Fragmentation in Distributed Database Design Based on KR Rough Clustering Technique

Van Nghia Luong[1(✉)], Van Son Le[2], and Van Ban Doan[3]

[1] Faculty of Information Technology,
Pham Van Dong University, Quang Ngai, Vietnam
nghia.itq@gmail.com, lvnghia@pdu.edu.vn
[2] Da Nang University of Education, Da Nang, Vietnam
levansupham2004@yahoo.com
[3] Institute of Information Technology,
Vietnamese Academy of Science and Technology, Hanoi, Vietnam
dvban@ioit.ac.vn

Abstract. Knowledge mining according to rough set approach is an effective method for large datasets containing many different types of data. Rough clustering, as in rough set theory, using lower approximation and upper approximation, allows objects to belong to multiple clusters in a dataset. KR Rough Clustering Technique (K-Means Rough) we propose in this paper follows k-Means primitive clustering algorithm improvement approach by combining distance, similarity with upper approximation and lower approximation. In particular, appropriate focuses will be calculated to determine whether an object will be assigned to lower approximation or upper approximation of each cluster.

Keywords: Rough set theory · Vertical fragmentation · Rough cluster
Cluster focus

1 Rationale

Rough clustering algorithms use distance measure to construct a similar matrix and each pair of objects in this matrix is assigned to the current cluster or new cluster depending on one or both objects in the pair currently being distributed [3]. With this approach, a large number of clusters will be created. It may be uncertain to ensure whether lower approximations of the clusters have the most effective overlay area of the dataset [4].

Clustering technique according to rough set theory supports clustering in two directions:

- Improve such classic clustering algorithms as k-Means, k-Medoids into rough_ k-Means (k-Means Rough), rough_k-Medoids (k-Medoids Rough), by combining distance, similarity with upper approximation and lower approximation [10].
- Support to identify the minimum number of clusters, based on the number of initial suggestion clusters provided by the user. Clusters will be clustered if approximations on the intersection clusters are non-empty [11].

P. Cong Vinh et al. (Eds.): ICCASA 2017/ICTCC 2017, LNICST 217, pp. 166–172, 2018.
https://doi.org/10.1007/978-3-319-77818-1_16

This article is organized as follows: Sect. 2 presents some related concepts of rough clustering technique. Proposed KR algorithm for vertical fragmentation in distributed data based on rough clustering technique is presented in Sect. 3. Section 4 in turn presents the experimental setup on KR and compares the experimental results with primitive k-Means. Section 5 is the conclusion.

2 Some Related Concepts

2.1 Data Discretization and Attribute Selection, Attribute Extraction According to Rough Set Approach

In the field of knowledge mining, the problem is how to process mixed data with continuous values. Many algorithms are used to discrete data such as logical reasoning methods, NAIVE algorithm, etc. However, there is no optimal algorithm. An algorithm is selected depending on the type of data to be processed. Authors in [2] outline some data discretization methods based on rough set and logical reasoning.

Attribute Selection, Attribute Extraction based on rough set [4]: Databases in practice often have many attributes. Attributes required for KPDL problem being processed are not all. Selecting the appropriate attributes for KPDL method is necessary.

2.2 Information System, Indistinguishable Relation

Definition 1. Information system [2] is a pair $SI = (U, A)$, in which $U = \{t_1, t_2, .., t_n\}$ is a finite set of objects, A is a non-empty finite set of attributes and $a : U \rightarrow V_a$ with all $a \in A$. Set V_a is called the value set of attribute a.

Definition 2. With any information system $SI = \{U, A\}$ and a non-empty set of attributes $B \subseteq A$, an *information function* B is defined as follows [2]:

$InfB = \{(a, a(x)) : a \in B\}$ with all $x \in A$.

In special case $B = A$, then set $\{InfA(x) : \mathrm{x} \in A\}$ is called *information set* A, abbreviated as $INF(A)$.

One of basic characteristics of rough set theory is to store and process data that is ambiguous, indistinguishable [3]. In information system as defined above, there can also be indistinguishable objects.

Definition 3. An indistinguishable relation, denoted by $IND_A(B)$, is defined as: $IND_A(B) = \{(x, x') \in U^2 | \forall a \in B : a(x) = a(x')\}$, in which:

- B: an attribute set of objects, $B \subseteq A$.
- x, x': any two objects belonging to U.

