**Abstract** — Data clustering algorithms are used in many fields like anonymization of databases, image processing, analysis of satellite images and medical data analysis. There are several C- Means clustering algorithms in the literature. Besides the hard C- Means, there are uncertainty based C-Means algorithms like the Fuzzy C-Means algorithm and its variants, the Rough C-Means algorithm, the Intuitionistic Fuzzy C- Means algorithm and the hybrid C-Means algorithms (Rough Fuzzy C-Means algorithm). In this paper we propose a new hybrid clustering algorithm called Rough Intuitionistic Fuzzy C-Means and evaluate its performance in comparison to the other algorithms mentioned above. We have applied these algorithms on numerical as well as image data of two different types and used some benchmarking indexes for the evaluation of their performance. The results show that the proposed algorithm outperforms the existing algorithms in almost all cases.

**Keywords** – outliers, outlier detection, fuzzy sets

1. **Introduction**

An outlier or an anomaly is an observation which deviates from the rest of observations significantly based on some measure. They are usually present due errors in measurements or different system conditions and thus, does not abide the with common properties of the system. With the increase in the amount of data, outlier detection has recently become an important data mining job. It is practically impossible to manually analyse this large amounts of data and detect outliers present in the data. Hence, a mechanism that can identify outliers present in the data is essential. Outlier detection finds usage in many applications like fraud detection in credit cards, network security, medicine and public health etc. For high dimensional data, locating the correct outliers is not an easy job as the traditional outlier detection methods are not efficient. Traditional outlier detection techniques are based on a full dimension space, and incapable of detection outliers hidden in partial dimensions because of dimensionality curse. Data is sparse and the actual outliers get masked by the noise effects of multiple dimensions in high dimensional space. However, the number of subspaces increases exponentially when the number of dimension increases, and exhausting all subspaces in high-dimensional data is impossible.

1. **Related Work**
2. **Algorithm**

The following section introduces the fuzzy set and fuzzy similarity scale.

**3.1 Fuzzy Set and Similarity Scale**

The notion of fuzzy set was introduced by Zadeh [1] as an extension of the notion of crisp sets in order to model uncertain data. In a crisp set, an element is either a member of the set or not. Fuzzy sets, on the other hand, allow elements to be *partially* in a set. Each element is given a degree of membership in a set. This membership value can range from 0 (not an element of the set) to 1 (a member of the set). A membership function is the relationship between the values of an element and its degree of membership in a set. Formally, fuzzy sets can be defined as follows:

**Definition of Fuzzy Set**

Let U be a universe of discourse. F is a fuzzy subset of U if there is a membership function nu(of f): U 🡪 [0, 1], which associates with each element u belonging U a membership value nu(of f)(u) in the interval [0, 1]. The membership value nu(of f)(u) for each u belonging U represents the grade of membership of the element u in the fuzzy set F.

Zadeh [2] proposed the following notation for a fuzzy set F:

F(U) = {(u, nu(of f)(x=u)) |u belongs to U}.

Here, A(x) is the membership function, which ranges between 0 and 1. A value of 1 indicates full membership, between 0 and 1 indicates partial membership and value of 0 indicates that x is not a member of fuzzy set F(X).

The fuzzy logic theory provides a mathematical method to apprehend the uncertainties related to the human cognitive process, for example, thinking and reasoning and it can also handle the issue of uncertainty and lexical imprecision.

Given A ∈ F(X), we need to know which class A should be identified with. To solve this problem, we need to measure how close two fuzzy sets are. This can be solved using nearness measure. Formally, nearness measure of two fuzzy sets can be defined as follows:

**Definition of Nearness measure**

If N: F (X)×F (X) → [0, 1] satisfies that

(1) N(∅, X) = 0 and N(A,A) = 1 whenever A ∈ F(X),

(2) N(A, B) = N(B,A) whenever A,B ∈ F(X),

(3) N(A, C) ≤ min(N(A,B),N(B,C)) whenever A ⊆ B ⊆ C,

then N is called a nearness measure.

1. **Fuzzy constraint based on nearness measure**

There are many types of nearness measures for numerical data such as Euclidean distance-related, Hamming distance-related, lattice-based methods and Minkowski distance-related. Equation x,y and z gives the formula for Minkowski, hamming and Euclidean distance measures respectively.

