Research on Vague soft clustering algorithm based on MapReduce

Wei Wang[1], Junsheng Wu[1], Zhixiang Zhu[2]

[1]School of software and microelectronics, Northwestern Poly technical University, Xi'an 710072, China;

[2]Institute of internet of things & integration of information and industrialization, University of posts and telecommunication, Xi'an 710061, China

szbwangw@163.com

Keywords: Vague soft sets, clustering, MapReduce.

Abstract. Aiming at the problems that traditional clustering algorithm based on Vague soft sets is difficult to deal with massive data, a parallel Vague soft clustering algorithm based on MapReduce is proposed. The algorithm calculate the similarity measure between Vague soft sets based on Map function and Reduce function of the MapReduce programming framework, and Vague similarity matrix is established as the basis for clustering at the same time. Secondly, the Vague matrix is partitioned according to the similarity matrix based on the idea of matrix partition, and we process the block matrix and merge the results based on MapReduce. Finally, the data items is clustered based on the similarity threshold. The contrast experiment between traditional Vague soft clustering algorithm and the new algorithm shows that the proposed algorithm has higher accuracy, it can not only achieve better speedup in large-scale data calculation but also divide project resources effectively and accurately.

1. Introduction

It is known that Vague sets theory[1]proposed by Gau and Buehrer in 1993 and soft sets theory[2] proposed by Moldtsov in 1999 are two new mathematical model to deal with the ambiguity problem, both two theories focus on the uncertainty, incompleteness and inaccuracy of information system from different angles, it is widely used in pattern recognition, data mining, fuzzy decision making, image retrieval and other practical problems at present. In practice Vague sets and soft sets theory are both interrelated and complementary each other, so they can be integrated in order to make up for their deficiencies and develop respective advantages. According to the combination problem based on Vague sets and soft sets, Vague soft sets and some related key techniques, such as some properties, similarity measures between Vague soft sets are presented by paper[3].One new clustering algorithm based on the similarity between the interval valued fuzzy soft sets (Vague soft sets) is proposed by paper[9], the computational complexity of the algorithm mainly comes from the computational complexity of the similarity between interval valued fuzzy soft sets (Vague soft sets), it was clear that the algorithm have a certain degree influence on the timeliness and the utilization of resources when dealing with massive data. It is well known that MapReduce[10]proposed by Google is not only a distributed parallel programming model for large scale data sets but also the core computing model in the cloud computing environment. At present, various fuzzy clustering algorithms[11-15]based on MapReduce are proposed by many scholars, however the parallelization clustering algorithm based on Vague soft sets has not been studied yet. Therefore, the traditional Vague soft clustering algorithm can be improved to adapt to the MapReduce parallel programming model in order to solve the problem of Vague soft clustering in massive data effectively.

According to the above consideration, one novel parallel Vague soft clustering algorithm based on MapReduce is proposed in this paper. First of all, we calculate the similarity of the project data based on the Map function and Reduce function in the MapReduce programming framework in order to establish the Vague similarity matrix; Secondly, the project is divide into several similar classes based on MapReduce according to the threshold value of Vague similarity matrix; Finally, the clustering problem of data points in large scale based on Vague soft sets project is completed. The experimental results show that the improved algorithm is superior to the traditional Vague soft clustering algorithm on accuracy and speedup performance.

2. Preliminaries

In this section, some preliminaries on the theory about Vague soft sets and MapReduce are presented.

2.1 Vague Soft Sets.

Definition 1. (Vague soft sets) Let U be a universal set and let E be a set of parameters, $A \subseteq E$, $F : A \to P(U)$ is a mapping,That is, $\forall e \in A$, $F(e)$ is a Vague sets into the power set of U , Also, (F, A) is considered as a Vague soft sets on U .

Definition 2. (Similarity measure between Vague soft sets) Let $VSS(U)$ be a Vague soft sets on universal set U , and let E be a set of parameters, $(F, E), (G, E) \in VSS(U)$, Function $M : VSS(U) \times VSS(U) \to [0,1]$ is considered as the similarity measure between Vague soft sets, if it meets the following conditions:

Rule 1 Boundedness: $M((F, E), (G, E)) \in [0,1]$;

Rule 2 Symmetry: $M((F, E), (G, E)) = M((G, E), (F, E))$;

Rule 3 Normalization: $M((F, E), (G, E)) = 1 \Leftrightarrow (F, E) = (G, E)$;

Rule 4 Monotonicity: if $(F, E) \subseteq (G, E) \subseteq (H, E)$,then $M((F, E), (H, E)) \le \min(M((F, E), (H, E)), M((G, E), (H, E)))$.

Based on the axiomatic definition of similarity measure between Vague soft sets, we can see that the larger of similarity measure between two Vague soft sets , the similar between two Vague soft sets.

