6.1385	2.8024	4.534	1.4108				
	3	6.7748	3.0523	5.6466	2.0535	6.3928	2.9764	5.2897	2.0084				
Algo		Iter. G-K, DPA (max = 3)				Iter. G-K, PD (max = 3)				Iter. G-K, FHV (max = 3)			
Center	1	5.0147	3.4383	1.4663	0.24451	5.0147	3.4383	1.4663	0.24451	5.0147	3.4383	1.4663	0.24451
	2	6.1385	2.8024	4.534	1.4108	6.1385	2.8024	4.534	1.4108	6.1385	2.8024	4.534	1.4108
	3	6.3928	2.9764	5.2897	2.0084	6.3928	2.9764	5.2897	2.0084	6.3928	2.9764	5.2897	2.0084
Algo		Iter. G-K, DPA (max = 10)				Iter. G-K, PD (max = 10)				Iter. G-K, FHV (max = 10)*			
Center	1	5.0147	3.4383	1.4663	0.24451	5.0147	3.4383	1.4663	0.24451	5.0247	3.3975	1.5854	0.32499
	2	6.1385	2.8024	4.534	1.4108	6.1385	2.8024	4.534	1.4108	5.7601	2.6904	4.2685	1.2956
	3	6.3928	2.9764	5.2897	2.0084	6.3928	2.9764	5.2897	2.0084	6.5041	2.9663	5.1772	1.8804

modification with the crisp cutoff values would be much more complex and difficult to tune for novel situations. This will be discussed further in Section 5.

Finally, rather than having to individually tune the upper and lower bounds of the crisp cutoff range, the users can simply adjust the α-cut value, which lends itself well to a GUI slider or other UI objects that are readily available in most programming languages, including MATLAB. A slider would allow users in the field an intuitive means to adjust the search criteria "on the fly," to achieve the best outcome. Overall, the authors believe that for a system like this, fuzzy cutoff values are the best choice.

4.2. Maximum Clusters. As is shown in Table 6, both FCM and G-K use the maximum number of clusters possible, which is in accordance with the algorithms. Since the iterative G-K algorithm was designed specifically to optimize the number of clusters, it should come as no surprise to see that the number of resultant clusters is varied across the three different performance measures and maximum number of clusters. Quantitatively, this result is shown in Table 6, but the results were qualitatively different, as well. Due to space constraints, not all of the resulting plots can be shown here, but Figure 11 shows the extremes, both good and bad. In general, the results fall into one of the following categories:

(I) round or near-round "clouds" of points,

(II) some subclustering of areas of interest.

Ideally, points belonging to a well-defined area of interest would fall into class II, while random clouds of points that were "forced" into a sub-cluster would fall into class I.

Most of the traditional G-K and FCM results fell into class I, since they were forced to consider the maximum number of subclusters available (see Figure 11(e)). Some datasets, however, like Figure 11(b), exhibit a clear lower linear feature as shown in Figures 8(a) and 8(b). Unfortunately, the other two linear features and the round features in this data set do not show up as clearly. This is most likely due to the close similarity between the magnetometer readings of the surrounding soil and the features. Similar results can be seen when comparing Figures 11(d), 8(c), and 8(d). One can see the beginnings of the linear features running from the bottom-left to the top center and the round feature in the top right-hand corner of the grid.

4.3. Fuzzy Clustering and Initialization. In addition to the type of algorithm used to cluster the data, there are parameters that can be modified to affect the output of the analysis: the fuzziness exponent m, and the initialization of the cluster centers. The parameter "m" (sometimes "q") controls the fuzziness of the resulting clusters and can be used to cluster datasets with overlapping point sets [6] and in most cases, is set to 2.0. Using the experimental datasets as a trial, the fuzziness exponent was changed for each of the different clustering algorithms, ranging from 1.0 to 5.0. For the data sets tested, there was very little difference in the resulting output. Table 8 shows the minor differences in the output across the various values for "m" for the FCM algorithm. The other algorithms showed a similar pattern; therefore, it was deemed that for these data, the fuzziness parameter did not have a large effect, so was used at its default value, 2.0.

Table 4: Number of resultant clusters using Iterative G-K algorithm, fuzzy hypervolume performance measure and the maximum number of clusters shown in row 1.