Equation X: ( (x 1 - y 1) p + (x 2 - y 2) p) 1/p

Equation Y: ( (x 1 - y 1) 2 + (x 2 - y 2) 2) 1/2

Equation Z: (x 1 - y 1)  + (x 2 - y 2) 

It can be seen that hamming and Euclidean distance measures are special forms of Minkowski distance when p=2 and p=1 respectively. These distance measures are independent of the underlying data distribution. In cases where the values along the x-dimension is much larger than the y-dimension, normalization such as z-transform or min-max normalization of each data object is performed. Formally, z-transform and min-max normalization is defined as follows:

**Definition of Z-transform**

Given an n-dimensional data object with numerical values (x1,x2,x3,…xn), such that (nu1, nu2, nu3,…,nun) and (sig1,sig2,sig3, … , sign) are the mean and the standard deviation of each of the n dimension then the data object is transformed as:

Equation X: ( (x 1 – μ x)/σ 1, (x 2 – μ 2)/σ 2 , (x 3 – μ 3)/σ 3 ,… (x n – μ n)/σ n ,)

**Definition of Min-Max Normalization**

Given an n-dimensional data object with numerical values (x1,x2,x3,…xn) such that (min1.min2,min3, …, minn) and (max1, max2, max3, … maxn) represent the minimum and maximum value of each of the n dimensions then the data object is transformed as:

Equation Y: ( (x 1 – min 1)/max1, (x 2 – min  2)/ max1  2 , (x 3 – min  3)/ max1  3 ,… (x n – min  n)/ max1  n ,)

**NEED TO WITE ABOUT LATTICE BASED NEAR NESS MEASURE**

In order to match the fuzzy constraints, the lattice based nearness measure is redefined as per [2] [3].

**Definition for lattice based measures:** Given a dataset DS consisting of n attributes A={A1,A2,A3,…AN}. Let D={d1,d2,d3..dk} be the set of k data objects in DS where di = {di1,di2,di3…din}. Therefore, dij represent the value of the jth attribute of the ith object. Let M=(M1,M2,M3…,Mn) represent the priori information given by the users, where Mi is the priori value on attribute i.

Let G(X) be a fuzzy set, where X is a subset of the attributes and Di,M ∈ F(X),

**Eqation X: di inner Mi =** ∨**x∈X (di(x)∧M(x))**

is the inner product of di and M.

**Equation y**: di outer M=**∧**􏰃x∈X (di(x)∨ M(x))

is the outer product of di and M.

The lattice based nearness measures ZL can be defined using x and y as follows:

ZL(Oi,U) = (di inner M)∧(1−di outer M) for all di, M ∈ F(X)

*Let us now describe how lattice based nearness can be used to prune the outliers dataset from the dataset.*

Given an object di, priori knowledge M, and threshold value σ, if ZL(di,M) ≥ σ, then object di is called a required object, which matches the constraint condition given by the user. This means the data object is of user’s interest. If ZL (di , M ) < σ, then object di is called a not required object, which does not match the constraint condition given by the user. This means the data object is not of user’s interest. In this algorithm, threshold value σ also known as nearness-threshold, is provided by users.

Thus, we calculate the nearness measure between each object in dataset DS and priori knowledge M. This prunes the dataset DS removing data objects of disinterest from DS. *This reduced dataset helps in improving the efficiency of outlier detection when further steps are applied on it.*

**4.1 Example of the pruning step**

Reference:

[1] L. Zadeh, "Fuzzy sets", Information and Control, vol. 8, no. 3, pp. 338-353, 1965. Available: 10.1016/s0019-9958(65)90241-x [Accessed 20 February 2019].

[2] D. Wu and J. M. Mendel, “A vector similarity measure for linguistic approximation: Interval type-2 and type-1 fuzzy sets,” Information Sciences, vol. 178, no. 2, pp. 381–402, 2008.

[3] D. Wu and J. M. Mendel, “A comparative study of ranking methods, similarity measures and uncertainty measures for interval type-2 fuzzy sets,” Information Sciences, vol. 179, no. 8, pp. 1169–1192, 2009.

[4]