Journal of Uncertain Systems
Vol.3, No.4, pp.298-306, 2009
Online at: www.jus.org.uk

Implementation of the Extended Fuzzy C-Means Algorithm in Geographic Information Systems

Ferdinando Di Martino[1,2,*], Salvatore Sessa[1]
[1]*Università degli Studi di Napoli Federico II, Dipartimento di Costruzioni e Metodi Matematici in Architettura, Via Monteoliveto 3, 80134 Napoli, Italy*
[2]*Università degli Studi di Salerno, Dipartimento di Matematica e Informatica Via Ponte Don Melillo, 84084 Fisciano, Italy*

Received 28 March 2009; Revised 23 June 2009

Abstract

Density cluster methods have elevated computational complexity and are used in spatial analysis for the determination of impact areas. We propose the extended fuzzy c-means (EFCM) algorithm like alternative method because it has three advantages: robustness to noise and outliers, linear computational complexity and automatic determination of the optimal number of clusters. We implement the EFCM algorithm inside a geographic information systems (GIS) for the determination of buffer areas as hypersphere volume prototypes which are circles in the case of bidimensional pattern data. Indeed we have applied this algorithm in the spatial analysis of buffer areas called hotspots, including fire point-events of the Santa Fè district (NM), downloaded from http://www.fs.fed.us/r3/gis/sfe_gis.shtml.

Keywords: extended fuzzy c-means, fuzzy c-means, GIS, hotspot

1 Introduction

In spatial analysis a buffer area is an area at a specified distance around to features of a theme. This area is determined as a polygon by defining distance parameters that can be set as constants or variables, determined by feature attributes: for instance, circular buffer areas are obtained around a feature of the theme by using the radius of the circle as distance parameter. Buffer areas are calculated in many fields of the spatial analysis and they can determine dangerous bounded zones: for examples, areas around an epicenter of an earthquake, areas of industrial pollution, urban areas where the construction of buildings is forbidden from the local legislation.

The buffering primitive operations in a geographical information system (GIS) concern points, lines and polygons. In spatial analysis, an area having dimensions of a continent can be considered, with a good approximation, as a plane and we apply the Euclidean geometry in the calculus of distances. For this reason the buffer area around a point on the map is formed by a circle ("circular polygon" in terms of analysis spatial) centered in that point. For instance, the epicentre of an earthquake or the location of a criminal event can be represented from a point. The radius of this circle is called the buffer distance which is assumed by the user either as a constant value for all the point data or as the value of a field in the point data table. Moreover the user has two options: to separate these circular buffer areas (cfr. Fig.1) or to merge some of them by obtaining new polygonal areas (cfr. Fig.2).

When the number of event-points is elevated, the classical density methods are not suitable for the determination of impact areas because of high computational complexity. Then the usage of cluster algorithms seems more appropriate: it is well known that the clusters contain similar data and the degree of association is weak between data of different clusters. Clustering algorithms (e.g., [8, 9, 11, 12, 13, 14]) are useful for the determination of buffer areas, called hotspots in crime analysis, car crash analysis, disease diffusion analysis, etc. For instance, the National Institute of Justice at Washington DC (USA) has developed a statistical tool, CrimeSTAT [9], for the GIS analysis of crime incident locations. We refer to [6] for an exhaustive list of clustering techniques which determine hotspots.

In order to determine the shape of each hotspot we have to use a density estimation method ([8, 13]), whereas the fuzzy c-means (FCM) algorithm [3] uses punctual cluster prototypes. However in many cases is not necessary to

* Corresponding author. Email: fdimarti@unina.it (F. Di Martino).

determine the exact shape of an hotspot; besides the FCM method has a linear computational complexity *O(N)* with respect to the number *N* of input vector data while the density estimation clustering methods have a computational complexity approximately equal to $O(N^2)$. Indeed the FCM algorithm has been used by many authors for determining areas with high concentrations of crimes (e.g., [9, 10, 15, 16, 17]).

