Research Method Of Clustering Of The Large Size Remote Sensing Images

Trung Nguyen Tu

Abstract: Remote sensing image segmentation is a very important stage in remote sensing image processing. In many different segmentation techniques such as KMeans, C-Means, Watersed ..., KMeans is one of the widely used algorithms for remote sensing image segmentation. However, this algorithm only considers the intensity of the pixel to lose the contextual information of the object, affecting clustering quality. The 2D-KMeans algorithm overcomes this disadvantage, but it increases the number of dimensions of each object. This results in the executing time of the algorithm is very large, especially when the large-scale remote sensing images is clustered. This paper presents the new clustering algorithm MapReduce_2D-KMeans in order to overcome the disadvantage of 2D-KMeans calculating time without reducing cluster quality.

Index Terms: Image Clustering, Remote sensing images, KMeans, 2D-KMeans, map_2D, reduce_2D, MapReduce_2D-KMeans.

1 INTRODUCTION

Image segmentation (or clustering) or remote sensing image segmentation has been studied for a long time and is a subject of concern. Remote sensing images are increasingly complex in terms of size, number of spectral channels and the level of the detail level of the image. There are many different segmentation methods like KMeans, morphology, Markov model, etc. Most methods only use the intensity of each pixel to segmentation. In [1], Balaji and colleagues presented a new method of segmenting images based on color characteristics from images with the conversion of pixels from RGB space to L*a*b* space and clustering on this space. In [5], the authors also combined fuzzy clustering algorithms and other gray level adjustment expressions to enhance the contrast of medical images. In [11], the authors used Wavelet to reduce noise for medical images. Currently, some algorithms use more contextual information in the process to reduce the complexity of segments [8]. In [13], the authors used a local approach based on Fuzzy C-Means clustering algorithm to enhance the contrast of remote sensing images. In the algorithms of KMeans family, the algorithm KMeans combines advantages: faster speed, cluster number controlling and effective clustering even with large images. Perhaps, these are the reasons why KMeans has been used widely in research and installed in remote sensing image processing softwares. However, when partitioning large remote sensing images, the convergence speed of the algorithm is still very slow. In [2], the authors proposed the algorithm CCEA to speed up the fuzzy KMeans algorithm. However, according to [7], KMeans loses the contextual characteristics (neighboring information) of each pixel when only the intensity feature is considered. Therefore, the authors proposed the 2D-KMeans algorithm with the addition of median values such as spatial parameters (local context information) to increase clustering efficiency [7]. However, this improvement doubles the data. This reduces the speed of data processing in general ... and the speed of clustering compared to the original KMeans in particular. The increasing size and complexity of images in general and remote sensing images in particular will be a challenge for traditional data processing methods. It will be more effective if the big data processing methods are applied. Currently, with the development of information technology, the Industrial Revolution 4.0 has led to the explosion of data (Big Data). Big data and its analysis play an important role in the Information Technology world with applications of Cloud Technology, Data Mining, Hadoop and MapReduce [9]. Traditional technologies only apply to structured data while big data includes both structured, semi-structured and unstructured data. Finding the method to effectively handle big data has become big challenges in the new age and there's a great need for new processing methods. MapReduce is a highly efficient distributed data processing model that has been widely used in large data processing [4]. This paper presents the new clustering algorithm MapReduce_2D-KMeans with using MapReduce model to overcome the disadvantages of 2D-KMeans calculating time without reducing cluster quality. In addition, the article also presents a formal representation of image clustering solution with MapReduce_2D-KMeans.

2 RELATED WORK

2.1 Overview of Remote sensing

According to [3], remote sensing is is a science which remotely gathers information on the Earth surface. It includes sensing and recording energy released, processing, analyzing data and applying the information after analysis. Besides, most of receiving systems and remote sensing images processing follow a seven-step procedure as shown in figure 1.

![Fig 1. Process of gathering and processing remote sensing images [3].](image)

In figure 1, A is energy source or bright source, B is radiance and atmosphere, C is interactive with destination object, D is energy gathered by sensor, E is energy transmission, reception and processing, F is interpretation and analysis, G is application. Remote sensing images have features: image channel, space resolution, spectrum resolution, radiant resolution and time resolution. There are many different types
of remote sensing images/satellites like Landsat, SPOT, MOS, IRS, IKONOS, WORLD VIEW – 2, COSMOS [10]...