Theorem 1. Let $U = \{x_1, x_2, \cdots, x_n\}$ be a universal set, $E = \{e_1, e_2, \cdots, e_m\}$ be a set of parameters, $VSS(U)$ be a Vague soft sets on universal set U , and $(F, E), (G, E) \in VSS(U)$, the similarity measure between Vague soft sets is as follows:

$$M((F, E), (G, E)) = \sum_{i=1}^{m} \lambda_i \left\{ 1 - \frac{1}{7n} \sum_{j=1}^{n} \left[\begin{array}{l} \left|t_{F(e_i)}(x_j) - t_{G(e_i)}(x_j)\right| + \left|f_{F(e_i)}(x_j) - f_{G(e_i)}(x_j)\right| \\ + \left|\pi_{F(e_i)}(x_j) - \pi_{G(e_i)}(x_j)\right| + \left|S_{F(e_i)}(x_j) - S_{G(e_i)}(x_j)\right| \\ + \left|\phi_{F(e_i)}(x_j) - \phi_{G(e_i)}(x_j)\right| \end{array} \right] \right\};$$

Which, $\pi_{F(e_i)}(x_j) = 1 - t_{F(e_i)}(x_j) - f_{F(e_i)}(x_j)$, $\pi_{\mathrm{G}(e_i)}(x_j) = 1 - t_{\mathrm{G}(e_i)}(x_j) - f_{G(e_i)}(x_j)$ be the degree of hesitation of two Vague soft sets, it characterizes the waiver of existing evidence of x_j for parameters e_i ; $S_{F(e_i)}(x_j) = t_{F(e_i)}(x_j) - f_{F(e_i)}(x_j)$, $S_{\mathrm{G}(e_i)}(x_j) = \mathrm{t}_{\mathrm{G}(e_i)}(x_j) - f_{G(e_i)}(x_j)$ be the core of the element of two Vague soft sets, it characterizes the comparison of the existing evidence to the two forces of support and opposition of existing evidence of x_j ; $\phi_{F(e_i)}(x_j) = \frac{1 - t_{F(e_i)}(x_j) + f_{F(e_i)}(x_j)}{2}$, $\phi_{G(e_i)}(x_j) = \frac{1 - t_{G(e_i)}(x_j) + f_{G(e_i)}(x_j)}{2}$ be the interval center of the element of two Vague soft sets $F(e_i)$ and $G(e_i)$. λ_i is Weight of parameters e_i .

Proof is omitted.

2.2 MapReduce

It is known that MapReduce is a parallel programming model for large scale data sets (more than 1TB). Hadoop proposed by Apache is not only an open source implementation of MapReduce but also the parallel processing standards of current academia and industry for massive data in fact. First of all, based on the idea of "divide and rule", MapReduce separated and processed the data sets into thousands of small data sets, Then these small data sets are handed over to the nodes in the cluster by NameNode, because of the parallel performance of these nodes, a task can be assigned to multiple nodes and the overall computational efficiency is improved also. A large number of calculation results obtained by Map function is calculated by Reduce function again and aggregated to form a new calculation result finally.

The framework of MapReduce is shown in figure 1:

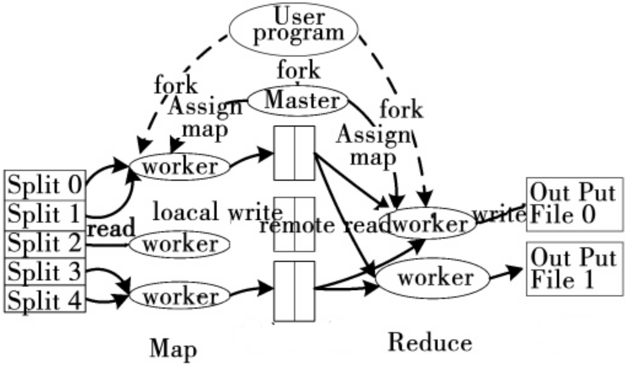


Fig.1 The framework of MapReduce

As shown above, MapReduce runs on the key pair, the framework looks at the input of the work as a set of key pairs, and also produces a set of key values for the output of the job, it is clear that the two sets of key pair types may be different. The input and output types of a MapReduce job are as follows:

$$(input)\langle k1,v1\rangle \to map \to \langle k2,v2\rangle \to combine \to \langle k2,v2\rangle \to reduce \to \langle k3,v3\rangle(output)$$

3. The Clustering Algorithm based on Vague Soft Sets

It is worth to mention here the tradition clustering algorithm based on Vague soft sets in this section.

Algorithm 1 Let $U = \{x_1, x_2, \cdots, x_n\}$ be a universal set, $E = \{e_1, e_2, \cdots, e_m\}$ be a set of parameters, $VSS(U)$ be a Vague soft sets on universal set U , $x \in U$, Each element $VSS(x_i)(i = 1,2,\cdots,n) = \{X_i(e_j)|j = 1,2,\cdots,m\}$ has n characteristic index, $VSS(x_i)(i = 1,2,\cdots,n) = \lfloor t_{(e_j)}(x_i), 1 - f_{(e_j)}(x_i)\rfloor(j = 1,2,\cdots,m)$ be the characteristic index of the object i. Then $VSS(x_i)$can be represented by a vector matrix as the following dimensions:

$$\begin{bmatrix} VSS(u_1) \\ VSS(u_2) \\ \vdots \\ VSS(u_n) \end{bmatrix} = \begin{bmatrix} [t(x_1), 1 - f(x_1), e_1] & [t(x_1), 1 - f(x_1), e_2] & \cdots & [t(x_1), 1 - f(x_1), e_m] \\ [t(x_2), 1 - f(x_2), e_1] & [t(x_2), 1 - f(x_2), e_2] & \cdots & [t(x_2), 1 - f(x_2), e_m] \\ \vdots & \vdots & \vdots & \vdots \\ [t(x_n), 1 - f(x_n), e_1] & [t(x_n), 1 - f(x_n), e_2] & \cdots & [t(x_n), 1 - f(x_n), e_m] \end{bmatrix};$$

