International Journal of Database Theory and Application
Vol.6, No.6 (2013), pp.1-18
http://dx.doi.org/10.14257/ijdta.2013.6.6.01

Implementation of the Fuzzy C-Means Clustering Algorithm in Meteorological Data

Yinghua Lu[1], Tinghuai Ma[1,*], Changhong Yin[2], Xiaoyu Xie[2], Wei Tian[1] and ShuiMing Zhong[1]

[1]*School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing 210044;*
[2]*Wuhan Meteorological Service, Wuhan 430040*

cloveryunyan@163.com

Abstract

An improved fuzzy c-means algorithm is put forward and applied to deal with meteorological data on top of the traditional fuzzy c-means algorithm. The proposed algorithm improves the classical fuzzy c-means algorithm (FCM) by adopting a novel strategy for selecting the initial cluster centers, to solve the problem that the traditional fuzzy c-means (FCM) clustering algorithm has difficulty in selecting the initial cluster centers. Furthermore, this paper introduces the features and the mining process of the open source data mining platform WEKA, while it doesn't implement the FCM algorithm. Considering this shortcoming of WEKA, we successfully implement the FCM algorithm and the advanced FCM algorithm taking advantage of the basic classes in WEKA. Finally, the experimental clustering results of meteorological data are given, which can exactly prove that our proposed algorithm will generate better clustering results than those of the K-Means algorithm and the traditional FCM algorithm.

Keywords: *Fuzzy C-Means algorithm; WEKA; Meteorological Data*

1. Introduction

Clustering analysis plays an important role in the data mining field, it is a method of clustering objects or patterns into several groups. It attempts to organize unlabeled input objects into clusters or "natural groups" such that data points within a cluster are more similar to each other than those belonging to different clusters, *i.e.*, to maximize the intra-cluster similarity while minimizing the inter-cluster similarity. In the field of clustering analysis, a number of methods have been put forward and many successful applications have been reported.

Clustering algorithms can be loosely categorized into the following categories: hierarchical, partition-based, density-based, grid-based and model-based clustering algorithms [1-4]. Among them, partition-based algorithms which partition objects with some membership matrices are most widely studied. Traditional partition-based clustering methods usually are deterministic clustering methods which usually obtain the specific group which objects belong to, *i.e.*, membership functions of these methods take on a value of 0 or 1. We can accurately know which group that the observation object pertains to. This characteristic brings about these clustering methods' common drawback, that we can not clearly know the

[*] Corresponding Author, email: thma@nuist.edu.cn

ISSN: 2005-4270 IJDTA

probability of the observation object being a part of different groups, which reduces the effectiveness of hard clustering methods in many real situations. For this purpose, fuzzy clustering methods which incorporate fuzzy set theory [5] have emerged. Fuzzy clustering methods [6-8] quantitatively determine the affinities of different objects with mathematical methods, described by a member function, to divide types objectively.

Among the fuzzy clustering method, the fuzzy c-means (FCM) algorithm [9] is the most well-known method because it has the advantage of robustness for ambiguity and maintains much more information than any hard clustering methods. The algorithm is an extension of the classical and the crisp k-means clustering method in fuzzy set domain. It is widely studied and applied in pattern recognition, image segmentation and image clustering [10-12], data mining [13], wireless sensor network [14] and so on.

WEKA (Waikato Environment for Knowledge Analysis) based on JAVA environment is a free, non-commercial and open-source platform aiming at machine learning and data mining. In WEKA, it implements several famous data mining algorithms. Users can call the appropriate algorithms according to their various purposes. However, the FCM algorithm is not integrated into WEKA.

In this paper, we implement the FCM algorithm and successfully integrate it into WEKA to expand the system functions of the open-source platform, so that users can directly call the FCM algorithm to do fuzzy clustering analysis. Besides, considering the shortcoming of the classical FCM algorithm in selecting the initial cluster centers, we represent an improved FCM algorithm which adopts a new strategy to optimize the selection of original cluster centers.

The structure of this paper is as follows. In the next section, we start a brief review of WEKA and the FCM algorithm. Section 3 describes the main ideas of the traditional FCM algorithm. In Section 4, we present our proposed algorithm based on the traditional FCM algorithm. Experiments results on meteorological data will be shown in Section 5. Finally, conclusions and future work are summarized.

2. Related Work

2.1. WEKA

The full name of WEKA is Waikato Environment for Knowledge Analysis and WEKA is also the name of a kind of birds which come from New Zealand. The package of WEKA can be downloaded from the website of Waikato University in New Zealand (http://www.cs.waikato.ac.nz/ml/weka/), the latest version number is 3.7.2.

2.1.1. The Data Mining Process in WEKA:

(1) Data input

According to different users' demands, WEKA provides three types of methods of inputting data sets: files, the web sites and databases. Note that the format of files is limited when we input data from existing files. There are three main formats of files: Arff data files (*.arff), C4.5 data files (*.names or *.data), CSV data files (*.csv). When we do clustering analysis or association analysis, Arff data files are the best choice. Meanwhile, the recognized attribute types are also restricted including numeric type, nominal type, string type, date type and relational type.

(2) Data preprocess

Data preprocess is also called data filter. Objects doing data preprocessing can be divided into two categories, attributes and instances. There are also two methods for data preprocessing, supervised methods and unsupervised methods.

Among so many methods of preprocessing different types of data, what we frequently use in this stage is: missing values processing, standardization processing, normalization processing and discrete processing for attributes.

1) missing values processing: the corresponding class is weak.filters.unsupervised.attribute.ReplaceMissingValues. For numeric attributes, WEKA replaces the missing values with the average value. For nominal attributes, WEKA replaces the missing values with the mode which occurs most frequently.