2.2 Overview of MapReduce model

MapReduce is a model of parallel and distributed computing model that is proposed by google (Figure 2). It includes two basic functions: “Map” and “Reduce” which are defined by the user [4]. Through the MapReduce library, the program fragments the input data file. Machines include: master and worker. The master machine coordinates the operation of the MapReduce implementation process on the worker machines, the worker machines perform the Map and Reduce tasks with the data it receives. Data is structured in the form of key and value.

The formal representation of MapReduce model

According to [6] [12], we have the formal representation of the MapReduce model as follows:

map: (K1 k1, V1 v1) → list(K2 k2, V2 v2)  \hspace{1cm} (1)
reduce: (K2 k2, list(V2 v2)) → list(K3 k3, V3 v3) \hspace{1cm} (2)

Where:

- K1, V1 are the input key and value types of the map function; k1, v1 are the corresponding objects with the types K1, V1.
- K2, V2 are the output key and value types of map function and still are the input key and value types of reduce function; k2, v2 are the corresponding objects with the types K2, V2.
- K3, V3 are the output key and value types of the reduce function; k3, v3 are the corresponding objects with the types K3, V3.

In other words, we can see:

- If k1, v1, k2, v2 are identified, we have the input and output of map function. Commonly, with text data, k1 is offset value of a data row, v1 is the content of a data row.
- If k2, v2, k3, v3 are identified, we have the input, and output of reduce function.

The formal Representation may be rewritten only with k1, v1, k2, v2, k3, v3 as follows:

map: (k1, v1) → list(k2, v2) \hspace{1cm} (3)
reduce: (k2, list(v2)) → list(k3, v3) \hspace{1cm} (4)

2.3 The algorithm 2D-KMeans

In [7], authors proposed 2D-Kmeans algorithm. The differences between K-means and 2D-KMeans are:

- With K-means, each object x is a vector whose components are the intensities of corresponding pixel object: \( x^{NT} \). Therefore, each center is an average vector of intensities belonging to corresponding cluster: \( C^{INT} \).
- With 2D-KMeans, each object x is a vector which includes 2 kinds of components: the first component with the intensities of \( x^{NT} \) and the second component of local median \( x^{MED} \) (formula 6). Therefore, each center C is a vector which includes 2 component kinds the first components are average intensities \( C^{INT} \) and và the second components are average intensities of median vectors: \( C^{MED} \) (formula 6):

\[
\begin{align*}
C = \frac{1}{n_c}(\sum_{x \in C} x^{INT}, \sum_{x \in C} x^{MED})
\end{align*}
\]

Thus, the algorithm 2D-KMeans can be presented as follows (after the median image is created):

**Table 1: The algorithm 2D-KMeans.**

| Input: n objects \( x_i \) (each object includes \( x^{INT} \) and \( x^{MED} \)) with \( i = 1..n \) and clustering number \( c \) |
| Output: clusters \( C_j \) (each center includes \( C^{INT}_j \) and \( C^{MED}_j \)) to objective function \( E \) following is minimal: |
| \[
E = \sum_{i=1}^{n} \sum_{j=1}^{c} d^2(x, C_j)
\] |

**Step 1: Initialize the center of the clusters**

Select k objects \( C_i \) (includes \( C^{INT}_i \) and \( C^{MED}_i \)) are initial centers of k clusters (random or experience)

**Step 2: Assign closest cluster to each data point**

Calculate the distance between each object \( x_i \) (\( i = 1..n \)) (includes \( x^{INT}_i \) and \( x^{MED}_i \) ) and each center \( C_i \) (includes \( C^{INT}_i \) and \( C^{MED}_i \) ) \( vj = 1..c \). The Object belongs to cluster \( C_0 \) if the distance between center \( C_0 \) and this object is minimal:

\[
d(x, C_j) = \min d(x, C_j), j = 1..c
\]