According to the similarity measure between Vague soft sets:

$$M((F, E), (G, E)) = \sum_{i=1}^{m} \lambda_i \left\{ 1 - \frac{1}{7n} \sum_{j=1}^{n} \left[\begin{array}{l} |t_{F(e_i)}(x_j) - t_{G(e_i)}(x_j)| + |f_{F(e_i)}(x_j) - f_{G(e_i)}(x_j)| \\ + |\pi_{F(e_i)}(x_j) - \pi_{G(e_i)}(x_j)| + |S_{F(e_i)}(x_j) - S_{G(e_i)}(x_j)| \\ + |\phi_{F(e_i)}(x_j) - \phi_{G(e_i)}(x_j)| \end{array} \right] \right\}; \quad (1)$$

We calculate the similarity matrix based on Vague soft sets as follows:

$$R = \begin{bmatrix} [VSS(X_1), VSS(X_1), E] & [VSS(X_1), VSS(X_2), E] & \cdots & [VSS(X_1), VSS(X_n), E] \\ [VSS(X_2), VSS(X_1), E] & [VSS(X_2), VSS(X_2), E] & \cdots & [VSS(X_2), VSS(X_n), E] \\ \vdots & \vdots & \vdots & \vdots \\ [VSS(X_n), VSS(X_1), E] & [VSS(X_n), VSS(X_2), E] & \cdots & [VSS(X_n), VSS(X_n), E] \end{bmatrix};$$

The clustering algorithm based on Vague soft sets is summarized as follows:

(1) Take $\lambda_1 = 1$ (the maximum value in R) ,merge u_i and u_j to a similar class ,if $M(VSS(X_i), VSS(X_j), E) = 1$;

(2) Take λ_2 as the second largest value in R, and look for the similarity of elements λ_2 from R, that is, $M(VSS(X_i), VSS(X_j), E) = \lambda_2$, merge u_i and u_j to a similar class.

(3) Take λ_3 as the third largest value in R, the operation is the same as (2).

(4) Repeat (1) - (3) until the data sets merged into a single equivalence class. End.

Example 1 Let the object to be classified $U = \{u_1.u_2, \cdots, u_5\}$ as the Vague soft sets on the universe of discourse U, $E = \{e_1, e_2, e_3, e_4\}$ be a set of parameters, by processing, each object in the parameter set of each feature is shown as table 1:

Table 1. The object to be classified $VSS(U_i, E)$

$VSS(U_i)$	e_1	e_2	e_3	e_4
$VSS(U_1)$	$[0.6,0.8]$	$[0.7,0.8]$	$[0.4,0.7]$	$[0.4,0.9]$
$VSS(U_2)$	$[0.7,0.8]$	$[0.5,0.7]$	$[0.2,0.6]$	$[0.5,0.6]$
$VSS(U_3)$	$[0.5,0.7]$	$[0.5,0.8]$	$[0.3,0.7]$	$[0.7,0.9]$
$VSS(U_4)$	$[0.3,0.6]$	$[0.7,0.9]$	$[0.2,0.7]$	$[0.4,0.7]$
$VSS(U_5)$	$[0.5,0.8]$	$[0.5,0.9]$	$[0.2,0.8]$	$[0.6,0.8]$

$VSS(U_i, E)$ can be represented by Vague matrix as follows:

$$\begin{bmatrix} VSS(u_1) \\ VSS(u_2) \\ VSS(u_3) \\ VSS(u_4) \\ VSS(u_5) \end{bmatrix} = \begin{bmatrix} [0.6,0.8] & [0.7,0.8] & [0.4,0.7] & [0.4,0.9] \\ [0.7,0.8] & [0.5,0.7] & [0.2,0.6] & [0.5,0.6] \\ [0.5,0.7] & [0.5,0.8] & [0.3,0.7] & [0.7,0.9] \\ [0.3,0.6] & [0.7,0.9] & [0.2,0.7] & [0.4,0.7] \\ [0.5,0.8] & [0.5,0.9] & [0.2,0.8] & [0.6,0.8] \end{bmatrix};$$

The clustering results are shown in table 2:

Table 2. The clustering results of $VSS(U_i, E)$

Threshold	**Results**	**Specific classification**
1	5 class	$\{VSS(U_1,E)\};\{VSS(U_2,E)\};\{VSS(U_3,E)\};\{VSS(U_4,E)\};\{VSS(U_5,E)\}$
0.972	4 class	$\{VSS(U_1,E)\};\{VSS(U_2,E)\};\{VSS(U_4,E)\};\{VSS(U_3,E),VSS(U_5,E)\}$
0.954	3 class	$\{VSS(U_2,E)\};\{VSS(U_4,E)\};\{VSS(U_1,E),VSS(U_3,E),VSS(U_5,E)\}$
0.945	2 class	$\{VSS(U_4,E)\};\{VSS(U_1,E),VSS(U_2,E),VSS(U_3,E),VSS(U_5,E)\}$
0.933	1 class	$\{VSS(U_1,E),VSS(U_2,E),VSS(U_3,E),VSS(U_4,E),VSS(U_5,E)\}$