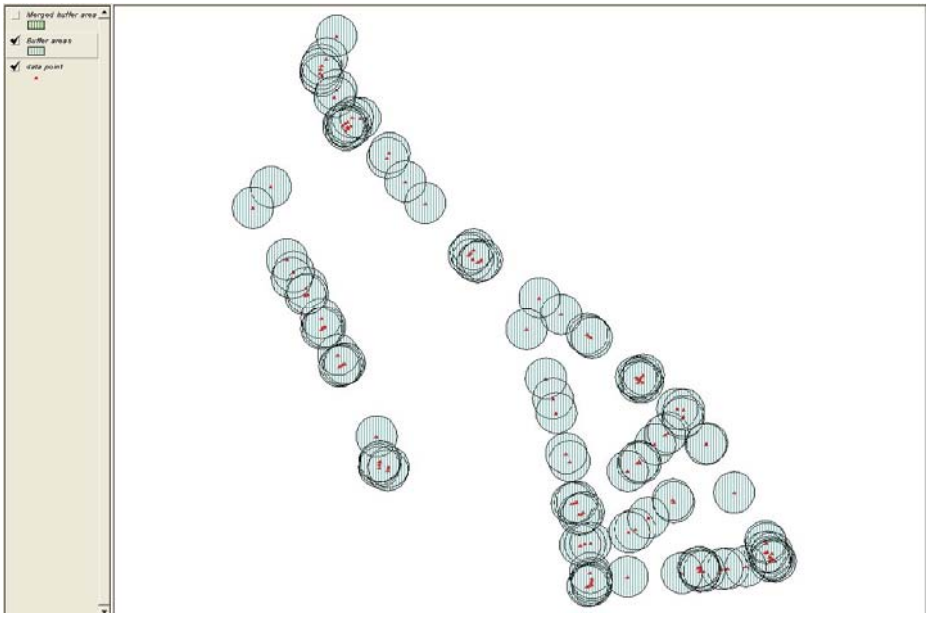

Figure 1: Example of separated circular buffer areas

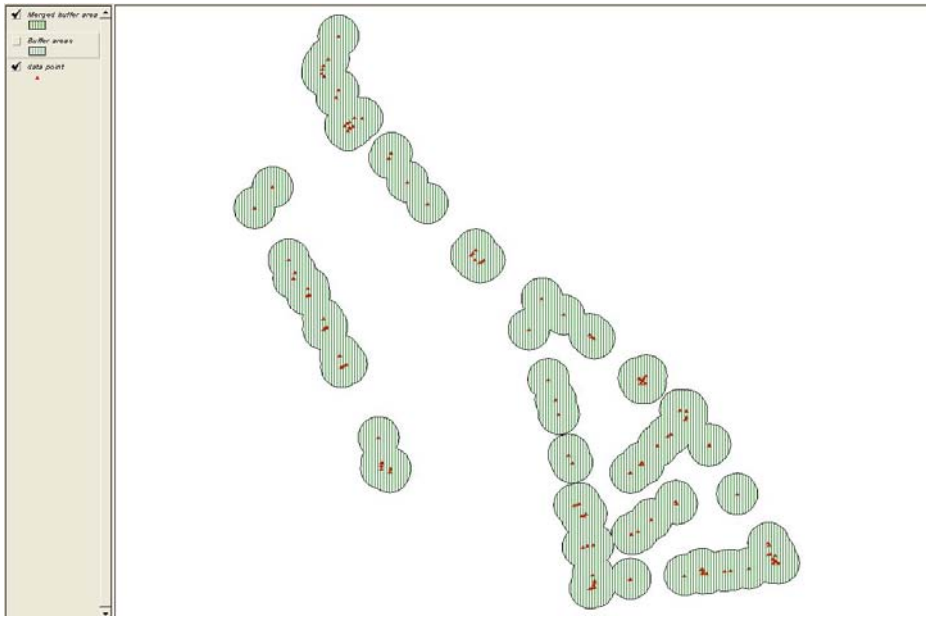

Figure 2: Example of merged circular buffer areas

The important parameters in a clustering algorithm are as follows: 1) the number of clusters is defined a priori in *K*-means [4] and FCM algorithms [3]; 2) the similarity (or the distance method) which reflects the nature of the dataset. In fact we adopt the distance that returns the best bidimensional geometrical shape of the clusters. In spatial analysis (e.g., [1, 2]) the Euclidean metric is used for small areas, such as in crime analysis [5] and incident analysis.

In the extended fuzzy c-means (EFCM), presented in [11, 12], the shape of the clusters are hyperspheres. The advantages in the usage of the EFCM algorithm are essentially two: 1) the determination of the number of the clusters

is precise; 2) the shape of every cluster volume prototype on the geographic map is circular because the Euclidean distance is used and it can be considered as an hotspot.

This paper is organized as follows: in Section 2, we describe the FCM and EFCM algorithms, respectively. In Section 3, we give an application of the EFCM algorithm in the specific problem of fire prevention of a forest area, located in New Mexico: indeed we construct buffer circular areas which represent dangerous areas of fire events. Section 5 gives our conclusions.

2 The FCM and EFCM Algorithms: An Overview

Let $X=\{x_1,...,x_N\}\subset R^n$ be the data set composed by N elements represented with the following matrix:

$$\mathbf{X}=\begin{pmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nN} \end{pmatrix}, \tag{1}$$

where $x_j = (x_{1j}, x_{2j},\dots,x_{nj})^T$ is the j-th feature vector for $j=1,\dots,N$. In the FCM algorithm the minimization of the following objective function is achieved:

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

where $m\geq 1$ is the fuzzifier parameter, u_{ij} is the membership degree of x_j to the i-th cluster, $i=1,\dots,C$. The set $V=\{v_1,\dots,v_C\}\subset R^n$ is the set of the centers of the C clusters, d_{ij} is the distance between the center $v_i=(v_{1i}, v_{2i},\dots,v_{nj})^T$ of the i-th cluster and the j-th feature vector x_j, calculated as

$$d_{ij}=\sqrt{(\mathbf{x_j}-\mathbf{v_i})^{\mathbf{T}} S(\mathbf{x_j}-\mathbf{v}_i)}, \tag{3}$$

where S is a positive and symmetric norm matrix. By definition, we have the following constraints:

$$\sum_{i=1}^{C} u_{ij}=1, \forall\ j\in\{1,...,N\}, \tag{4}$$

$$\sum_{j=1}^{N} u_{ij}<N, \forall\ i\in\{1,...,C\}. \tag{5}$$

It is not difficult to prove that the center of each volume prototype is obtained with the following:

$$v_i=\frac{\sum_{j=1}^{N} u_{ij}^{m} x_j}{\sum_{j=1}^{N} u_{ij}^{m}}, \tag{6}$$

for $i=1,\dots,C$ and the membership degrees of belongness are given by

$$u_{ij}=1\Big/\left(\sum_{k=1}^{C} d_{ij}^{2}/d_{kj}^{2}\right)^{2/m-1}. \tag{7}$$

Precisely speaking, initially the u_{ij} and the centers of the clusters are assigned randomly, moreover the u_{ij} are updated in each iteration. The iterative process stops when

$$\left\|\mathbf{U^{(s)}}-\mathbf{U^{(s-1)}}\right\|=\max_{i,j}\left|u_{ij}^{(s)}-u_{ij}^{(s-1)}\right|<\varepsilon, \tag{8}$$

where $\varepsilon>0$ is a prefixed parameter and $U^{(s)}=\left(u_{ij}^{(s)}\right)$ is the matrix $\boldsymbol{U}$ of the membership degrees calculated at the s-th step. The algorithm FCM presents two shortcomings: 1) a priori the number C of prototypes must be defined or one calculates C as minimum or maximum of a suitable function [6]; 2) the cluster centers tend to locate in areas with high concentrations of features and the zones with low density data points could be relevant. Generally speaking, the distribution of the data is sensitive to the initialization phase.

The EFCM algorithm was firstly proposed in [11, 12]. In general, the volume prototypes are hyperellissoids which become hyperspheres in case of Euclidean distance. If d_{ij} is the distance between the feacture vector x_j and the volume prototype V_i and if r_i is the radius of V_i, we say that x_j belongs to V_i if $d_{ij}\leq r_i$. The covariance matrix P_i associated to the i-th cluster V_i is calculated as [11]

$$\mathbf{P_i} = \sum_{j=1}^{N} u_{ij}^{m} (\mathbf{x_j} - \mathbf{v_i})(\mathbf{x_j} - \mathbf{v_i})^T \Big/ \sum_{j=1}^{N} u_{ij}^{m} \tag{9}$$

whose determinant gives the volume of the *i*-th cluster. P_i is symmetric and positive, hence it is decomposed in the form:

$$\mathbf{P_i} = \mathbf{Q_i} \mathbf{\Lambda_i} \mathbf{Q_i^T}, \tag{10}$$

where Q_i is an orthonormal matrix and $\Delta_i = (\lambda_{ij})$ is a diagonal matrix. Generally a value between the minimal and the maximum values of the diagonal matrix Λ_i can be used to estimate the radius r_i. Indeed we have that [11]:

$$r_i(s) = \frac{1}{n} \left\{ \sum_{k=1}^{n} \lambda_{ik}^{s} \right\}^{s}, \mathrm{s} \in \mathrm{R}. \tag{11}$$

The parameter *s* is inserted in order to control the error introduced from the aggregation of the dimensions of λ_{ij}. For $s \to -\infty$, formula (11) gives the radius of the largest hypersphere included in the hyperellipsoid that constitutes the cluster volume and for $s \to \infty$, formula (11) gives the radius of the smallest hypersphere including such volume.

For s→0, formula (11) becomes:

$$r_i = \frac{1}{n} \sqrt{\prod_{k=1}^{n} \lambda_{ik}^{1/n}} = \sqrt{\det(P_i)^{1/n}} \tag{12}$$

In EFCM algorithm, the objective function to be minimized is the following [11]:

$$J = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^{m} (d_{ij}^2 - r_i^2). \tag{13}$$

The update equation for u_{ij} is given by

$$\begin{aligned} u_{ij} &= \frac{1}{\sum_{k=1}^{C} \left(d_{ij}^2 \max\left(0, 1 - r_i^2 / d_{ij}^2\right) \right) / \left(d_{kj}^2 \max\left(0, 1 - r_k^2 / d_{kj}^2\right) \right)^{1/(m-1)}} \\ &= \frac{1}{\sum_{k=1}^{C} \left(d_{ij}^2 w_{ij} / d_{kj}^2 w_{kj} \right)^{1/(m-1)}}. \end{aligned} \tag{14}$$