**Step 3: Update centers**

Update center \( C_j \) (\( j = 1..c \)) (includes \( C^{INT}_j \) and \( C^{MED}_j \)) by calculating the average of all data points which belongs to that cluster. (similar to formula 6)

\[
C_j = \frac{\sum_{\text{cluster} C_j(x)}}{\text{count(cluster } C_j)}
\]

**Step 4: Repeat steps 2-3 until convergence**

Where: \( d(x, C_j) \): the distance from \( x \) to center \( C_j \)

3 PROPOSE THE ALGORITHM MAPREDUCE_2D-KMEANS

Disadvantages of the algorithm 2D-KMeans: According to [7], clustering quality of the algorithm 2D-KMeans is better than the original KMeans algorithm. However, the doubling of the dimensions of each clustered object \( x \) makes the algorithm execution speed much slower. This becomes even more ineffective for large images like remote sensing images. To overcome this limitations, in this subsection, we propose the algorithm MapReduce_2D-KMeans for clustering image data. Figure 4 is the diagram of the clustering algorithm MapReduce_2D-KMeans.

3.1 The algorithm MapReduce_2D-KMeans
(i,j, x\text{INT}_{ij}, x\text{MED}_{ij}) into the list of rows.

Each row includes: position information (row and column indices) and the list of values as vector elements representing a pixel (intensity components \(x_{ij}\) and median components \(x_{ij}\)).

The reason for the position information is restoring clustered images and performing post-clustering later... Thus, the output of the clustering, data elements must include intensity information, median and corresponding positions.

### 3.2.2. Formal representation of procedures map_2D-KMeans and reduce_2D-KMeans

**Input:** Each data element \(x_i\) is a set of: row and column indices, intensity components and median components: \((i,j, x_{ij}\text{INT}, x_{ij}\text{MED})\).

**Output:** The result after convergence is a set of: cluster index \(c\) and the list of elements belonging to cluster \(c\): \(x_{ij}\text{INT}, x_{ij}\text{MED}\).

Then, the pairs of \(k1, v1\) and \(k3, v3\) are determined as follows:

- **k1** is offset value, \(v1\) is the content of data row (corresponding to the object \(x_i\)), it means \((i,j, x_{ij}\text{INT}, x_{ij}\text{MED})\)
- **k3** is the information of new clusters after recalculation \(c_{new}(c_{new}, c_{new})\), \(v3\) is the list of sets \((i,j, x_{ij}\text{INT}, x_{ij}\text{MED})\) of the elements that belong to the cluster stored in \(k3\)

The Map function performs the assignment of data to the nearest cluster so \(k2, v2\) deducing as follows:

- **k2** that is the cluster index center_ind closest to \(x_{ij}\), \(v2\) is set \((i,j, x_{ij}\text{INT}, x_{ij}\text{MED})\)

At this time, the Map and Reduce procedures are be represented formally as follows:

**map2D:** \((\text{offset, } x_{ij}) \rightarrow \text{list(center_ind, } x_{ij})\) (10)

**reduce2D:** \((\text{center_ind, list(x}_{ij}) \rightarrow \text{list(Offset, list(x}_{ij})\) (11)

### 3.2.3. The algorithm of procedures map_2D-KMeans and reduce_2D-KMeans

Table 2 describes the algorithm for the map_2D-KMeans procedure. The purpose of the algorithm map_2D-KMeans is to find the nearest center (in the shared center set) to the input data object.