Max	3	4	5	6	7	8	9	10
Actual	3	3	3	3	7	8	9	9

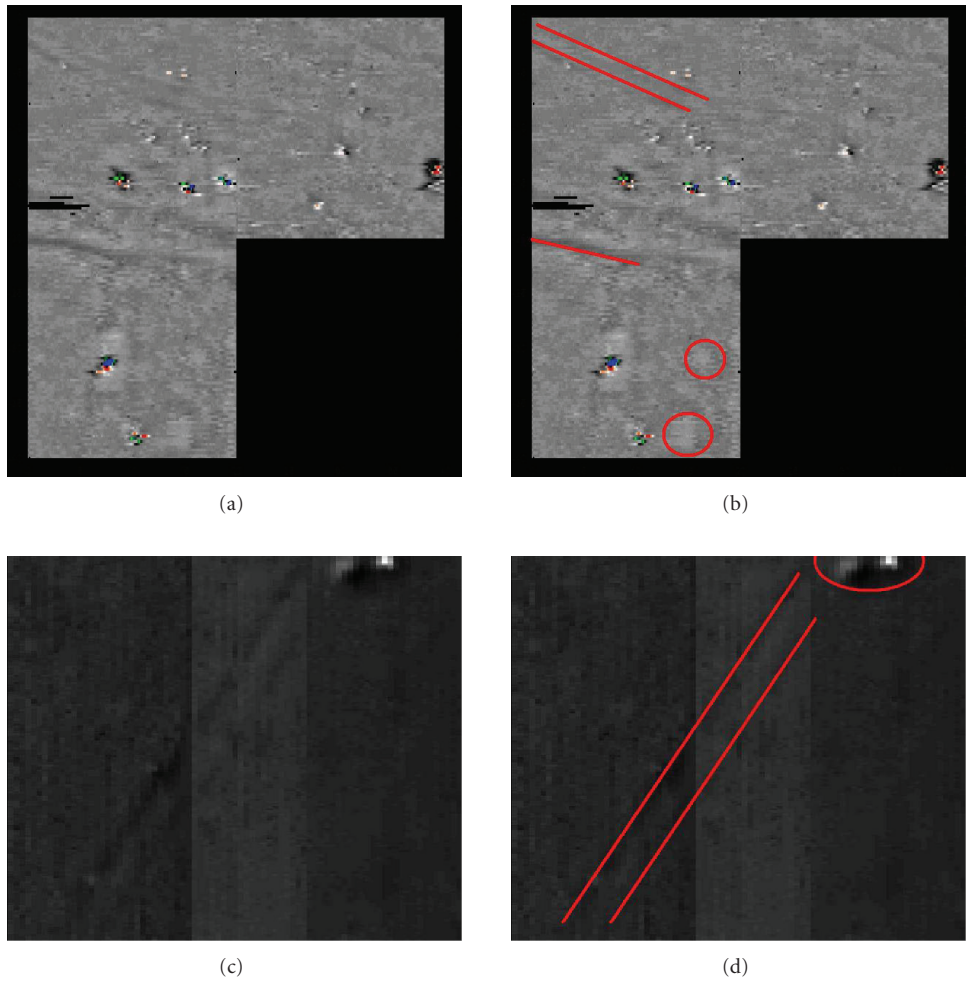

Figure 8: (a) Graphical representation of data gathered at Goetz grids D, E, and F. (b) Expert opinion on archaeological features of interest contained in magnetometer data from part (a). (c) Graphical representation of data gathered at Goetz, grid C. (d) Expert opinion on archaeological features of interest contained in magnetometer data from part (c). Red, blue, and green pixels represent readings of metallic objects in the scan field; red lines and circled areas represent features of interest.

Table 5: Cluster centers using Iterative G-K algorithm, fuzzy hypervolume performance measure and up to 10 clusters.

6.2262	2.9524	5.399	2.087
6.5788	2.9829	4.5774	1.4299
5.1406	3.3637	1.8641	0.53001
4.8242	3.3835	1.4217	0.24568
7.0132	2.9938	5.7007	2.0942
5.9606	2.8838	4.5994	1.3934
6.2008	2.9365	5.0255	1.9051
5.1093	3.4454	1.4704	0.19927
5.5596	2.497	3.9375	1.1978

The other parameter that can have an effect on the resultant clusters is the initial condition of each cluster center; however, with each of the algorithms in this study, the initial partition matrix is randomly initiated, eliminating this as a possible factor.

4.4. Expert Opinion. When consulting an expert about the results shown in these suboptimal clustering outcomes, the general feeling was positive. Despite the lack of any clear subclusters (Figures 10(a) and 10(b)), the software was deemed to be providing results that were no worse than traditional geophysics programs. Since the output of traditional programs shows every data point in grayscale, the

Table 6: Number of resultant clusters using three algorithms and three performance measures with respect to maximum number of clusters and fuzzy or crisp cutoff. Resulting 4 tuples are for different data files: $\langle A, B, C, D \rangle$.