As shown in table 2, the clustering results are summarized as follows:

Step1 Threshold is 1, the clustering results for the class is 5:
$\{VSS(U_1, E)\}; \{VSS(U_2, E)\}; \{VSS(U_3, E)\}; \{VSS(U_4, E)\}; \{VSS(U_5, E)\};$

Step2 Threshold is 0.972, the clustering results for the class is 4:
$\{VSS(U_1, E)\}; \{VSS(U_2, E)\}; \{VSS(U_4, E)\}; \{VSS(U_3, E), VSS(U_5, E)\};$

Step3 Threshold is 0.954, the clustering results for the class is 3:
$\{VSS(U_2, E)\}; \{VSS(U_4, E)\}; \{VSS(U_1, E), VSS(U_3, E), VSS(U_5, E)\};$

Step4 Threshold is 0.945, the clustering results for the class is 2:
$\{VSS(U_4, E)\}; \{VSS(U_1, E), VSS(U_2, E), VSS(U_3, E), VSS(U_5, E)\};$

Step5 Threshold is 0.933, the clustering results for the class is 1:
$\{VSS(U_1,E),VSS(U_2,E),VSS(U_3,E),VSS(U_4,E),VSS(U_5,E)\};$

End.

The experimental results show that the algorithm is an effective clustering algorithm based on Vague soft sets.

4. The Parallel Design of Clustering Algorithm based on Vague Soft Sets

The analysis shows that in order to achieve Vague soft clustering algorithm based on MapReduce, we need the processing of data initialization, Map function, Reduce function realization, etc. Firstly, the data records that need to be processed are stored in the form of rows, so that the correlation between the data to be processed is reduced, and can meet the needs of the block calculation. In the first stage of the algorithm, the similarity measure of the data points should be calculated as the basis of direct clustering. It is found that a large number of similarity measures between computational data points are needed in the process of computing the Vague similarity matrix, According to the formula (1), it is necessary to calculate the true membership value, the false membership value, the degree of hesitation, the core and the interval center value of each data point. With the increase of the amount of data the time complexity will gradually increase, which seriously affecting the efficiency of the algorithm, so this step of the algorithm can be realized based on MapReduce parallel programming framework in order to improve the algorithm performance. In addition, when calculating the similarity measure between Vague soft sets, if we take the key X to the Map-Reduce direct, we have to send all the data to each node that caused the complexity of the algorithm is large. To solve this problem, we can use two Map-Reduce process to complete. The first MR procedure is inverted index. For each data point, the Vague value of each data point is characterized as a key, and the data point label and the digital characteristic values are output:

$$\left\langle t, \left[\left(x_1, v_1\right), \left(x_2, v_2\right), \cdots, \left(x_n, v_n\right)\right]\right\rangle;$$

$$\left\langle f, \left[\left(x_1, v_1\right), \left(x_2, v_2\right), \cdots, \left(x_n, v_n\right)\right]\right\rangle;$$

$$\left\langle \pi, \left[\left(x_1, v_1\right), \left(x_2, v_2\right), \cdots, \left(x_n, v_n\right)\right]\right\rangle;$$

$$\left\langle S, \left[\left(x_1, v_1\right), \left(x_2, v_2\right), \cdots, \left(x_n, v_n\right)\right]\right\rangle;$$

$$\left\langle \phi, \left[\left(x_1, v_1\right), \left(x_2, v_2\right), \cdots, \left(x_n, v_n\right)\right]\right\rangle.$$

The second MR process is to calculate the similarity.

We take the output of above Map-Reduce process as the input, and take the data pair as the key in the Map process. The absolute value of the difference of the numerical characteristics calculated by the formula (1) is the output, and the output of each digital characteristic value converted by the process of Map is shown as follows:

$$\left\langle \left[\left(x_1, x_2\right), \left|t_1 - t_2\right|\right], \left[\left(x_1, x_3\right), \left|t_1 - t_3\right|\right], \cdots, \left[\left(x_n, x_{n-1}\right), \left|t_n - t_{n-1}\right|\right]\right\rangle;$$

$$\left\langle \left[\left(x_1, x_2\right), \left|f_1 - f_2\right|\right], \left[\left(x_1, x_3\right), \left|f_1 - f_3\right|\right], \cdots, \left[\left(x_n, x_{n-1}\right), \left|f_n - f_{n-1}\right|\right]\right\rangle;$$