Then $IND_A(B)$ is an *equivalence relation* B [3].

When two objects x, x', that $(x, x') \in IND_A(B)$, then two objects x, x' is called *indistinguishable* by attributes in B. When considering a definite information system, symbol A is often omitted, and we will abbreviate it as *IND(B)* instead of $IND_A(B)$. Equivalence class containing x of *indistinguishable relation* on B is denoted by $[x]_B$.

2.3 Reference-Specific Vector and Similarity

Definition 4. Reference-specific vector VA_j of attribute A_j corresponding to reference of transactions $(q_1, q_2, .., q_m)$ is determined [12] as follows:

VA_j =

q_1	q_2	...	q_m
M_{1j}	M_{2j}	...	M_{mj}

Definition 5. Similarity measure [12] of two attributes A_k, A_l, with two reference-specific vectors corresponding to set of transactions $Q = (q_1, q_2, .., q_m)$ of:

$VA_k = (M_{1k}, M_{2k}, .., M_{mk})$ and $VA_l = (M_{1l}, M_{2l}, .., M_{ml})$, is determined by *cosine measure* as follows:

$$s(A_k, A_l) = \frac{VA_k * VA_l}{\|VA_k\| * \|VA_l\|} = \frac{\sum_{i=1}^{m} M_{ik} * M_{il}}{\sqrt{\sum_{i=1}^{m} M_{ik}^2} * \sqrt{\sum_{i=1}^{m} M_{il}^2}} \quad (1)$$

3 Proposed Vertical Fragmentation Algorithm Based on KR Rough Clustering

3.1 KR Rough Clustering Algorithm

The most common rough clustering technique [2] is derived from primitive *k-Means* clustering. The goal is to randomly generate k clusters from n objects. Assume that objects are represented by m-dimensional vectors.

Each cluster is also represented by a m-dimensional vector, which is the *focus* or *vector* for that cluster. The process starts by randomly selecting k focuses of k *clusters*. Objects are assigned to one of k clusters based on the minimum value of the distance $d\ (v,\ x)$ between the object vectors $v = \{v_1, \ldots, v_j, \ldots, v_m\}$ and cluster vectors $x = \{x_1, \ldots, x_j, \ldots, x_m\}$ with $1 \le j \le m$. Distance $d(v, x)$ given: $d(v, x) = |v - x|$, is usually the Euclidean standard [5].

The process stops when the focuses of the cluster are stable, i.e. the *focus vectors* in the previous iteration coincide with the new *cluster focus* in the current iteration. Combining rough set into *k-Means* clustering [6] requires the addition of concepts of *lower approximation* and *upper approximation*. In particular, appropriate focuses will be calculated to determine whether an object will be assigned to lower approximation or upper approximation of each cluster. *KR rough clustering algorithm uses three basic properties*:

(1). Each object belongs only to one *lower approximation.*
(2). If the object belongs to a *lower approximation*, it also belongs to a corresponding *upper approximation*.
(3). An object belongs to at least two *upper approximations* if it does not belong to any *lower approximation*.

Describe the KR rough clustering improvement algorithm in the following steps:

Step 1: Calculate the cluster focuses according to primitive *k-Means*, with modifications including *lower approximation* and *upper approximation* [9].
Step 2: Determine whether an *object* is assigned to a *lower approximation* or *upper approximation* of a *cluster*.
Step 3: Determine the distance to the previous focus.

For each object vector v, distance $d(v, x_j)$ between v and the cluster focus x_j, there are two options to identify members of an object [10]:

Option 1. Determine the nearest focus [6] by the formula:

$$d_{\min} = d(v, x_i) = \min_{1 \le j \le k} d(v, x_j) \tag{2}$$

Option 2. Check the distance with the nearest cluster focus and other focuses: $T = \{t : d(v, x_i) - d(v, x_j) <= Th_i, i \neq j\}$ [11].

- If $T \neq \varnothing$ then v belongs to *upper approximation* of two or more clusters.
- If $T = \varnothing$ then v belongs to *lower approximation* of only one cluster.