Algorithm Max. Clusters	FCM 3	FCM 10	G-K 3	G-K 10	Iter. G-K, DPA 3	Iter. G-K, DPA 10	Iter. G-K, PD 3	Iter. G-K, PD 10	Iter. G-K, FHV 3	Iter. G-K, FHV 10
Fuzzy	3,3,3,3	10,10,10,10	3,3,3,3	10,10,10,10	2,2,3,3	2,2,8,3	2,2,3,3	2,2,8,3	3,2,2,3	3,8,10,10
Crisp	3,3,3,3	10,10,10,10	3,3,3,3	10,10,10,10	2,2,3,3	2,2,9,3	2,2,3,3	2,2,9,3	2,2,2,3	2,9,10,10

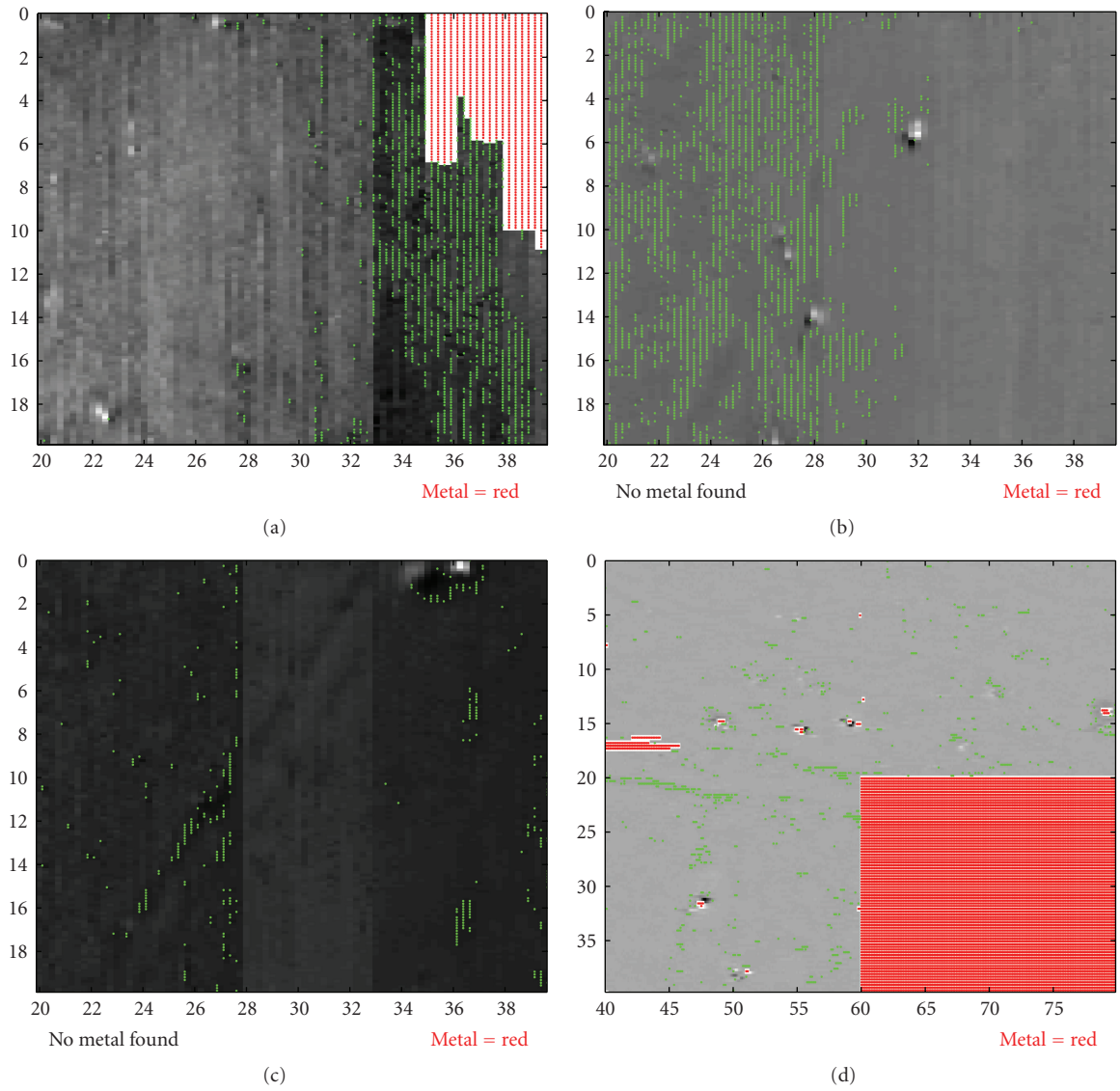

Figure 9: Data sets resulting from crisp cutoff values $[-5, -3]$. Subplots (a)–(d) represent different data files from different areas within the survey site (see Figure 1); (a) represents region A, (b) represents region B, (c) represents region C, and (d) represents regions D, E, and F (rotated 90° counterclockwise). Red pixels represent metallic objects and green pixels represent possible areas of interest.