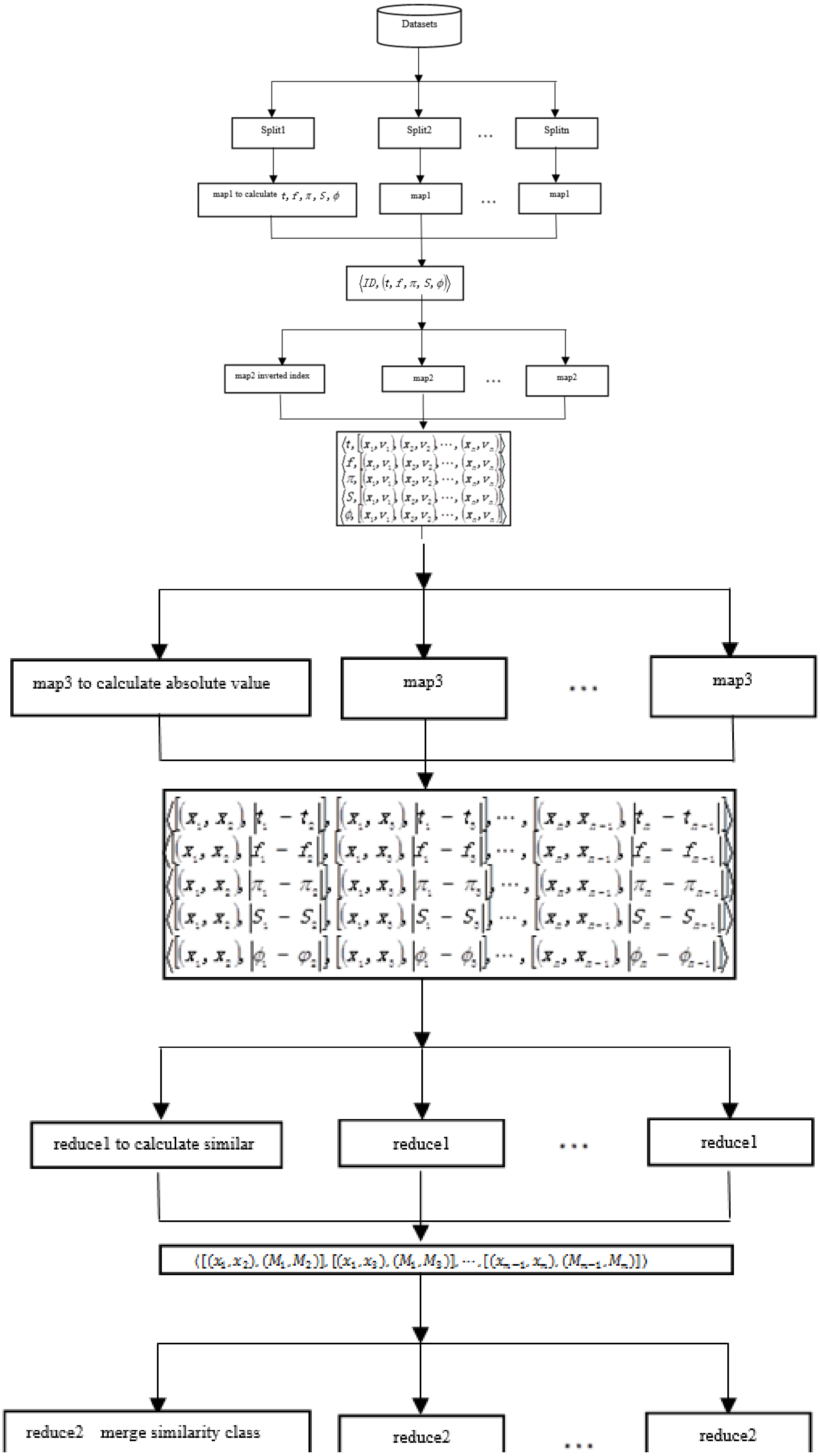
$$\left\langle \left[\left(x_1, x_2\right), \left|\pi_1 - \pi_2\right|\right], \left[\left(x_1, x_3\right), \left|\pi_1 - \pi_3\right|\right], \cdots, \left[\left(x_n, x_{n-1}\right), \left|\pi_n - \pi_{n-1}\right|\right]\right\rangle;$$

$$\left\langle \left[\left(x_1, x_2\right), \left|S_1 - S_2\right|\right], \left[\left(x_1, x_3\right), \left|S_1 - S_3\right|\right], \cdots, \left[\left(x_n, x_{n-1}\right), \left|S_n - S_{n-1}\right|\right]\right\rangle;$$

$$\left\langle \left[\left(x_1, x_2\right), \left|\phi_1 - \varphi_2\right|\right], \left[\left(x_1, x_3\right), \left|\phi_1 - \phi_3\right|\right], \cdots, \left[\left(x_n, x_{n-1}\right), \left|\phi_n - \phi_{n-1}\right|\right]\right\rangle \circ$$

Then, according to the formula (1), we calculate the average value of the same key weighted in order to obtain the similarity measure of all data pairs. The second phase of the algorithm is the clustering, which can be achieved by MapReduce programming framework also. Firstly, the Reduce function is used to select different thresholds to merge similar classes until the data items are merged into a single equivalence class. Finally, the clustering result is output. Algorithm end.

The flow chart of vague soft clustering algorithm based on MapReduce is shown in figure 2.