The terms $d_{kj}^2 w_{kj} = \max\left(0, d_{kj}^2 - r_k^2\right) = d_{kj}^{'2}$ is seen as a squared distance of x_j from V_k. For each x_j, we consider the value φ_j equal to the number of clusters for which $d_{kj}^{'} = 0$ with $k=\{1,\ldots,C\}$; thus u_{ij} is given by [11]:

$$\begin{aligned} u_{ij} &= \frac{1}{\sum_{k=1}^{C} (d_{ij}^{'} / d_{kj}^{'})^{2/(m-1)}}, \varphi_j = 0 \\ u_{ij} &= \begin{cases} 0 & \text{if } d_{ij}^{'} > 0 \\ \dfrac{1}{\varphi_j} & \text{if } d_{ij}^{'} = 0 \end{cases}, \varphi_j > 0. \end{aligned} \tag{15}$$

The use of the formulas (15) produces the negative effect of diminishing the objective function when a meaningful number of features is placed in a cluster: this fact can prevent the separation of the clusters. In order to overcome this problem, the radius r_i starts with a small value and then gradually is increased with the factor $\beta^{(l)} / C^{(l)}$, where $C^{(l)}$ is the number of clusters at the *l*-th iteration and $\beta^{(l)}$ is a parameter defined recursively as

$$\beta^{(0)} = 1, \beta^{(l)} = \min(C^{(l-1)}, \beta^{(l-1)} + 1). \tag{16}$$

The determination of the number of clusters is achieved by adopting the following measure of inclusion of the *i*-th cluster in the *k*-th one:

$$I_{ik} = \sum_{j=1}^{N} \min(u_{ij}, u_{kj}) \Big/ \sum_{j=1}^{N} u_{ij}. \tag{17}$$

For symmetry, we use the following measure of cluster merging given by

$$S_{ik} = \max(I_{ik}, I_{ki}). \tag{18}$$

The merging between the two clusters *i* and *k* is done when at the *l*-th iteration we have that

$$\left|S_{\max}^{(l)} - S_{\max}^{(l-1)}\right| \le \varepsilon\,, \tag{19}$$

where $S_{\max}^{(l)} = S_{ik} \ge \alpha^{(l)}$ and the parameter $\alpha^{(l)} \in [0,1]$ is determined with the following adaptive formula [11]:

$$\alpha^{(l)} = \frac{1}{C^{(l-1)} - 1}. \tag{20}$$

The EFCM algorithm determines two indexes i^* and k^* such that $S_{i^*k^*} \ge \alpha^{(l)}$, then i^* and k^* are merged by setting

$$\begin{cases} u_{i^*j}^{(l)} = u_{i^*j}^{(l)} + u_{k^*j}^{(l)}, \forall j \in \{1,\ldots, N\} \\ C^{(l)} = C^{(l-1)} + 1, \end{cases} \tag{21}$$

and by removing the k^*-th row from the matrix U. Thus the EFCM algorithm can be summarized in the following procedure:

1) The user assigns the initial number of clusters $C^{(0)}$, $m>1$, $\varepsilon>0$, the initial value $S_{\max}^{(0)} = 1$ and $\beta^{(0)}=1$.

2) The membership degree $u_{ij}^{(0)}$ for $j \in \{1,\ldots, N\}$ and $I=\{1,\ldots,C^{(0)}\}$ are assigned randomly.

3) The centers of the clusters v_i are calculated by using formula (6).

4) The radii of the clusters are calculated by using formula (12).

5) The elements $u_{i,j}$ of the matrix U are calculated by using formula (15).

6) The elements $s_{i,k}$ of the similarity matrix S are calculated by using formula (18) and are determined i^* and k^* for which S_{i^*,k^*} has the maximum value.

7) If $\left|S_{i^*k^*}^{(l)} - S_{i^*k^*}^{(l-1)}\right| < \varepsilon$ and $S_{i^*k^*}^{(l)} > \alpha^{(l)} = 1/(M^{(l-1)}-1)$, then the i^*-th and the k^*-th clusters are merged via (21).

8) If $\|U^{(s)} - U^{(s-1)}\| < \varepsilon$ at the s-th iteration, then the process stops otherwise go to 3) for the (s+1)-th iteration.

3 Tests with Hotspot Fire Events

The authors of [6, 7]) have implemented the EFCM method in a GIS environment created with the tools ESRI/ARCGIS and ESRI/ARCVIEW.

In [6] it is proved that the EFCM algorithm finds the optimal number C of clusters during the iteration process while the use of a validity index in a pre-processing phase of the FCM algorithm does not supply always optimal values of C (e.g., [18, 19]).