addition of color to the resulting plot helps the investigator better visualize the regions that may fall within the range of interest. When no features of interest are present, as in Figures 10(a) and 10(b), the program shows the lack of features in a very clear manner; when features are present as in Figure 11, the clustering algorithm has been shown to clearly identify regions of interest (Figures 11(b) and 11(d)). The main problem with the algorithm is that with the settings as they are, there is no way to extract all the features without overestimating the number of points in the data set; however, this problem will be addressed in future versions of the software.

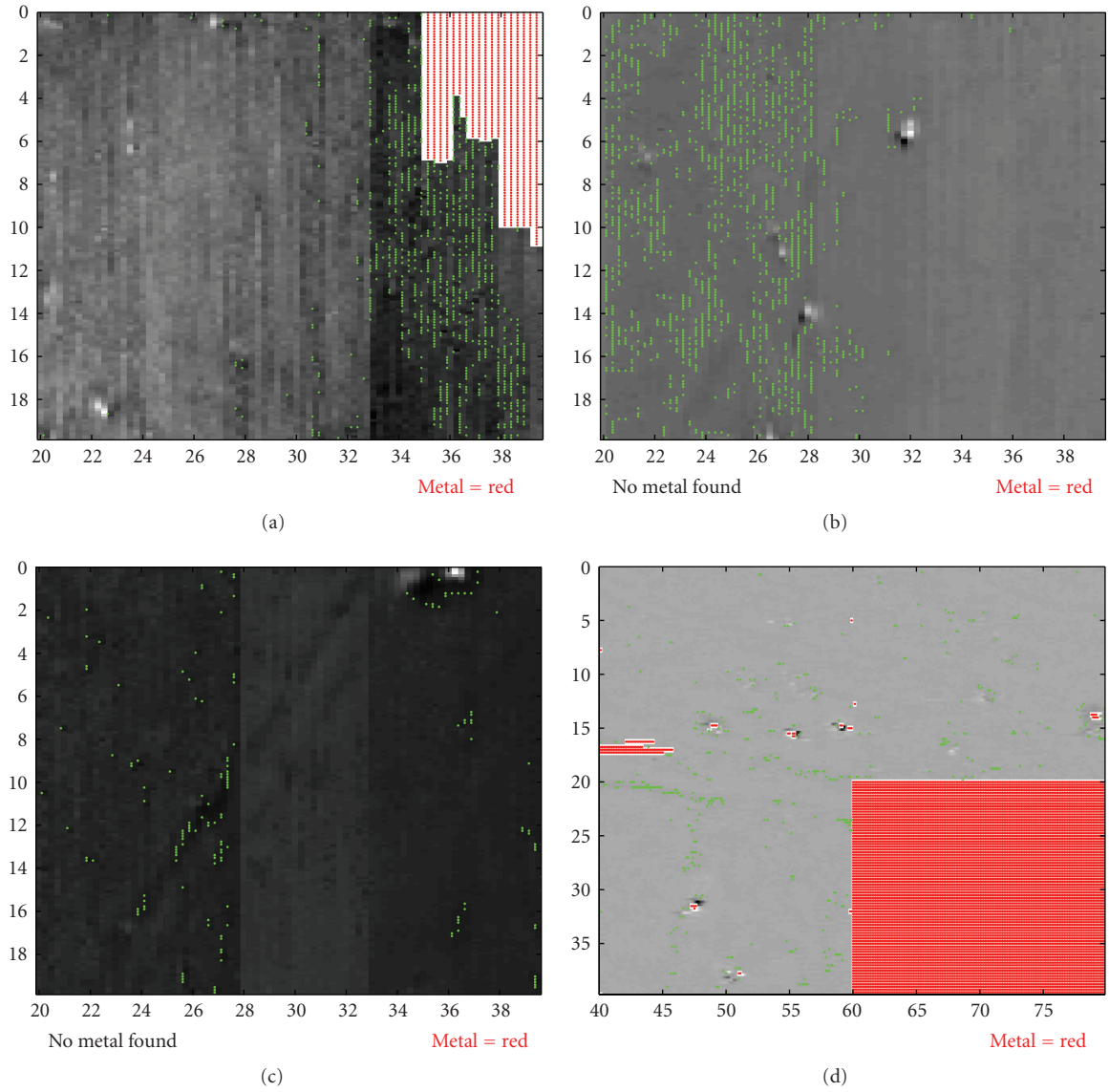

Figure 10: Data sets resulting from fuzzy cutoff values using the MATLAB function gbellmf(x, $[0.75, 3, -4]$). As in Figure 8, subplots represent different scan areas enumerated in Figure 1. For a more detailed description of **gbellmf**, see the appendix.