2) standardization processing: the corresponding class is weka.filters.unsupervised.attribute.Standardize. It is just applicable to numeric attributes. After the standardization processing, all values of numeric attributes form a normal distribution.

3) normalization processing: the corresponding class is weka.filters.unsupervised.attribute.Normalize. It is also limited to numeric attributes. We can normalize numeric values into any interval taking advantage of zoom and pan parameters. In default the resulting values are restricted in the interval [0, 1].

4) discrete processing: the corresponding class is weka.filters.unsupervised.attribute.Discretize and weka.filters.supervised.attribute.Discretize. These two categories respectively discretize numeric attributes in supervised and unsupervised ways.

(3) Data mining

The data mining process consists of classification, clustering analysis, association rules and other pattern analysis. WEKA almost implements all frequently used algorithms for mining different patterns. Here we illustrate the process of classification and clustering analysis.

The process of classification in WEKA is as follows:

1) Input training and test samples;

2) Initialize the classifiers;

3) Use training samples to train classifiers;

4) Test the performance of classifiers making use of test samples;

5) Output the classification results.

The process of clustering analysis in WEKA is as follows:

1) Read samples which need to predict;

2) Initialize the clustering algorithms and set parameters;

3) Cluster the samples using algorithms;

4) Output the clustering results.

(4) Visualization

Generally simple data results may not satisfy users' demands, sometimes we need to vividly describe the changing trend of data results. Hence visualization appears.

Visualization makes the data mining process and results of data mining visualize. It is a useful tool for the mining process and improves the efficiency of the mining process.

2.1.2. The Comparison between WEKA and Other Data Mining Platforms: Nowadays data mining is still an emerging field, and is closely associated with other fields like statistics, machine learning and artificial intelligence. Recently, more and more data mining platforms are appearing. Here we describe some frequently used data mining platforms and discuss their advantages and disadvantages.

(1) Intelligent Miner

Intelligent Miner developed by IBM is the data mining software, consisting of Intelligent Miner for Data and Intelligent Miner for Text, that can extract useful information from both databases and texts.

Intelligent Miner for Data can extract implicit information from databases, data warehouses and data centre, and can find patterns from traditional databases or ordinary files taking advantage of structural data. It is widely used in market analysis, fraud monitoring and customer contact management.

Intelligent Miner for Text allows enterprises to execute the data mining process from texts, here texts can be text files, web pages, e-mails and so on.

Intelligent Miner has the capability of linguistic analysis and the ability of aggregation and filtering. It can deal with the huge amount of data and support parallel processing. Nevertheless, its GUI may not be friendly for users, and users need to be familiar with UNIX.

(2) Enterprise Miner

Enterprise Miner from SAS is a common data mining tool. The data mining process in Enterprise Miner is usually in the following order: sampling-exploration-conversion-modeling-assessment. It can be integrated with SAS data warehouse and OLAP, and can implement the end to end knowledge discovery process of inputting data, analyzing data and obtaining results.

Compared with Intelligent Miner, Enterprise Miner provides graphical interfaces and visual operations for beginners. However, it desires more space requirements to store temporary files and is difficult to output the decision tree.

(3) SPSS Clementine

As an open data mining tool, SPSS Clementine not only supports the overall data mining process, from data acquisition, transformation, modeling, evaluation to final deployment, but it also supports the industry standard (CRISP-DM) in the field of data mining. Generally speaking, SPSS Clementine pays more attention to solving the problem itself instead of doing some technical work.

(4) Darwin

Darwin developed by Oracle Corporation supports several data mining algorithms, like neural networks, classification, regression, K-nearest neighbor algorithm (KNN), genetic algorithms and so on.

Darwin has the following three important advantages. Firstly, the parallel processing of data mining algorithms accelerates Darwin while dealing with huge amounts of data. Secondly, its simplicity of extracting patterns is beneficial for incorporating itself with other applications. Finally, GUI with the windows style is friendly for clients to use.

(5)WEKA

WEKA is the most well known open-source machine learning and data mining software. WEKA provides a friendly interactive interface for ordinary users who need simple data analysis, while for the researchers who research the theory of data mining, WEKA provides open-source methods for learning and implementation. For instance, when researchers want to compare the performance between the existing algorithms and their own algorithms, what they need to do is to achieve their own algorithms without considering the realization of all comparison algorithms that have been implemented in WEKA [15].

Nevertheless, comparing to its high performance in machine learning, WEKA is weak in statistical analysis. The most important drawbacks of WEKA are the inability to perform multi-relational process and the lack of a merge tool for interconnected tables [16].

2.2. The Fuzzy c-means Algorithm

The fuzzy c-means algorithm was introduced by Ruspini [17] and later extended by Dunn [18] and Bezdek [9, 19, 20] and has been widely used in cluster analysis, pattern recognition and image processing *etc*. The fuzzy c-means clustering algorithm (FCM) introduces the fuzziness for the belongingness of each object and can retain more information of the data set than the hard k-means clustering algorithm (HCM). Although the FCM algorithm has considerable advantages compared to the k-means clustering algorithm, there are also some shortcomings when using the FCM algorithm in practice.