3.2 Proposed KR Rough Clustering Algorithm

KR algorithm

Input: - *D*: Set of n objects to be clustered;
- *k*: Number of clusters;
- Threshold Th_i ;

Output: Set of clusters of *D;*

Algorithm
Begin
1. Initialize randomly k focuses of the derived objects $x=\{x_1,\ldots x_k\}$;
2. **Repeat**
3. Assign objects v to the upper and lower approximations of the clusters; /* *Cluster* */
4. Calculate the distance $d(v, x_i)$, $d(v,x_j)$ between objects v with the cluster focus x_i , x_j ; /* $1 \le i,j \le k$ */
5. **If** $(d(v, x_i) - d(v, x_j) <= Th_i)$ **Then** object vector v will not belong to any lower approximations /* by nature 3*/ ;
6. **Else** $d(v, x_i)$ is minimal;
7. Update focus x_i with new focus;
8. **If** the cluster focus coincides with the previous iteration **Then** stop;
9. **Else** go back to Step 2;
10. **Until** <*Cluster focuses do not change*>

End.

3.3 Evaluation of KR Rough Clustering Algorithm

- KR rough clustering solution *is similar to KO* [12], which is capable of grouping objects in different clusters. In addition, KR also generates more clusters than number of clusters needed to describe the data depending on the measurement distance. This causes the opportunity for an object to be high when *clustering* in the same cluster [1].
- However, *KR r*ough clustering solution proceeds with a large set of data, making the solution more complex, degree of overlap among clusters to increase, so calculating the focus is slower than primitive *k-Means.*
- KR algorithm complexity is $O(t^*n^*k)$, in which t is number of iterations, n is number of objects to be clustered, and k is number of clusters. However *t, k* are usually very small compared to *n* when the dataset is large enough and contains many objects. Therefore, the complexity is usually calculated as $O(n)$. This complexity is more optimal than vertical clustering algorithm according to attribute affinity such as BEA algorithm [7] of $O(n^2)$.

4 Experimental Results of KR Rough Clustering Algorithm

We compared experimental results of vertical fragmentation according to KR rough clustering and primitive k-Means by total time cost and memory cost. Dataset installed [8] consists of 20 objects as (Table 1):

Table 1. Dataset D consists of 20 instance

@NAME = Instance 1	@NAME = Instance 6	@NAME = Instance 11	@NAME = Instance 16
5.1 3.5 1.4 0.2	4.4 2.9 1.4 0.2	5 3 2 1	20 50 52 21
@NAME = Instance 2	@NAME = Instance 7	@NAME = Instance 12	@NAME = Instance 17
4.9 3.0 1.4 0.2	4.9 3.1 1.4 0.2	15 13 12 11	10 15 52 21
@NAME = Instance 3	@NAME = Instance 8	@NAME = Instance 13	@NAME = Instance 18
4.7 3.2 1.3 0.2	5.4 3.7 1.5 0.2	30 60 52 51	21 25 25 22
@NAME = Instance 4	@NAME = Instance 9	@NAME = Instance 14	@NAME = Instance 19
4.6 3.4 1.7 0.2	4.8 3.7 1.5 0.2	50 40 42 41	11 15 35 42
@NAME = Instance 5	@NAME = Instance 10	@NAME = Instance 15	@NAME = Instance 20
5.0 3.6 1.4 0.2	4.8 3 1.4 0.1	30 50 42 31	11 25 45 45

With k-Means algorithm:

- *Experiment with* $(k = 6)$*, result as (Fig. 1):*

```
========== KMEANS - SPMF 2.09 - STATS ============
 Distance function: euclidian
 Total time ~: 8192 ms
 SSE (Sum of Squared Errors) (lower is better) : 64.80000000000001
 Max memory:0.6792984008789062 mb
 Iteration count: 4
===================================
```

Fig. 1. Clustering result by k-Means algorithm (k = 6)

With KR rough clustering algorithm:

After similar experiment with number of clusters ($k = 6$), experimental results of *vertical fragmentation according to KR rough clustering* (similar to KO [12]), Fig. 2:

```
======
Improved cluster KR =============
 Distance function: euclidian
 Total time ~: 16 ms
 SSE (Sum of Squared Errors) (lower is better) : 248.76096491228066
 Max memory:1.2878875732421875 mb
 Iteration count: 8
=======================================
```

Fig. 2. Clustering results by KR with (k = 6)

Based on above two experimental results [8], the pager compiles a comparison table between two algorithms as primitive *k-Means* and proposed *KR* algorithm according to 3 tests, corresponding to number of clusters k selected ($k = 6$; $k = 13$; $k = 15$) as (Table 2).