Table 7: Size of data sets after cutoffs using two fuzzy thresholds (α-cuts) and crisp cutoff values.

Dataset	A	B	C	D
Fuzzy ($\alpha = 0.98$)	971	1535	157	442
Fuzzy ($\alpha = 0.85$)	971	1535	157	442
Crisp	1593	2712	342	820
Factor	1.64	1.77	2.18	1.86

5. Conclusions and Future Work

Though these algorithms may be utilized in a number of different situations, the software used in this study was designed specifically for this application, so would only be of limited utility in other situations. As with any scientific endeavor, there are a number of different methods that could be used to segregate the data in this study. The authors chose the algorithms presented due to their proven track record and wide acceptance; however, there are alternatives that could be considered. For instance, works by Pal et al. [16] and Yang and Wu [17] expand on the traditional fuzzy c-means algorithm by adding different clustering methods, membership functions, and cluster validity indices. The complexity of these algorithms precluded their inclusion in the current work, but in order to expand the utility of this program to other problem types, their inclusion in future versions may be warranted.

Initially, it was observed that the clustering methods presented here work well on a data set with overlapping subclusters, namely, Fisher's Iris data. When running a noniterative measure like FCM or G-K with exactly the

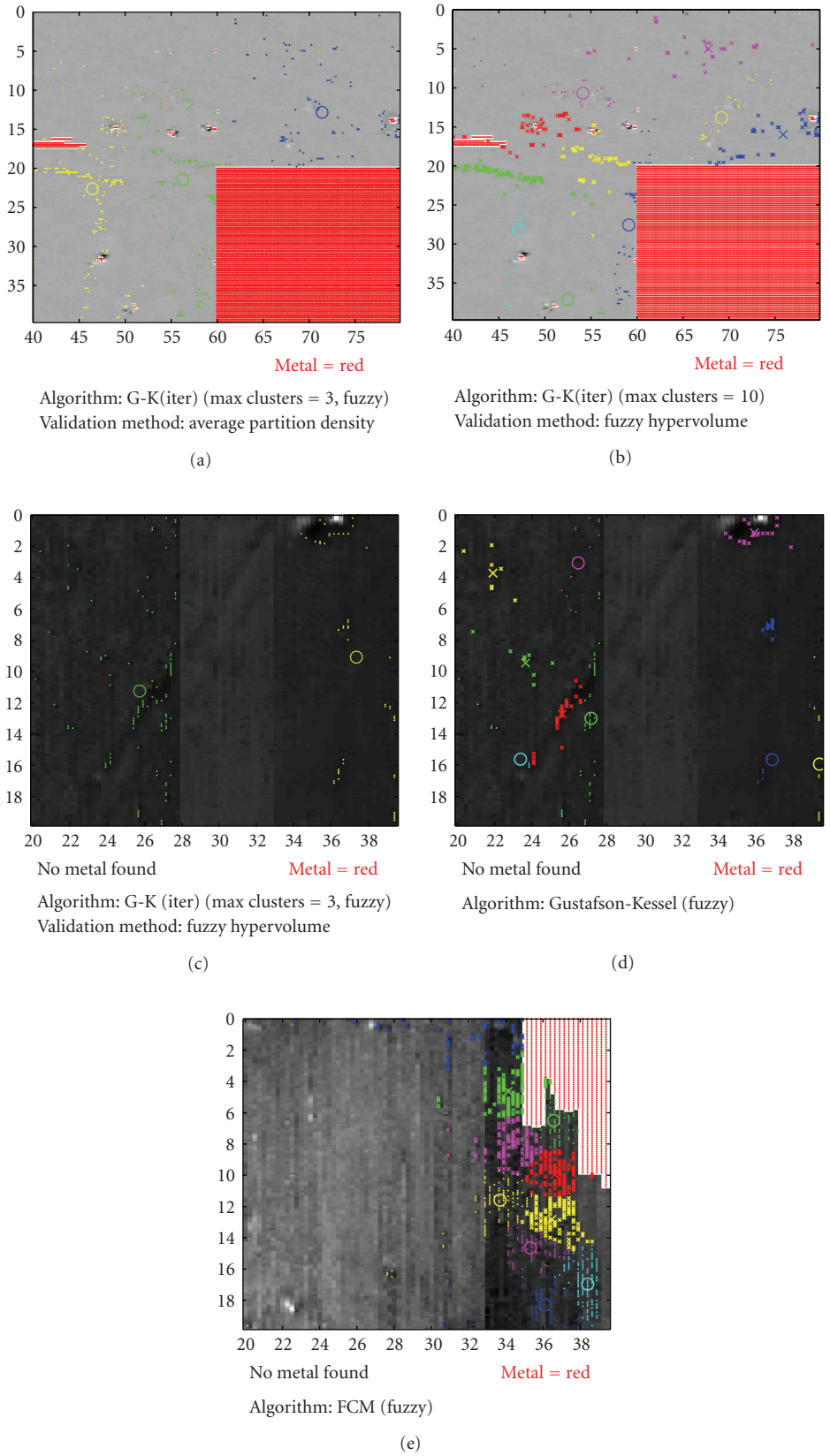

Figure 11: Representative results of various clustering algorithm runs: (a) Poor result: Region "D," iterative G-K, average partition density performance measure, maximum clusters = 3; (b) Good result: Region "D," iterative G-K, fuzzy hypervolume performance measure, maximum clusters = 10; (c) Poor result: Region "C," iterative G-K, fuzzy hypervolume performance measure, maximum clusters = 3; (d) Good result: Region "C," G-K, max clusters = 10; (e) Poor result: Region "A," FCM, max clusters = 10.