Datasets
Split1
Split2
…
Splitn
map1 to calculate t, f, π, S, ϕ
map1
…
map1
$\langle ID, (t, f, \pi, S, \phi)\rangle$
map2 inverted index
map2
…
map2
$\langle t, [(x_1, v_1), (x_2, v_2), \cdots, (x_n, v_n)]\rangle$
$\langle f, [(x_1, v_1), (x_2, v_2), \cdots, (x_n, v_n)]\rangle$
$\langle \pi, [(x_1, v_1), (x_2, v_2), \cdots, (x_n, v_n)]\rangle$
$\langle S, [(x_1, v_1), (x_2, v_2), \cdots, (x_n, v_n)]\rangle$
$\langle \phi, [(x_1, v_1), (x_2, v_2), \cdots, (x_n, v_n)]\rangle$
map3 to calculate absolute value
map3
…
map3
$\langle [(x_1, x_2), \lvert t_1 - t_2\rvert], [(x_1, x_3), \lvert t_1 - t_3\rvert], \cdots, [(x_n, x_{n-1}), \lvert t_n - t_{n-1}\rvert]\rangle$
$\langle [(x_1, x_2), \lvert f_1 - f_2\rvert], [(x_1, x_3), \lvert f_1 - f_3\rvert], \cdots, [(x_n, x_{n-1}), \lvert f_n - f_{n-1}\rvert]\rangle$
$\langle [(x_1, x_2), \lvert \pi_1 - \pi_2\rvert], [(x_1, x_3), \lvert \pi_1 - \pi_3\rvert], \cdots, [(x_n, x_{n-1}), \lvert \pi_n - \pi_{n-1}\rvert]\rangle$
$\langle [(x_1, x_2), \lvert S_1 - S_2\rvert], [(x_1, x_3), \lvert S_1 - S_3\rvert], \cdots, [(x_n, x_{n-1}), \lvert S_n - S_{n-1}\rvert]\rangle$
$\langle [(x_1, x_2), \lvert \phi_1 - \varphi_2\rvert], [(x_1, x_3), \lvert \phi_1 - \phi_3\rvert], \cdots, [(x_n, x_{n-1}), \lvert \phi_n - \phi_{n-1}\rvert]\rangle$
reduce1 to calculate similar
reduce1
…
reduce1
$\langle [(x_1, x_2), (M_1, M_2)], [(x_1, x_3), (M_1, M_3)], \cdots, [(x_{n-1}, x_n), (M_{n-1}, M_n)]\rangle$
reduce2 merge similarity class
reduce2
…
reduce2

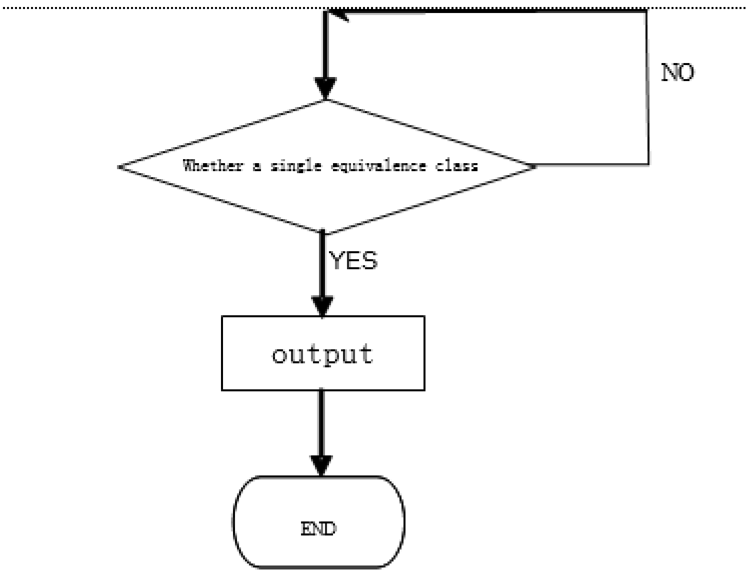


Fig. 2 the flow chart of vague soft clustering algorithm based on MapReduce

4.1 To Calculate the Digital Characteristics of Each Data Point.

The main task of this stage is to calculate the true membership value, the false membership value, the degree of hesitation, the score, and the interval center value of each data point. The pseudo-code of the algorithm is as follows:

Algorithm 1 To calculates $t_{F(e_i)}(x_j)$、 $f_{F(e_i)}(x_j)$、 $\pi_{F(e_i)}(x_j)$、 $S_{F(e_i)}(x_j)$and$\phi_{F(e_i)}(x_j)$ of the data points.

```
imput:(key,value)
output: <ID, ($t, f, \pi, S, \phi$)>
function VSSclusterMap1(key,value,ID)
{
while i=1 to n do // All initialized Vague soft set data items;
Truthmembership= Truth_value(ID,value) // To calculate the true membership values of all data points;
Falsemembership= False_value(ID,value) // To calculate the false membership values of all data points;
Hesitancydegree =$\pi_{F(e_i)}(x_j) = 1 - t_{F(e_i)}(x_j) - f_{F(e_i)}(x_j)$// To calculate the degree of hesitation of all data points;
Scoremembership=$S_{F(e_i)}(x_j) = t_{F(e_i)}(x_j) - f_{F(e_i)}(x_j)$// To calculate the score values of all data points;
Intervalcentervalues =$\phi_{F(e_i)}(x_j) = \frac{1 - t_{F(e_i)}(x_j) + f_{F(e_i)}(x_j)}{2}$ // To calculate the interval center values of all data points;
output <ID, ($t, f, \pi, S, \phi$)>;
}
```