The main limitation of the FCM algorithm is its sensitivity to noises. The FCM algorithm implements the clustering task for a data set by minimizing an objective-function subject to the probabilistic constraint that the summation of all the membership degrees of every data point to all clusters must be one. This constraint results in the problem of this membership assignment, that noises are treated the same as points which are close to the cluster centers. However, in reality, these points should be assigned very low or even zero membership in either cluster. In order to further enhance its robustness to noise and outliers, many researches have been conducted. The possibilistic clustering algorithm (PCA) was put forward by Krishnapuram and Keller [7, 21] to improve this drawback of FCM. PCA relaxes the column sum constraint so that the sum of each column satisfies the looser constraint. However, PCA is heavily dependent on parameters used and may obtains coincident clusters. So several variants of PCA have been brought about to solve the above two restricts of PCA [22, 25]. Nevertheless, although PCA can achieve an overall higher accuracy, FCM is proved to be more consistent and stable experimental results. Another method of solving the FCM's sensitivity to noises is incorporating the FCM algorithm with kernel methods, which has been proved to be robust to outliers or noises of the dataset [26]. M. Gong *et al.*, introduced an improved fuzzy c-means algorithm by applying a kernel distance measure to the objective function [10]. The main idea of kernel methods is to transform complex nonlinear problems in original low-dimensional feature space to the easily solved problems in the high-dimensional transformed space. FLICM [11] was proposed by S. Krinidis and V. Chatzis taking advantage of a fuzzy local similarity measure, which achieves the goal of ensuring noise insensitiveness.

Another shortcoming of the FCM algorithm is the difficulty in selecting appropriate parameters. One of the important parameters is the fuzziness index m which influences the performance of the FCM algorithm when clusters in the data set have different densities. When $m=1$, the FCM algorithm degenerates into the HCM algorithm. A good choice of m should take the data distribution of the given data set into accout [10]. L. Zhu *et al.*, [27] presented a generalized algorithm called GIFP-FCM, which allows the fuzziness index m not to be fixed at the usual value $m=2$ and improves the robustness and convergence. The method that GIFP-FCM utilizes is conducting a new objective function making use of a novel membership constraint function. The other way to deal with the parameter m is realizing the management of uncertainty on the basis of the fuzziness index. I. Ozkan and I. Turksen [28] introduced a approach that evaluate m according to entropies after removing uncertainties from all other parameters. C. Hwang *et al.*, [29] incorporated the interval type-2 fuzzy set into the FCM algorithm to manage the uncertainty for fuzziness index m.

The last drawback we have to indicate is that the FCM algorithm is easy to get struck in the local minima, while what we want to find is the global extrema. To increase the probability of finding global extrema, various alternative methods for the optimization of clustering algorithm were suggested in many literatures. The most commonly adopted way to ameliorate this problem is integrating genetic algorithms into the FCM algorithm. Ant colony optimization (ACO) [30] has been successfully applied to fuzzy clustering [31]. Runkler introduced an ACO algorithm that explicitly minimizes the HCM and FCM cluster models [32]. Also particle swarm optimization (PSO) [33] has been applied to clustering [34, 35]. ACO and PSO both belong to swarm intelligent algorithms [36]. Swarm intelligent algorithms have the advantages of convenient implementation, parallel capability and ability to avoid local minima.

In addition to the disadvantages mentioned above, the FCM algorithm still has other restricts, for example, influenced by equal partition trend of the data set and sensitivity to initial conditions like the cluster number and the cluster centers. These issues have also been studied in many literatures [37, 38].

Based on the above discussions about limitations of the FCM algorithm, this paper brings about an improved FCM algorithm which realizes the optimization of selecting the initial cluster centers.

3. FCM in WEKA

In this section, we first describe the theory about the FCM algorithm in detail. Secondly, we introduce the basic core classes used in the FCM algorithm at the platform WEKA. Finally, the implement classes of the FCM algorithm are shown.

3.1. Theory on the FCM Algorithm

Contrary to traditional clustering analysis methods, which distribute each object to a unique group, fuzzy clustering algorithms gain the membership values between 0 and 1 that indicate the degree of membership for each objects to each group. Obviously, the sum of the membership values for each object to all the groups is definitely equal to 1. Different membership values show the probability of each object to different groups.

The FCM algorithm is one of the most popular clustering methods based on minimization of a generalized least-squared errors function. Given a data set $X=\{x_1,x_2,...,x_N\}\subseteq R^{N\times q}$, n is the number of samples, q is the dimension of the sample $x_j(j=1,2,...,N)$. The FCM algorithm is based on minimizing the criterion with respect to the membership value u_{ij} and the distance d_{ij}.

$$J_m = \sum_{j=1}^{N}\sum_{i=1}^{C}(u_{ij})^m d_{ij}{}^2(x_j, \omega_i) \qquad (1)$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}(\frac{\| x_i - \omega_j \|}{\| x_i - \omega_k \|})^{\frac{2}{m-1}}} \qquad (2)$$

$$\omega_j = \frac{\sum_{i=1}^{N} u_{ij}{}^m x_i}{\sum_{i=1}^{N} u_{ij}{}^m} \qquad (3)$$

Subject to

$$\begin{cases} \sum_{i=1}^{C} u_{ij} = 1,\ j = 1, 2, ..., N \\ 1 \geq u_{ij} \geq 0,\ i = 1, 2, ..., C,\ j = 1, 2, ..., N \\ n > \sum_{j=1}^{N} u_{ij} > 0,\ i = 1, 2, ..., C \end{cases} \qquad (4)$$

Here N is the number of objects and C is the number of clusters, Where u_{ij} is the degree of membership that the object x_j pertains to the cluster center ω_i , $U = \{u_{ij},\ i = 1, 2, ..., C,\ j = 1, 2, ..., N\}$ which is the membership matrix has to satisfy the constraints in (2). $V = \{v_i,\ i = 1, 2, ..., C\}$ is the cluster prototype matrix and v_i is the prototype of the center of cluster i . $m \in [1, \infty)$ is the fuzzy factor. According to many studies, $m \in [2, 2.5]$ is practical [39].