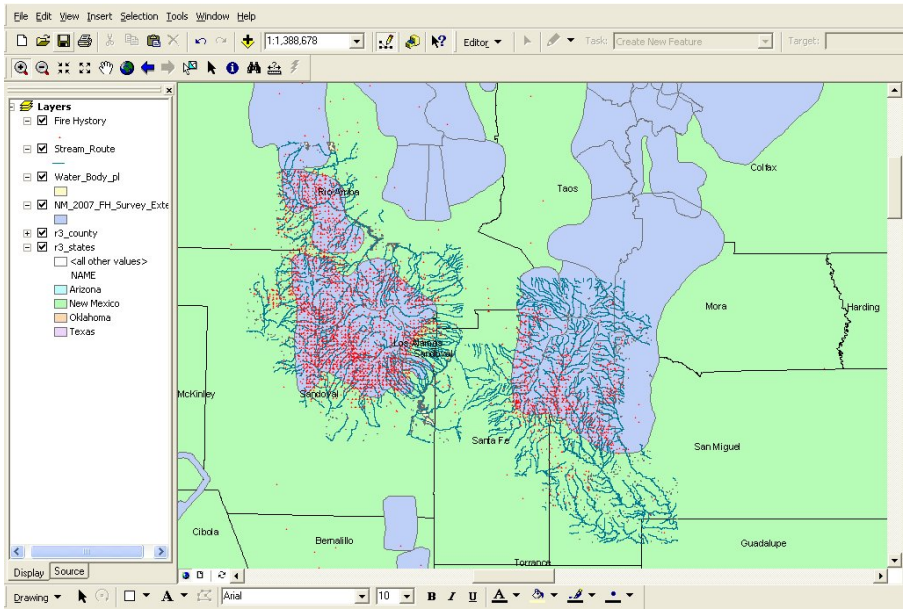

Figure 3: "Fire History" Santa Fe National Forest and related point data

As reported in Section 1, a circle on the geographic map can represent a prototype volume obtained with the usage of the EFCM algorithm and it can be considered as a good approximation of an hotspot. Indeed the exact shape of the clusters could be determined by using the density based clustering method as in crime analysis (cfr., e.g., [5, 16, 17]), but this method is more expensive than EFCM in terms of computational complexity. The patterns included in the *i*-th volume prototype have a membership value equal to 1 for the *i*-th cluster and 0 for the remaining clusters.

One of the main problems in the use of the FCM algorithm is the determination of the optimal number of clusters C. In [6] the authors compared several results (cfr., also [8, 18, 19]) and they pointed out that the EFCM algorithm finds the exact number of clusters during the iteration process.

Here we experiment the use of this algorithm on point-events considered as input data of the theme "Fire History" of USDA Santa Fe Service National Forest, downloaded from URL http://www.fs.fed.us/r3/gis/sfe_gis.shtml. The "Fire History" data (points) represent the locations at which the fires began in the last three years. The latitude and longitude of each point (location) were considered as *X*, *Y* coordinates. The data-test is made about 5000 features distributed in the geographic area of New Mexico, mainly covered by the counties of Los Alamos, Sandoval, San Miguel, Mora, Rio Arriba and Santa Fe. The "Fire History" data are shown on the map of Fig.3, where are displayed also the stream routes, the water bodies, the base vegetation location and the surveyed forest areas in New Mexico.

Our aim is to use the EFCM algorithm in the determination of circular areas with high fire frequencies considered as dangerous buffer areas.

In Fig.4 we show the "Fire History" data (points), the Santa Fe National Forest visitor map revised in 2005, the roads, the trial routes and the recreation opportunity areas. We are interested to analyze fire dangerous buffer areas: indeed we study, for instance, their impact with the recreation opportunity areas.

We have tested the use of the EFCM algorithm varying the initial number of clusters as shown in Table 1.

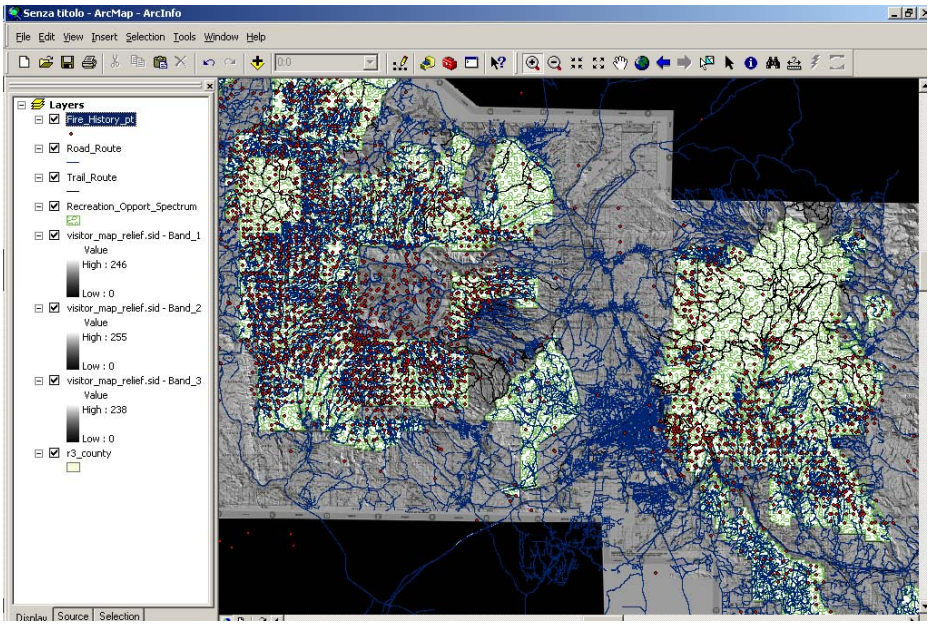

Figure 4: Santa FE Visitor Map, roads, trial routes and recreation opportunity areas