4.2 To Calculate the Similarity between Data Points.

The main task of this stage is to calculate the similarity measure between each data point, as the basis of the next stage of direct clustering. The pseudo code of the algorithm is as follows:

Algorithm 2 To calculate the similarity measure between data points.

```
imput：<ID, ($t, f, \pi, S, \phi$)>
output: $\langle (x_i, x_j), M(VSS(X_i), VSS(X_j), E) \rangle$
function VSSclusterMap2(<ID, ($t, f, \pi, S, \phi$)>)//inverted index;
{
while i=1 to n do // All Vague soft data items;
```

```
    Invertedindex (ID,($t, f, \pi, S, \phi$)); // Take all Vague soft data items as the key to the data point labels
and the number of features as output in order to inverted index;
    output$\langle (t, f, S, \pi, \phi), [(x_1, v_1), (x_2, v_2), \cdots, (x_n, v_n)] \rangle$;
    }
    function VSSclusterMap3(<ID, ($t, f, \pi, S, \phi$)>)//To calculate the absolute value of the difference
of the attribute value;
    {
    while i=1 to n do // All Vague soft data items;
    Distance=Euclidean_distance(<ID, ($t, f, \pi, S, \phi$)>)//To calculte the Euclidean distance of the
difference of attribute values of all data points;
    output$\langle (x_i, x_j), (|t_i - t_j|, |f_i - f_j|, |S_i - S_j|, |\pi_i - \pi_j|, \phi_i - \phi_j) \rangle$;
    }
    function VSSclusterReduce1( $\langle (x_i, x_j), (|t_i - t_j|, |f_i - f_j|, |S_i - S_j|, |\pi_i - \pi_j|, \phi_i - \phi_j) \rangle$ )//To calculte the
similarity;
    {
    while i=1 to n do // All Vague soft data items;
    Measuremembership=Measure ($\langle (x_i, x_j), (|t_i - t_j|, |f_i - f_j|, |S_i - S_j|, |\pi_i - \pi_j|, \phi_i - \phi_j) \rangle$)
    //To calculate the similarity between data points according to formula (1);
    output$\langle (x_i, x_j), M(VSS(X_i), VSS(X_j), E) \rangle$;
    }
```

4.3 To Direct Clustering based on Similarity.

The main task of this stage is to accurate clustering, The pseudo code of the algorithm is as follows:

Algorithm 3 To direct clustering based on similarity measure between Vague soft sets:

```
impute:$\langle (x_i, x_j), M(VSS(X_i), VSS(X_j), E) \rangle$
output: <ID, cluster>
function VSSclusterReduce2($\langle (x_i, x_j), M(VSS(X_i), VSS(X_j), E) \rangle$)
{
merge(cluster,(membership, value))// to merge all sample points of the same class;
if Convergence(cluster)//To determine whether a single equivalence class;
output(cluster,(membership,value));
else VSSclusterReduce2(membership,value);
}
```

5. Experiment and Result Analysis

5.1 Experimental Environment and Data Sets.

This experiment runs on a cluster composed of 7 computers and uses the Hadoop distributed framework under the Apache foundation. One of the 1 machines as the main node that is NameNode (or JobTracker) node, the remaining 6 machines as a node from the Data-Node (or TaskTracker) node. The hardware configuration of each machine is as follows: CPU type is Intel Xeon7420 with 64 quad core processor, support for virtualization, frequency is 2.13GHz, the memory size is 64G, hard disk size is 6T, the operating system is Ubuntu 13.10, Ruijie RG-S2928G-E Gigabit switch, development tools and platform for Eclipse 8.5, JDK 1.7, Hadoop 2.7.1.

Experimental data used a large number data collected from the real micro-blog on the public opinion platform. The platform collected real-time from 200 uninterrupted servers group involving the national, global key sites, forums, the site of the 150000, micro-blog, other domestic and foreign data. At present, the data set has been collected to cover more than 350000 acquisition points, more than 1 hundred million of micro-blog blogger information, micro-blog storage volume of 10 million.

Experiment research on clustering micro-blog hot topics, respectively from the analysis of clustering accuracy and recall of REC&PRE to the quality of clustering algorithm, the speedup from Sp to measure MapReduce block fuzzy clustering based on parallel performance and effect.

5.2 Speedup Analysis.

In order to test the performance of the algorithm, the experimental group were randomly selected 5 data sets were tested, 3000,10000, 100000, 500000, 1000000 respectively from the micro-blog data, considering the scale, diversity, speed, the value of the 4 parameters of the special characteristics of public opinion of micro-blog, its weight is $\{0.29, 0.31, 0.18, 0.22\}$.For each set of data, the Vague soft clustering algorithm based on Mapreduce runs for 8 times, and the speedup of the algorithm is shown in table 4:

Table 4. Speedup analysis

Data sets	3000	10000	100000	500000	1000000
Speedup	0.447	1.141	2.447	3.834	7.737

From the experimental results, we can see that when the data set is small, the runtime of the algorithm in the Hadoop distributed framework is longer than that in the single machine environment. The main reason is that the data set of MapReduce and the merging of the clustering result take more time; With the increasing amount of data, the running time of the clustering algorithm in the Hadoop distributed framework is significantly lower than that in the single machine environment, The larger the amount of data ,the more obvious the advantage of parallel computing, and the stronger the ability of Hadoop system to deal with large data sets. The experimental results show that the Vague soft clustering algorithm based on MapReduce can get a better speedup ratio for large scale data processing.