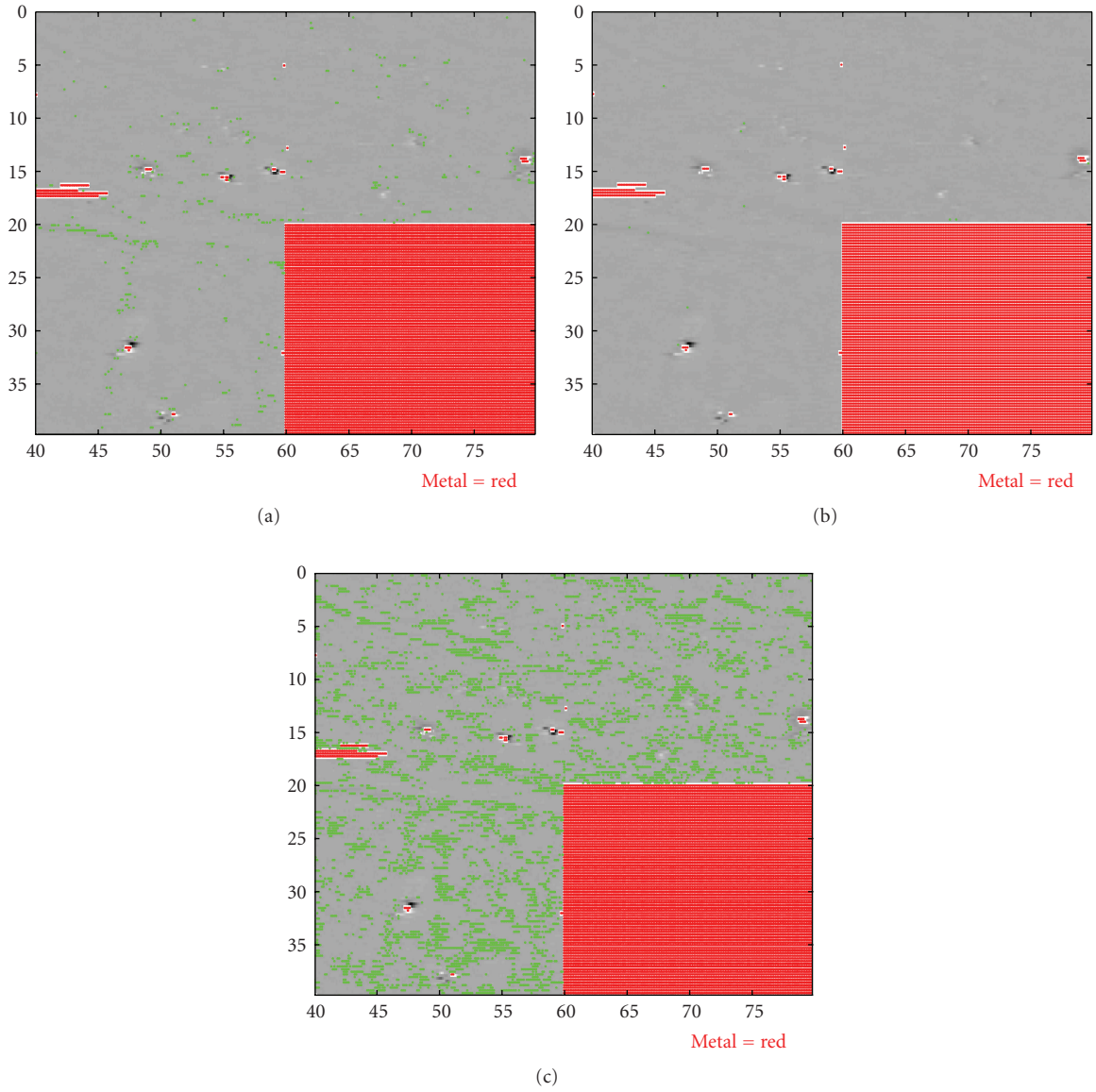

FIGURE 12: Representative results of various membership function values: (a) $A = 0.75$, $B = 3$, $C = -4$. These values were used in the study because they provide a good balance of eliminating outliers and high cluster compactness; (b) $A = 0.25$, $B = 1$, $C = -6$. These values were not used because of low, noncompact clusters (lowering individual A, B, or C values gave similar results); (c) $A = 1.25$, $B = 5$, $C = -2$. These values were not used because too many points remained (raising individual A, B, or C values gave similar results).

right number of subclusters, it was shown that the clusters approach what has been traditionally accepted as the appropriate centers and partitions. This was validated by the use of the original data from Fisher's work.

Secondly, it was found that the results of an iterative algorithm with an increasing number of subclusters agree with the results presented in [6]; namely, that as the overlap between clusters increases, the fuzzy hypervolume measure tends to provide suboptimal results when compared to either of the density-based performance measures. This was shown in Table 2. Because of this fact, the results shown in Figure 11(b) should be taken with some reservation, since they might reflect some of this poor performance; however, the features of interest in geophysics data are often well separated and do not exhibit the overlap present in the Iris data, so the poor performance of the fuzzy hypervolume measure may not be as great a factor. This subject definitely warrants further study before a stand-alone software package is taken into the field.

In general, it was shown that fuzzy clustering techniques are applicable to geophysics data gathered using techniques such as magnetometry, for purposes of sub-surface feature identification. The algorithms presented here represent only a small fraction of the types of clustering available, and all have room for improvement. Also, it was shown that fuzzy membership functions are applicable to this field since the fuzzy membership cutoff method provided fewer outlier points and fewer overall points than the crisp cutoff