Table 1: Number of clusters and corresponding number of features

Initial number of clusters	Final number of clusters	Difference $\|\|U^{(s)} - U^{(s-1)}\|\|$
130	38	0.69×10^{-4}
120	38	0.71×10^{-4}
110	38	0.67×10^{-4}
100	38	0.85×10^{-4}
90	38	0.53×10^{-4}
80	38	0.49×10^{-4}
70	38	0.66×10^{-4}
60	38	0.88×10^{-4}
50	38	0.72×10^{-4}

The results in Table 1 prove that the EFCM algorithm is quite stable; indeed, the final number of clusters is always 38. The circular volume prototypes are geographically situated on the map given in Fig.5.

In Fig.6 we show the intersection of the recreation opportunity areas with the cluster prototypes used as circular buffer areas. So we can consider the intersection of an opportunity area with a cluster prototype as a recreation zone with high fire hazard.

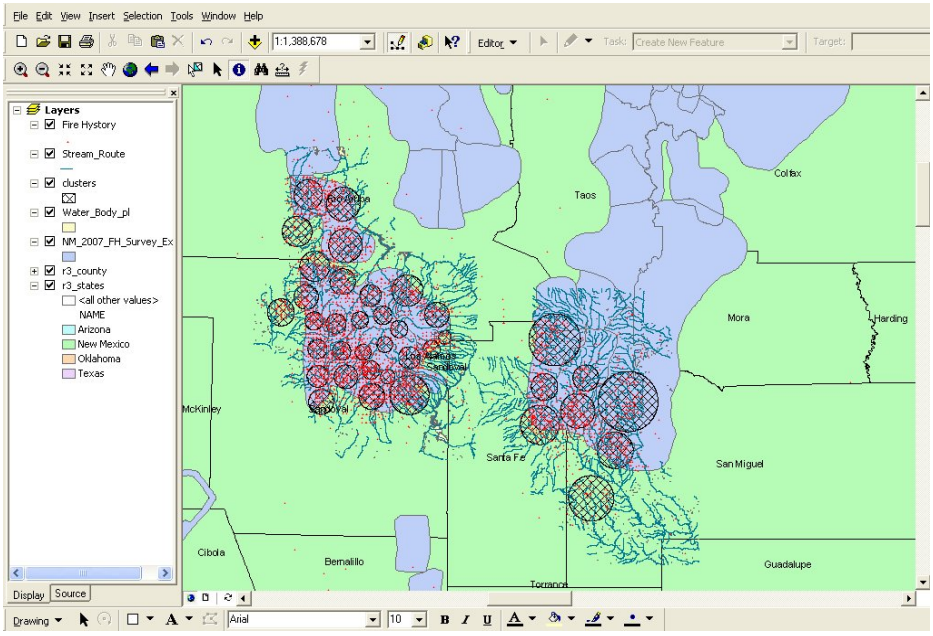

Figure 5: Cluster volume prototypes for the “Fire History” data (points)

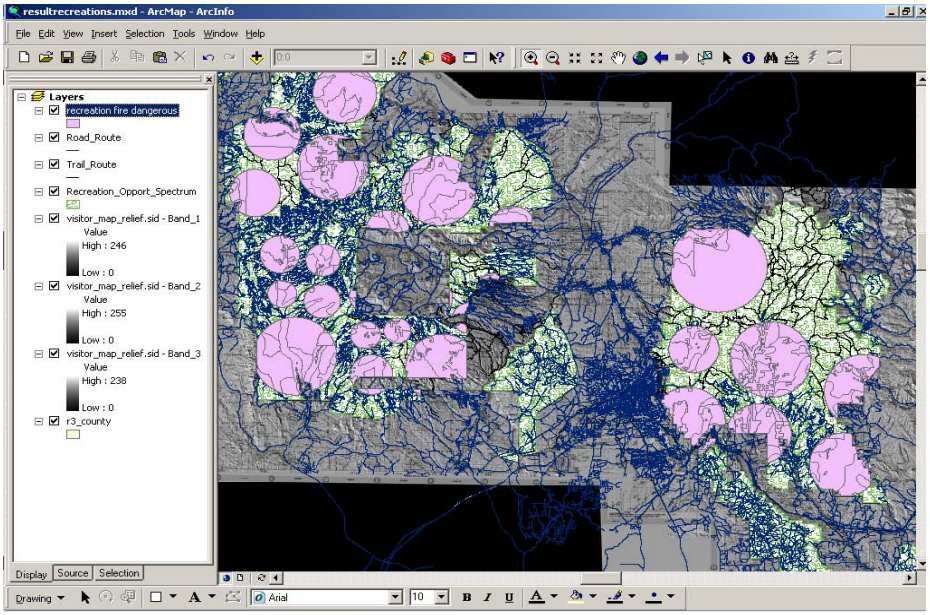

Figure 6: Fire dangerous recreation opportunity areas included in cluster prototypes