Table 2. Comparison of KR and k-Means clustering results

Algorithm	Number cluster k	Total time (ms)	Sum of squared errors (Min)	Max memory usage (Mb)	Frequent itemsets count
k-Means	$k = 6$	8192	64.8000	0.6793	4
	$k = 13$	2623	455.4550	1.3000	3
	$k = 15$	1689	751.6216	1.6000	3
KR improved	$k = 6$	16	248.7609	1.2878	8
	$k = 13$	15	548.8960	1.2879	8
	$k = 15$	15	548.8960	1.2879	8

5 Conclusion

In this paper, we have proposed an improvement in the vertical fragmentation problem in distributed data based on *k-Means* rough clustering technique by combining *distance and similarity with upper and lower approximations.* In particular, calculate

appropriate focuses to determine whether an object will be assigned to lower approximation or upper approximation of each cluster [11].

Experimental results using KR rough clustering technique show:

- With a small number of clusters k ($k = 6$), *k-Means* algorithm has large total time, satisfactory error average cost and memory cost. Meanwhile, KR rough clustering algorithm optimizes all three criteria.
- When increasing number of *clusters* k ($k = 13$, $k = 15$), *KR* algorithm clearly expresses optimizations on all three criteria in comparison with *k-Means* algorithm. However, error average cost of KR is high as both *upper* and *lower approximations* are to be considered during the process of updating the new focus.

Complexity KR is usually calculated as $O(n)$. This complexity is more optimal than k-Means clustering algorithm [9] as $O(t^*n^*k)$ in which t is number of iterations, k is number of clusters, and n is number of objects on the set D to be clustered.

References

1. Darabant, A.S., Darabant, L.: Clustering methods in data fragmentation. Rom. J. Inf. Sci. Technol. **14**(1), 81–97 (2011)
2. Bean, C.L., Kambhampati, C., Rajasekharan, S.: A rough set solution to a fuzzy set problem. In: Proceeding of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), World Congress in Computational Intelligence, Honolulu, Hawaii (2002)
3. Bean, C.L., Kambhampati, C.: Automonous clustering using rough set theory. Int. J. Autom. Comput. **5**(1), 90–102 (2008). ISSN 1476-8186
4. Darabant, A.S.: A new approach in fragmentation of distributed object oriented databases using clustering techniques. Studia Univ. Babes-Bolyai, Informatica **3**(2), 91–106 (2005)
5. Sinwar, D., Kaushik, R.: Study of Euclidean and Manhattan distance metrics using simple k-means clustering. IJRASET **2**(V), 270–274 (2014). ISSN 2321-9653
6. Park, H.-S., Lee, J.-S., Jun, C.-H.: A k-means-like algorithm for k-medoids clustering and its performance. Department of Industrial and Management Engineering, POSTECH, South Korea (2016)
7. Hui, M., Schewe, K.D., Kirchberg, M.: A heuristic approach to vertical fragmentation incorporating query information. In: Seventh International Baltic Conference on Databases and Information Systems, pp. 69–76. IEEE (2006)
8. http://www.philippe-fournier-viger.com/spmf/
9. Jun, S., Kobayashi, S.: Large-scale k-means clustering with user-centric privacy-preservation. Knowl. Inf. Syst. **25**(2), 253–279 (2010). ©Springer
10. Shalini, S., Singh, N., Chauhan, C.: K-means v/s K-medoids: a comparative study. In: National Conference on Recent Trends in Engineering and Technology, May 2011
11. Okuzaki, T., Hirano, S., Kobashi, S., Hata, Y., Takahashi, Y.: A rough set based clustering method by knowledge combination. IEICE Trans. Inf. Syst. **E85-D**(12), 1898–1908 (2002)
12. Luong, V.N., Nguyen, H.H.C., Le, V.S.: An improvement on fragmentation in distribution database design based on knowledge-oriented clustering techniques. Int. J. Comput. Sci. Inf. Secur. (IJCSIS)-USA **13**(5), 13–17 (2015). ©IJCSIS publication 2015 Pennsylvania