Table 8: Cluster centers (X, Y) for various values of fuzziness parameter: $m = 1.1, 2.0, 2.5, 3.0, 5.0$.

fcm ($m = 1.1$)	fcm ($m = 2.0$)	fcm ($m = 2.5$)	fcm ($m = 3.0$)	fcm ($m = 5.0$)
12.38, 71.54	12.90, 72.60	13.06, 72.60	13.14, 72.34	12.42, 71.02
21.74, 45.68	35.64, 53.60	35.52, 54.31	35.35, 54.89	34.76, 55.05
20.13, 57.09	20.32, 57.80	19.75, 57.93	19.48, 57.87	19.15, 57.39
35.09, 53.18	21.09, 45.78	21.02, 45.74	20.98, 45.77	20.93, 46.03
11.56, 52.02	12.27, 53.24	13.18, 52.85	13.73, 52.60	15.27, 53.38

method for a similar range of values. Finally, a version of the "controversial" Iris data was presented that has been independently verified in an effort to eliminate the confusion between the different versions of this data set in the literature and unify the benchmark measure for comparing the three algorithms discussed.

There is still much work to be done before there is a stand-alone piece of software available to fully classify all features of interest in magnetometer scans. However, the authors feel that the current system has promise. The major reason for this feeling is the result shown in Figure 11(b). The lowest linear feature was identified without many extra outliers. The first major downfall, as one can plainly see, is the overabundance of noise and outlier points in the figure. This problem can be addressed in one of two different ways: adjusting the threshold values of the membership function, or creating additional membership functions comprising a fuzzy inference system (FIS) to identify which points are of interest and which are noise based not only on magnetometer readings, but also on proximity to other points. This FIS would act as a preemptive clustering or nearest-neighbors algorithm. This would aid in the elimination of outliers, possibly decrease computation time, and increase separation between the various clusters, resulting in an improved result.