The FCM algorithm can be summarized by the following steps:

Step1: Initialize matrix $U = [u_{ij}]$ with the initial value $U^{(0)}$;

Step2: At k-step: calculate the cluster prototype matrix $V^{(k)} = [v_i]$ with $U^{(k)}$;

Step3: Update $U^{(k)}$, $U^{(k+1)}$;

Step4: if $\| U^{(k+1)} - U^{(k)} \| < \varepsilon$ then stop, or to step2.

To sum up, the basic idea of the FCM algorithm is that use iterative method for solving equation (2) and (3), until a termination condition is met. Here, ε is the threshold of the termination condition.

3.2. Basic Core WEKA Classes used in Clustering

(1) ArffLoader

The ArffLoader class is used to read the standard file type ARFF (Attribute-Relation File Format) used in algorithms.

In WEKA, limited file types are allowed to be handled with. When we do clustering analysis, the standard file type is the ARFF (Attribute-Relation File Format) file. Before we deal with files storing data sets, we must transfer them into the standard type. The conversion process is complicated before WEKA v6.0 appears. WEKA whose version is higher than 6.0 provides the tool used to transfer file types into ARFF. The dataset representation in ARFF is shown below with some numeric data:

```
@relation name

@attribute a1 numeric

@attribute a2 numeric

…

@attribute an numeric

@data
```

As is shown above, the ARFF file consists of two distinct sections, the header section and the data section. The header section contains the name of the relation, while the data section includes the name and the type of attributes.

(2) Instances

The Instances class shows the way to store all the records in data sets. It is responsible for inputting all the records from data source files which are always the ARFF files. In this class, we should pay more attention to the following functions.

Instances(data): the constructor function which return instances from the given data;

numAttributes(): return the number of attributes in the data set;

numInstances(); return the number of instances in the data set;

instance(i): return the *ith* instance of the instances;

(3) Instance

The instance class store just one record of the data set, it just can be seen as the format of a single record in the algorithm.

(4) DecisionTableHashKey

The function of this class is exploiting decision tables to randomly generate the cluster prototype matrix. The FCM algorithm should initialize the cluster prototype matrix or the membership matrix after setting the parameters specified. In this paper, initializing the cluster prototype matrix is adopted, and then computing the membership matrix according the formula (2). So this class is applied to confirm the original cluster prototype matrix.

(5) Matrix

According to the theory of the FCM algorithm, matrix is the most manifestation of the data in FCM. For instance, cluster centers are stored in the format of the cluster prototype matrix; the possibilities of each object belongs to different clusters are also in the form of the membership matrix.

3.3. The Implement Classes of the FCM Algorithm

In traditional WEKA, the FCM algorithm is not integrated in WEKA. Now, we implement the FCM algorithm making use of the existing classes in WEKA. The class implementing the

FCM algorithm in WEKA is called FCMWeka. In this class, several interesting functions are used:

(1) setNumClusters() and getNumClusters()

This function is used to set the number of cluster centers. If you don't call this function, the FCM algorithm also set the default value of the number of cluster centers which is 2.

(2) setFuzzifier() and getFuzzifier()

In the FCM algorithm, a significant parameter is the fuzzy factor which is 2 in default. The value of the fuzzy factor plays an important role in the performance of the FCM algorithm. Although repeated studies have shown that 2 is the most appropriate value of the fuzzy factor, the fuzzy factor is closely related to the different experimental data. So we still give users more choices to choose the most favorable value according to their own experimental data. For this intent, the function setFuzzifier() appears.

(3) getClusterCentriods()

The getClusterCentriods() function returns the specific value of the cluster centroids.

(4) getClusterSizes()

The getClusterSizes() function returns the number of objects in each cluster.

(5) buildClusterer()

The buildCluster() function is the most important function to generate clusters. Updating the membership matrix and the cluster matrix and computing the objective function to close to the threshold are all implemented in this function.

(6) ClusterProcessedInstance()

This function can tell which cluster every instance belongs to and return the corresponding cluster number.

4. The Improved FCM Algorithm for Meteorological Data

According to the discussion about the traditional FCM algorithm in Section 2, the initial condition of cluster centers influences the performance of the algorithm. The best choice of the original cluster centers needs to consider the features of the data set. In this paper, meteorological data is chosen as our experimental data. Meteorological data is different from other experiment data. If we just use the traditional FCM algorithm to deal with the meteorological data, there will be a large error when clustering a certain object. To solve the initialization problem, we put forward an improved FCM algorithm in term of selecting the initial cluster centers.

Nowadays, there are several methods to select the original cluster centers. In the following section we will go through some commonly used methods.

(1) Randomly

The traditional FCM algorithm determines initial cluster centers randomly. This method is simple and generally applicable to all data but usually causes local minima.

(2) user-specified

Normally, users decide original cluster centers by some priori knowledge. According to the understanding of the data, users always can obtain logical cluster centers to achieve the purpose of the global optimum.

(3) Randomly classify objects into several clusters, compute the center of each cluster and determine them as cluster centers

More time consumption is spent when randomly classifying objects in this method. When the number of objects in data sets is very small, the cost of time can be ignored. Nevertheless, as the number of objects increases, the speed of the increasing cost of time can be largely rapid.

(4) Select the farthest points as cluster centers

Generally speaking, this method selects initial cluster centers following to the maximum distance principle. It can achieve high efficiency if there are no outliers or noisy points in data sets. But if the data sets contain some outliers, outliers are easier to be chosen as the cluster centers.

(5) Select points with the maximum density

The number of objects whose distance is less than the given radius r from the observed object is defined as the density of the observed object. After computing the density of each object, the object whose density is the largest is chosen as the cluster center. Then compute densities of objects whose distances are larger than the given distance d from the selected center centers, also choose the object whose density is the largest as the second cluster center. And so forth until the number of cluster centers reaches the given number. This method ensures that cluster centers are far away from each other to avoid the objective function into local minima.