5.3 Algorithm Accuracy and Recall Analysis.

Because Vague soft sets clustering is affected by the similarity threshold selection between the Vague soft sets, the experiments are carried out with several different similarity thresholds, For each threshold calculated average clustering accuracy and average recall, the results show that the average accuracy and recall rate of Vague soft clustering based on MapReduce algorithm in 5 data sets is higher than the traditional Vague soft clustering algorithm. The experimental results are shown in table 5.

Table 5. PRE and REC analysis

Data sets	**evaluating indicator**	**3000**	**10000**	**100000**	**500000**	**1000000**
Traditional Vague soft clustering algorithm	PRE	0.88	0.75	0.65	0.59	0.53
	REC	0.87	0.77	0.74	0.71	0.68
Vague soft clustering algorithm based on Mapreduce	PRE	0.91	0.83	0.79	0.71	0.66
	REC	0.94	0.85	0.83	0.79	0.75

The results show that the accuracy and recall rate of the two algorithms are more than 0.85 when the clustering data sets are small,.However, when the data samples are increasing, the accuracy and recall of the traditional Vague soft clustering algorithm are significantly different from those of the parallel clustering algorithm based on MapReduce, The reason is because when the amount of data increases, there will be a lot of non spherical irregular clusters in the data sets, while the traditional Vague soft clustering algorithm does not have a good clustering effect for non spherical clusters. The accuracy and recall rate of the vague soft clustering algorithm based on MapReduce is superior to the traditional Vague soft clustering algorithm.

6. Conclusion

In this paper, We propose a Vague soft clustering algorithm based on MapReduce in order to solve the problem of Vague soft clustering for large scale, compared with the traditional Vague soft clustering algorithm, the proposed algorithm has better clustering results in terms of accuracy and recall rate, and can obtain higher speedup in the calculation of large-scale data. Our future work will focus on how to optimize the Vague soft clustering algorithm based on MapReduce.

Acknowledgments

The National Natural Science Funds （61075014/F030303）

References

[1]. Gau W L., Buehrer D J.Vague sets [J].IEEE Transactions on Systems, Man, and Cybmetics,1993, 23(2): 610-614.

[2]. Molodtsov D.Soft set theory-first results [J]. Ccomputers and mathematics with applications, 1999,37: 19 -31.

[3]. Wei X., Jian M., Shou W., Gang H. Vague soft sets and their properties [J]. Computers & Mathematics with Applications, 2010, 59(2): 787-794.

[4]. Ganeshsree S. Vague Soft Rings and Vague Soft Ideals [J]. International Journal of Pure and Applied Mathematics, 2012, 6 (12): 557- 572.

[5]. Yun Y., Young J., Jianming Z. Vague soft hemirings[J]. International Journal of Pure and Applied Mathematics, 2011, 62(1): 199-213.

[6]. Alhazaymeh K. Generalized Vague Soft Set And Its Applications [J]. International Journal of Pure and Applied Mathematics, 2012, 77(3): 1-401.

[7]. Alhazaymeh K., Nasruddin H. Interval-valued vague soft sets and its application [J]. Advances in Fuzzy Systems, 2012, 2012(15): 1077-1083.

[8]. Teng Y., Wang C. Multicriteria fuzzy decision-making method based on Vague soft sets[J]. Computer Engineering and Applications, 2012, 48(10): 6-8.

[9]. PENG Xindong., YANG Yong.. Information measures for interval-valued fuzzy soft sets and their clustering algorithm [J]. Journal of Computer Applications, 2015,35(8):2350-2354.

[10]. LI Jian-jiang., CUI Jian., WANG Dan., YAN Lin., HUANG Yi-shuang.Survey of MapReduce Parallel Programming Model[J]. Chinese Journal of Electronics, 2011,39(11):2635-2642.

[11]. Gao xianwe., Shi zhibi. Hadoop secondary parallel Fuzzy c-Means clustering algorithm[J]. Computer Measurement & Control,2015,23(3):842-846.

[12]. ZHANG Guangrong., CHEN Qingkui., ZHANG Gang., ZHAO Haiyan., GAO Liping., HUO Huan. Parallel fuzzy partition algorithm based on MapReduce [J]. Journal of Computer Applications, 2014, 34(11): 3073-3077.

[13]. Zhao hu., ZUO kaiwei. Implementation of FCM algorithm based on the iterative MapReduce model[J]. Computer Measurement & Control, 2016, 24 (11):240-252.

[14]. LI zhao., Li xiao., WANG chunmei. Text clustering method study based on MapReduce [J]. Computer Science, 2016,43(1):246-250.

[15]. GOU Jie., MA Zitang. Parallel SFLA-FCM clustering algorithm based on MapReduce [J]. Computer Engineering and Applications, 2016,52(1): 66-70.