In Fig.7 we show the intersection of the base vegetation sites with the cluster prototypes and we obtain the areas (evidenced in red) considered as base vegetation zones with high fire hazard.

Sometimes the extension of the high fire hazard zones is augmented for safety reasons. Indeed we show the intersection of the base vegetation sites with circular buffer areas having radius of km. 2 more longer than the radius of the cluster prototypes in Fig.8.

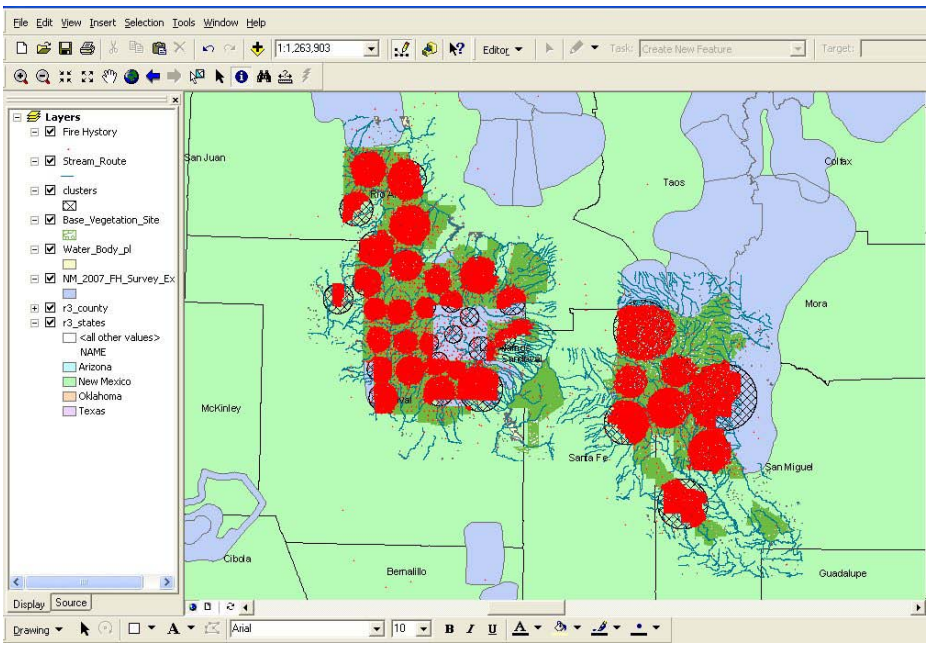

Figure 7: Base vegetation areas included in cluster prototypes

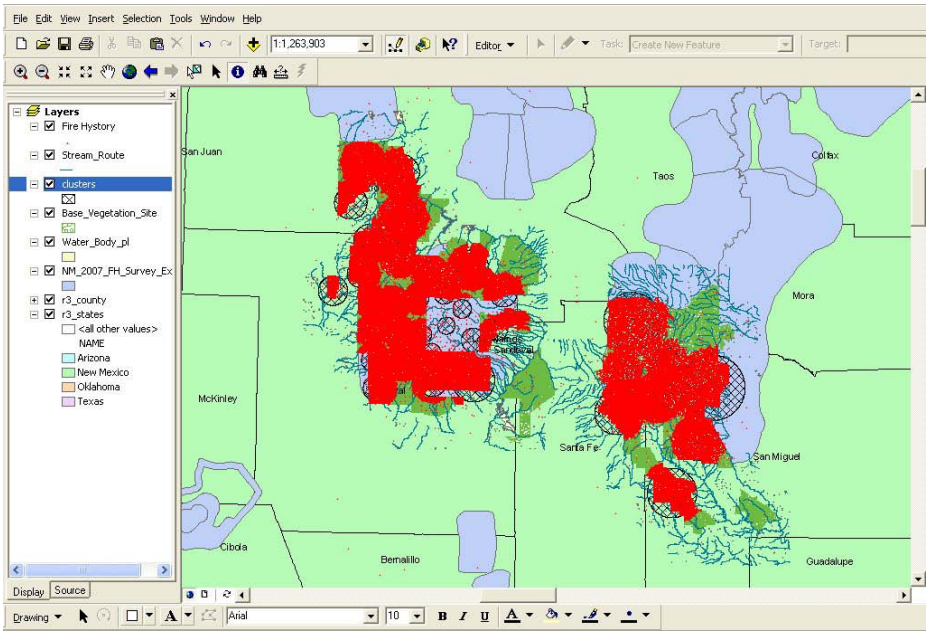

Figure 8: Base vegetation sites inside circular buffer areas having a radius of km. 2 more longer than the radius of the clusters