In our paper, we adopt a new method to determine cluster centers which is based on the fifth method as mentioned above. In our method, we first randomly select the observed object and compute the density of the observed object. If the density of the observed object is not less than the given density parameter, the observed object can be seen as the cluster center. Secondly we keep selecting the second cluster center satisfying the above constraints in the data set which excludes the objects which are cluster centers or objects whose distances are less than the given distance parameter. Finally we obtain the given number of cluster centers after repeating the above process. The distance parameter and the density parameter are decided by users according to the characteristics of the data sets and the priori knowledge. This selection strategy spends less time than the fifth method because time of computing densities of all objects in the data set is saved, while this method maintains the advantage of avoiding the object function into local minima.

5. Experiment and Discussion

In this section, we describe the detail information of our experiments. The traditional K-means algorithm, the traditional FCM algorithm and above mentioned improved FCM algorithm are all implemented on the platform WEKA with our experimental data. Meanwhile, our experimental results are visualized.

Currently, there are no standard metrics to measure the performance of clustering algorithms in the meteorological application field. But there are many comparison methods to compare different clustering algorithms, like iterative counts, MSE (Mean Square Error), partition coefficient and partition entropy and so on [37, 38]. In our experiment, what we use to compare the strengths and weakness of different algorithms depends on the basic metrics in the clustering field: number of iterations and squared errors, and some priori knowledge of meteorology in addition.

5.1. Data Description

Our experimental data come from the drought data in different counties of Anhui province. There are three attributes in the data set, respectively latitude, longitude and the drought area.

Table I. Experimental Datasets Summarize

Dataset	#Objects	#Dimensions	Missing
Drought Areas	463	3	Null

In Table I, the information about the dataset can be concluded that there are 463 objects with 3 attributes in the dataset. Missing represents whether the data set includes the missing values, Null means there is no missing value in the current data set.

5.2. Data Preprocessing

Generally, the experimental data set usually has some objects which have uncertain value or missing value. In our paper, we just ignore these objects to simplify the experiment. To better visualize our outcome of the experiment and effectively describe the characteristics of objects, we firstly normalize the experimental data sets after eliminating the missing objects before the clustering analysis.

Here, we adopt the min-max normalization to preprocess our experiment data.

$$v' = \frac{v - \min_A}{\max_A - \min_A} \qquad （5）$$

Where v is the original value of the attribute A, v' is the value after data normalization, $\min_A$ is the minimum value of the attribute A and $\max_A$ is the maximum value of the attribute A.

After this kind of data normalization, the value of all attributes can be limited in the interval [0, 1]. But, there is a problem about min-max normalization. It will encounter an “out-of-bounds” error if a future inputting value for normalization falls outside of the original data range of the attribute A [40].

5.3. Clustering Analysis

The main result of our experiments is that the traditional FCM algorithm has the advantage of robustness for ambiguity and maintains much more information than the traditional k-means algorithm. Besides, our proposed algorithm is more applicable to the meteorological data than the traditional FCM algorithm.

For the traditional FCM algorithm and the improved FCM algorithm, several parameters must to be specified. The fuzziness index $m = 2$, the number of clusters

$C = 3$. After setting these parameters, we can start our FCM algorithm. For the reason that we have successfully integrate the FCM algorithm and improved FCM algorithm into WEKA, what we need to do is just calling the corresponding class to satisfy our experimental purpose.

In Table II, number of iteration represents the count of the objective function needs to satisfy the constraints; cluster centers means the cluster centers of the data set; squared errors hold the squared errors for all clusters; Clustered Instances represent the distribution of all instances in the data set.

Comparing with the results of the K-means algorithm and the traditional FCM algorithm, what we can discover is that the traditional FCM algorithm needs much fewer iterative counts than the K-Means algorithm, while squared errors of the traditional FCM algorithm is larger than the K-Means algorithm. Now we consider the results of the traditional FCM algorithm and the improved FCM algorithm. These two algorithms share the same number of iterations. However the squared error of the improved FCM algorithm is smaller than that of the traditional FCM algorithm but still larger than the K-means algorithm. This phenomenon appears because the squared error is not appropriate to measure the performance of fuzzy clustering algorithms. In this paper, we determine the cluster which each object belongs to according to the membership values of each object in different clusters. This method may be not appropriate because the membership just describes the probability of belongingness of different clusters. The conclusion can be made that both the traditional FCM algorithm and the improved FCM algorithm converge faster than the K-means algorithm.

Table II. Comparison of Different Algorithms

Name of algorithm	Number of iterations	Cluster centers	squared errors	Clustered Instances
K-Means	14	0.761179 0.401403 0.109383	4.30497234165914	0 134(29%) 1 143(31%) 2 186(40%)
FCM	2	0.705194 0.300523 0.069379	83.60259931111237	0 175(37%) 1 154(33%) 2 134(28%)
Improved FCM	2	0.717584 0.085929 0.349827	80.76605579692948	0 158(34%) 1 164(35%) 2 141(30%)

Figure 1, Figure 2 and Figure 3 respectively describe the distribution of the data set with the K-means algorithm, the traditional FCM algorithm and the improved FCM algorithm. In these figures, X coordinate represents the latitude, Y coordinate represents the longitude and Z coordinate represents the drought area in the given latitude and longitude.

In terms of specific meteorological characteristics, we carefully compare the difference between Figure 1 and Figure 3. The percentage of the drought area in Si town, Suzhou city of Anhui province is 0.23873 after normalization which is obviously belong to the second cluster while K-means groups it into the third cluster, while this point is correctly classified into the second cluster in the improved FCM algorithm. This point is represented as a pentagram in Figure 4 on the basis of Figure 1. Apart from this point, there are other points misclassified into inappropriate clusters according to the priori knowledge. So the conclusion is that the improved FCM algorithm is better than the traditional K-means algorithm from the perspective of the priori meteorological knowledge.