The second major downfall is the opposite of the first. Namely, the features of interest have magnetometer readings too similar to the surrounding matrix readings. This leads to otherwise interesting points not being identified, as was the case with the vertical stripe of lighter points in Figure 8(c) and 8(d). The exact reason for this light portion was not known by the expert, but clearly caused the linear features to be lost to the software. It is believed that one reason for this problem is that the algorithms used in this case were unsupervised. The addition of some type of error back-propagation or running average magnetometer reading could further improve the results. Using the average magnetometer reading could help the system automatically identify what is a true feature of interest and what is not by selecting points for which the magnetometer value falls outside a standard deviation of the mean of the scan. This would be helpful for novel soil types or areas that have not previously been scanned, since the cutoff values are somewhat subjective. It might also help eliminate the problem just discussed relating to unforeseen changes in the scan characteristics.

In general it was found that, due to the nature of geophysics data (with round and linear features mixed together), Gustafson-Kessel is preferable to fuzzy c-means, since it is designed to handle both sub-cluster types. Also, using an iterative approach is advisable since most non-experts will have a hard time deciding what is and is not of interest on their own, but will use the software to determine what is worthy of further study; having said that, it is also advisable to overestimate the maximum number of clusters possible in order to avoid missing out on features of interest due to the nature of the algorithms. Finally, since the areas of interest in geophysics data are often well separated, using the fuzzy hypervolume performance measure will not lead to major problems as it would in a dataset with overlapping subclusters, like the Iris data. The iterative algorithm with a high maximum number of clusters and using the fuzzy hypervolume performance measure appeared to provide superior overall detection of features for the type of geophysics data being analyzed.

Appendix

The gbellmf$\{X, [A, B, C]\}$ membership function is based on an extension to the Cauchy probability distribution function (see (A.1)). Various combinations of the **gbellmf** membership function parameters were tested for optimum performance. The parameters used for this study were $[0.75, 3, -4]$. Other combinations were tried but were disregarded because they either left too many points in consideration or too few (see Figure 12)

$$\text{gbellmf}(X, [A, B, C]) = \frac{1}{1 + \text{abs}((X - C)/A)^{2B}}. \quad \text{(A.1)}$$

Acknowledgments

Funding for this project was provided by the Nebraska Tobacco Settlement Biomedical Research Development Funds. The US Fish and Wildlife Service, Earthwatch Institute, and the National Park Service provided funds for the collection of archaeological data. The authors would like to thank Dipika Singh for her help with assembling the various algorithm scripts for this paper and the reviewers for their insightful comments.

References

[1] M. J. Aitken, *Physics and Archaeology*, Interscience, New York, NY, USA, 1961.

[2] A. J. Clark, *Seeing beneath the Soil: Prospecting Methods in Archaeology*, T. Batsford, London, UK, 1996.

[3] K. L. Kvamme, "Geophysical surveys as landscape archaeology," *American Antiquity*, vol. 68, no. 3, pp. 435–457, 2003.
[4] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
[5] R. Babuska, *Fuzzy Modeling for Control*, Kluwer Academic Publishers, Norwell, Mass, USA, 1998.
[6] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 773–780, 1989.
[7] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.
[8] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer Academic Publishers, Norwell, Mass, USA, 1981.
[9] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proceedings of the IEEE Conference on Decision and Control (CDC '78)*, pp. 761–766, San Diego, Calif, USA, 1979.
[10] E. Anderson, "The irises of the Gaspe peninsula," *Bulletin of the American Iris Society*, vol. 59, pp. 2–5, 1935.
[11] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.
[12] J. C. Bezdek, J. M. Keller, R. Krishnapuram, L. I. Kuncheva, and N. R. Pal, "Will the real iris data please stand up?" *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 3, pp. 368–369, 1999.
[13] T. W. Cheng, D. B. Goldgof, and L. O. Hall, "Fast fuzzy clustering," *Fuzzy Sets and Systems*, vol. 93, no. 1, pp. 49–56, 1998.
[14] T. J. Ross, *Fuzzy Logic with Engineering Applications*, John Wiley & Sons, Hoboken, NJ, USA, 2004.
[15] J. C. Bezdek, E. C. K. Tsao, and N. R. Pal, "Fuzzy Kohonen clustering networks," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 1035–1043, 1992.
[16] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 4, pp. 517–530, 2005.
[17] M.-S. Yang and K.-L. Wu, "Unsupervised possibilistic clustering," *Pattern Recognition*, vol. 39, no. 1, pp. 5–21, 2006.