4 Conclusions

In spatial analysis usually impact areas are determined by using density clustering algorithms which have an elevated computational complexity. Here we propose the EFCM algorithm because it has the following advantages: robustness to noise and outliers, linear computational complexity and automatic determination of the optimal number of clusters. We derive dynamic buffer areas as hypersphere volume prototypes which become circles in the case of bidimensional pattern data, like in the case of point-events in a GIS. Indeed we have implemented the EFCM algorithm in a GIS

created with the usage of ESRI/ARCGIS and ESRI/ARCVIEW software tools and we have determined hotspots of fire events in the Santa Fè district (NM). All the data were downloaded from URL www.fs.fed.us/r3/gis/sfe_gis.shtml.

The above experiments have pointed out that we can use the EFCM algorithm in spatial analysis for the determination of circular buffer areas. These areas can be considered on the geographic map as a good approximation of classical hotspots. Applications to other frameworks like crime analysis, industrial pollution, etc. shall be tried in future works.

Acknowledgement

We thank the referees whose suggestions have greatly improved the presentation of this paper.

References

[1] Bailey, T.C., and A.C. Gatrell, *Interactive Spatial Data Analysis*, Longman Scientific and Technical, London, 1995.

[2] Besag, J., and J. Newell, The detection of clusters in rare diseases, *Journal of the Royal Statistic Society*, vol.154, pp.143–155, 1991.

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

[4] Burrough, P.A., J.P. Wilson, P.F.M. van Gaans, and A.J. Hansen, Fuzzy *k*–means classification of digital elevation models as an aid to forest geographic mapping in the greater yellowstone area (USA), *Landscape Ecology*, vol.16, pp.323–346, 2001.

[5] Chainey, S.P., S. Reid, and N. Stuart, When is a hotspot a hotspot? A procedure for creating statistically robust hotspot geographic maps of crime, in: Innovations in GIS 9: Socioeconomic Applications of Geographic Information Science, Taylor and Francis, London, 2002.

[6] Di Martino, F., V. Loia, and S. Sessa, Extended fuzzy c-means clustering algorithm for hotspot events in spatial analysis, *International Journal of Hybrid Intelligent Systems*, vol.4, pp.1–14, 2007.

[7] Di Martino, F., and S. Sessa, Dynamic buffer areas obtained by EFCM method in GIS environment, Lecture Notes in Computer Science, vol.5188, pp.92–95, 2008.

[8] Gath, I., and A.B. Geva, Unsupervised optimal fuzzy clustering, *IEEE Trans. Pattern Anal. Machine Intell.*, vol.11, pp.773–781, 1989.

[9] Grubesic, T.H., and A.T. Murray, Detecting hotspots using cluster analysis and GIS, *Annual Conference of CMRC*, 2001.

[10] Harries, K., *Geographic Mapping Crime: Principle and Practice*, National Institute of Justice, Washington DC, 1999.

[11] Kaymak, U., R. Babuska, *et al.*, Methods for simplification of fuzzy models, *Intelligent Hybrid Systems*, pp.91–108, 1997.

[12] Kaymak, U., and M. Setnes, Fuzzy clustering with volume prototype and adaptive cluster merging, *IEEE Transactions on Fuzzy Systems*, vol.10, no.6, pp.705–712, 2002.

[13] Krishnapuram, R., and J. Kim, Clustering algorithms based on volume criteria, *IEEE Transactions on Fuzzy Systems*, vol.8, no.2, pp.228–236, 2000.

[14] Lozano, J.A., P. Larranaga, and M. Grana, Partitional cluster analysis with genetic algorithms: Searching for the number of clusters, *Data Science, Classification and Related Methods*, pp.189–214, 1996.

[15] Lu, Y., and J.C. Thill, Assessing the cluster correspondence between paired point locations, *Geographical Analysis*, vol.35, no.4, pp.290–309, 2003.

[16] McGuire, P.G., and D. Williamson, Geographic mapping tools for management and accountability, *Third International Crime Geographic Mapping Research Center Conference*, 1999.

[17] Murray, A.T., I. McGuffog, J.S. Western, and P. Mullins, Exploratory spatial data analysis techniques for examining urban crime, *British Journal of Criminology*, vol.41, pp.309–329, 2001.

[18] Wu, K.L., and M.S. Yang, A fuzzy validity index for fuzzy clustering, *Pattern Recognition Lett.*, vol.26, pp.1275–1291, 2005.

[19] Xie, X.L., and I.G. Beni, A validity measure for fuzzy clustering, *IEEE Transactions Analysis Machine Intell.*, vol.13, pp.841–847, 1991.