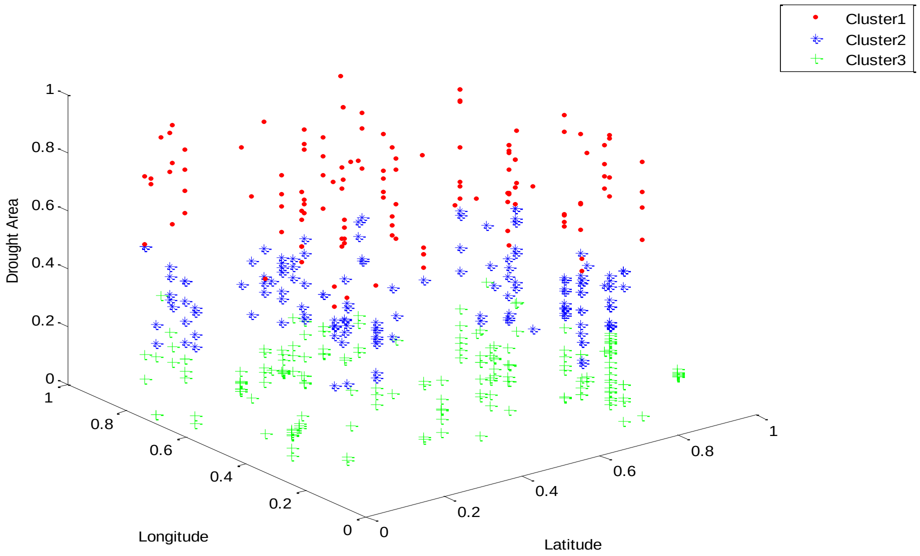


Figure 1. The Distribution of the Data Set with K-Means

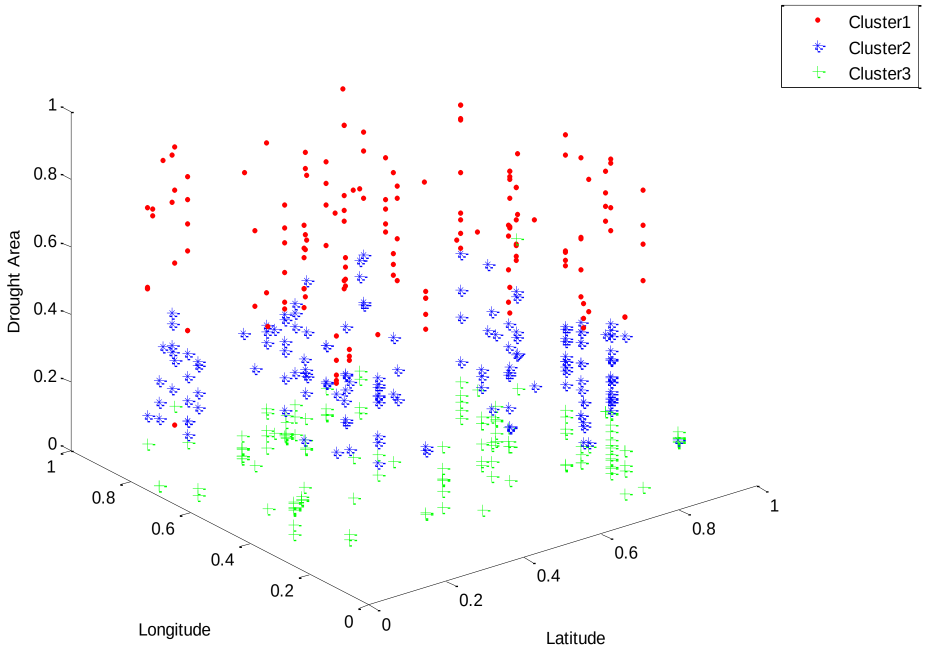


Figure 2. The Distribution of the Data Set with Traditional FCM

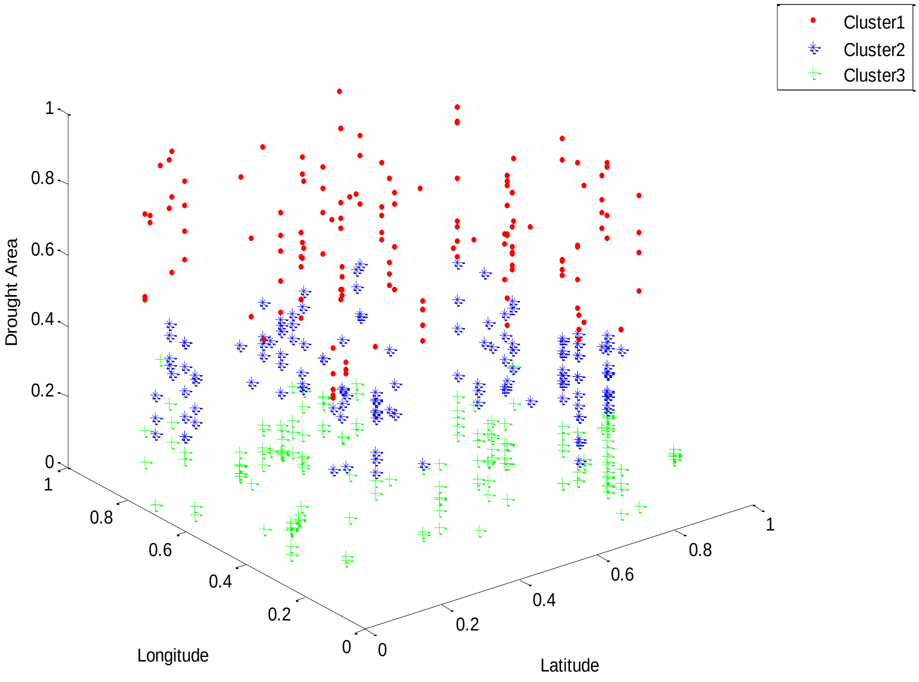


Figure 3. The Distribution of the Data Set with Improved FCM

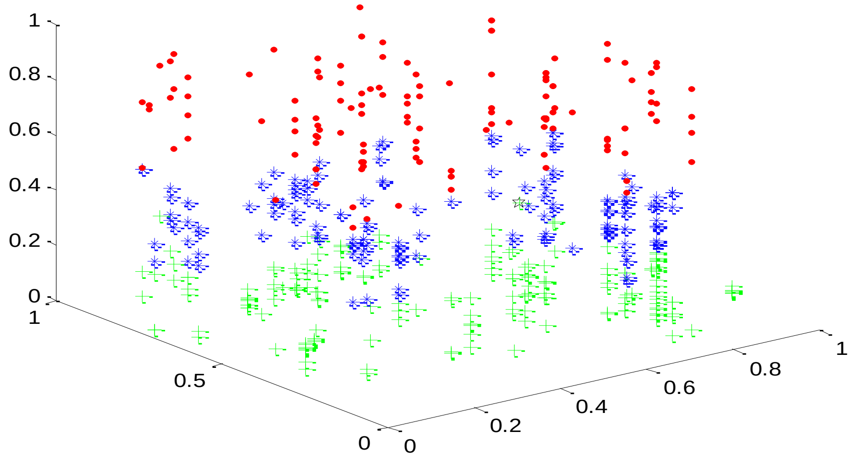


Figure 4. The Wrong Distribution of the Data Set with K-Means

6. Conclusions and Future Research

As the well-known open-source machine learning and data mining software, WEKA includes many java packages such as associations, classifiers, core, clusters and so on.

However, it doesn’t implement the traditional fuzzy clustering algorithm-FCM. In our paper, we successfully integrate the FCM algorithm into WEKA. Compared with the K-Means algorithm existing in WEKA, the FCM algorithm has few iterative counts to faster converge to the global minima than the K-Means algorithm. Furthermore, we improve the traditional FCM algorithm in term of the selection strategy of initial cluster centers to fit the characteristics of meteorological data. The improved FCM algorithm has smaller squared errors than the traditional FCM algorithm while maintaining the rapid speed of convergence. Besides, the performance of the improved FCM algorithm is better than K-means algorithm in terms of the priori meteorological knowledge. Nevertheless, this paper just improves the FCM algorithm in selecting the better cluster centers but does not consider other shortcomings of the FCM algorithm. In the future research, we will improve the performance of the FCM algorithm in the field of meteorology from other aspects.

Acknowledgements

This work was supported in part by National Science Foundation of China (No. 61173143), China Postdoctoral Science Foundation (No.2012M511783), also was supported by Qing Lan Project of Jiangsu Province and was also supported by PAPD.

References

[1] G. Karypis, E. H. Han and V. Kumar, J. Computer, vol. 32, no. 8, **(1999)**.
[2] T. Zhang, R. Ramakrishnan and M. Livny, J. Data Mining Knowledge Discovery, vol. 1, no. 2, **(1997)**.
[3] G. Sheikholeslami, S. Chatterjee and A. Zhang, Editors, A. Gupta, O. Shmueli and J. Widom. Proceedings of the 24th International Conference on Very Large Data Bases, New York, USA, **(1998)** August 24-27.
[4] W. Wang, J. Yang and R. Muntz, Editors. M. Jarke, M. J. Carey, K. R. Dittrich, F. H. Lochovsky, P. Loucopoulos and M. A. Jeusfeld, Proceedings of the 23rd International Conference on Very Large Data Bases, Athens, Greece, **(1997)** August 25-29.
[5] L. A. Zadeh, J. Infection Control, vol. 8, **(1965)**.
[6] J. Valente de Oliveira and W. Pedrycz, Editor, “Advances in Fuzzy Clustering and its Applications”, Wiley, Hoboken, **(2007)**.
[7] R. Krishnapuram and J. M. Keller, J. IEEE Transactions on Fuzzy System, vol. 1, no. 2, **(1993)**.
[8] F. L. Chung and T. Lee, J. Neural Networks, vol. 7, no. 3, **(1994)**.
[9] J. C. Bezdek, Editor, “Pattern Recognition with Fuzzy Objective Function Algorithms”, Springer Publishers, New York, **(1981)**.
[10] M. Gong, Y. Liang, W. Ma and J. Ma, J. IEEE Transactions on Image Processing, vol. 22, no. 2, **(2013)**.
[11] S. Krinidis and V. Chatzis, J. IEEE Transactions on Image Processing, vol. 19, no. 5, **(2010)**.
[12] C. C. Hung, S. Kulkarni and B. C. Kuo, J. Selected Topics in Signal Processing, vol. 5, no. 3, **(2011)**.
[13] X. Yang, J. Lu and J. Ma, J. IEEE Transactions on Fuzzy Systems, vol. 19, no. 1, **(2011)**.
[14] D. C. Hoang, R. Kumar and S. K. Panda, J. IET Wireless Sensor Systems, vol. 3, no. 3, **(2012)**.
[15] Y. Shen, J. Liu and J. Shen, Editors. IEEE Computer Society Washington, DC, USA. Proceedings of International Conference on Intelligent Computation Technology and Automation, Changsha, China, **(2010)** May 11-12.
[16] I. Charalampopoulos and I. Anagnostopoulos, Editors, IEEE Computer Society Washington, DC, USA, Proceedings of the 15th Panhellenic Conference on Informatics, Kastoria, Greece, **(2011)** September 30-October 2.
[17] E. H. Ruspini, J. Information Sciences, vol. 2, no. 3, **(1970)**.
[18] J. C. Dunn, J. Cybernetics, vol. 3, no. 3, **(1974)**.
[19] J. C. Bezdek, R. Ehrlich and W. Full, J. Computers & Geosciences, vol. 10, **(1984)**, pp. 2-3.
[20] R. L. Cannon, J. V. Dave and J. C. Bezdek, J. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 2, **(1986)**.
[21] R. Krishnapuram and J. M. Keller, J. IEEE Transactions on Fuzzy Systems, vol. 4, no. 3, **(1996)**.
[22] M. S. Yang and K. L. Wu, J. Pattern Recognition, vol. 39, no. 1, **(2006)**.

[23] H. Timm, C. Borgelt and R. Kruse, Editors. P. Sincak, J. Strackeljan, M. Kolcun, J. Bojtos, A. Toth, P. Szathmary, M. Hrehus and D. Novotny, Proceedings of Europe Symp. Intelligent Technologies, Tenerife, Spain, **(2001)** November 20-26.

[24] H. Timm and R. Kruse, Editors, IEEE, Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, Honolulu, Hawaii, **(2002)** May 12-17.

[25] H. Timm, C. Borgelt, C. Doring and R. Kruse, J. Fuzzy Sets and Systems, vol. 147, no. 1, **(2004)**.

[26] K. L. Wu and M. S. Yang, J. Pattern Recognition, vol. 35, no. 10, **(2002)**.

[27] L. Zhu, F. L. Chung and S. Wang, J. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 39, no. 3, **(2009)**.

[28] I. Ozkan and I. B. Turksen, Editors, IEEE. Proceedings of the 13th International Conference on Fuzzy Systems, Budapest, Hungary, **(2004)** July 25-29.

[29] C. Hwang and F. C. H. Rhee, J. IEEE Transactions on Fuzzy Systems, vol. 15, no. 1, **(2007)**.

[30] M. Dorigo, M. Birattari and T. Stiitzle, J. IEEE Computational Intelligence Magazine, vol. 1, no. 4, **(2006)**.

[31] P. M. Kanade and O. H. Lawrence, J. IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans, vol. 37, no. 5, **(2007)**.

[32] T. A. Runkler, J. Intelligent Systems, vol. 20, **(2005)**.

[33] J. Kennedy and R. Eberhart, Editors, IEEE, Proceedings of IEEE International Conference on Neural Network, Perth, Australia, **(1995)** November 27-December 1.

[34] T. A. Runkler and C. Katz, IEEE, Proceedings of the 15th International Conference on Fuzzy Systems, Sheraton Vancouver Wall Centre, Vancouver, BC, Canada, **(2006)** July 16-21.

[35] F. Q. Yang, T. L. Sun and C. H. Zhang, J. Expert Systems with Applications, vol. 36, no. 6, **(2009)**.

[36] R. Xu, J. Xu and D. C. Wunsch, J. IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics, vol. 42, no. 4, **(2012)**.

[37] H. Z. Zhang, H. Chen and L. X. Bao, Editors, L. Wang, L. Jiao, G. Shi and J. Liu, Proceedings of the 3rd International Conference on Fuzzy Systems and Knowledge Discovery, Xi'an, China, **(2006)** September 24-28.

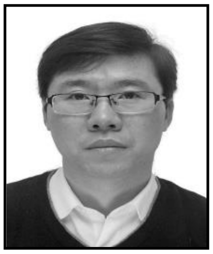
[38] X. Liu and C. Yang, Editors, "Circuits and System Society", Proceedings of the 6th International Conference on Natural Computation, Yantai, China, **(2010)** August 10-12.

[39] N. R. Pal and I. C. Bezdek, J. IEEE Transactions on Fuzzy Systems, vol. 3, no. 4, **(1995)**.

[40] S. N. Deepa, N. Malathi and C. V. Subbulakshmi, Editors, Madras and Women in Engineering, Proceedings of IEEE International Conference on Advanced Communication Control and Computing Technoligies, Ramanathapuram, India, **(2012)** August 23-25.

Authors

Yinhua Lu, received her Bachelor degree in Computer Science and Engineering from Nanjing University of Information Science & Technology, China in 2009. Currently, she is a candidate for the degree of Master of Computer Science and Engineering in Nanjing University of Information Science & Technology. Her research interests include GPU programming, image processing etc.



Tinghuai Ma, he received his Bachelor (HUST, China, 1997), Master (HUST, China, 2000), PhD (Chinese Academy of Science, 2003) and was Post-doctoral associate (AJOU University, 2004) and a visiting Professor in Kyung Hee University, Korea (KHU, 2009). Now, he is a professor in Computer Sciences at Nanjing University of Information Science & Technology, China. His research interests are data mining, grid computing, ubiquitous computing, privacy preserving, *etc*.

Wei Tian, he is an assistant professor of Computer Sciences at Nanjing University of Information Science & Technology, China. He received his master degree from Nanjing University of Information Science & Technology, China, 2006. Now, he is a doctoral candidate of Applied Meteorology, Nanjing University of Information and Science Technology. His main research interests are in the areas of Cloud Copmputing and Meteorological Data Processing.

Shuiming Zhong, he received the M.S. and Ph.D. degrees from the Hohai University, China, in 2007 and 2011 respectively. He currently is a lecturer in the School of Computer and Software, Nanjing University of Information Science and Technology, China. His research interests involve artificial neural networks, machine learning, and pattern recognition.