<table>
<thead>
<tr>
<th>Table 2: The algorithm of the function map_2D-KMeans.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> The shared center set lstCenter, key (k1) is offset, value (v1) is object information (x_i): info(x_i), it means ((i,j, x_{ij}\text{INT}, x_{ij}\text{MED}))</td>
</tr>
<tr>
<td><strong>Output:</strong> The pair ((k2, v2)): (k2) is cluster index which is nearest to (x_i), (v2) is set info(x_i)</td>
</tr>
<tr>
<td><strong>Step 1:</strong> Extract components of intensity and median information: (x_{ij}\text{INT}, x_{ij}\text{MED})</td>
</tr>
<tr>
<td><strong>Step 2:</strong> Initialization</td>
</tr>
<tr>
<td><strong>Step 2.1:</strong> minD = Double.MAX VALUE</td>
</tr>
<tr>
<td><strong>Step 2.2:</strong> cen_ind = -1</td>
</tr>
<tr>
<td><strong>Step 3:</strong> For (i = 0) to lstCenter.length</td>
</tr>
<tr>
<td><strong>Step 3.1:</strong> Assign (d = \text{CalD}(x_i, \text{lstCenter[i]}))</td>
</tr>
<tr>
<td><strong>Step 3.2:</strong> If (d &lt; \text{minD}) then</td>
</tr>
<tr>
<td>B3.2.1:** Assign (\text{minD} = d )</td>
</tr>
<tr>
<td>B3.2.2:** Assign (\text{cen_ind} = i)</td>
</tr>
<tr>
<td><strong>Step 5:</strong> Assign (k2 = \text{cen_ind})</td>
</tr>
<tr>
<td><strong>Step 6:</strong> Assign (v2 = v1)</td>
</tr>
<tr>
<td>Where, (\text{CalD}(x_i, \text{lstCenter[i]})) is the distance from the object (x_i) to the center lstCenter[i].</td>
</tr>
</tbody>
</table>

Table 3 describes the algorithm of the reduce_2D-KMeans procedure. The purpose of reduce_2D-KMeans algorithm is to recalculate the value of new cluster center from the the list of objects that belong to that cluster.

<table>
<thead>
<tr>
<th>Table 3: The algorithm of the function reduce_2D-KMeans.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> key is cluster index cen_ind, value is the list of objects (x_i) that belong to cluster whose index is cen_ind, it means list(info(x_i)\text{INT, MED}))</td>
</tr>
<tr>
<td><strong>Output:</strong> The pair ((k3, v3)): (k3) is the new center (c_{new}), (v3) is list((info(x_i)\text{INT, MED}))</td>
</tr>
<tr>
<td><strong>Step 1:</strong> Initialize the (c_{new}) array with the number of elements equal to the dimensions of the objects (x_i), which consists of 2 components (c_{new}\text{INT, MED})</td>
</tr>
<tr>
<td><strong>Step 2:</strong> Initialize (\text{num} = 0)</td>
</tr>
<tr>
<td><strong>Step 3:</strong> Foreach list(info(x_i))</td>
</tr>
<tr>
<td><strong>Step 3.1:</strong> Extract components of intensity and median information:</td>
</tr>
</tbody>
</table>
3.2.4. Generate the clustered image and the stage of post-clustering

From output data of the reduce 2D-KMeans function, most simply, the clustered image can be retrieved from the position information and intensity value of the cluster centers. In addition, after that, we can implement other things like data evaluation, data analysis, identification, classification, decision making, etc.

3.2 Proving that the quality of 2D-KMeans and MapReduce_2D-KMeans algorithms are the same

Clause: If the same input data set and the central set are initialized, two algorithms 2D-KMeans and MapReduce_2D-KMeans give the same clustering results.

Input: Data set \( X = \{x_1, x_2, ..., x_n\} \), the initialized center set \( C = \{c_1, c_2, ..., c_m\} \)

Proving:
- At the first loop:
  - With each element \( x_i \), considering the distance to each center in center set \( C = \{c_1, c_2, ..., c_m\} \). Suppose \( c_s \) is the nearest center with \( x_i \) such that:
    \[
    d(x_i, c_s) = \min(d(x_i, c_t)) \text{ for } k = 1 .. m
    \]  
    (12)
  - Comment (a): Because the centers are considered in sequence, \( c_s \) will be the final center that satisfies the condition of formula (12). Therefore, two centers \( c_s \) which calculated in the algorithms 2D-KMeans and MapReduce_2D-KMeans, are the same.
  - Comment (b): From comment (a), deducing that the set of \( x_i \) elements which assigned to the nearest center \( c_s \) is the same with the algorithms 2D-KMeans and MapReduce_2D-KMeans.
  - Comment (c): From comment (b), with each cluster represented by center \( c_s \), \( c_s \) new calculated by the algorithms 2D-KMeans and MapReduce_2D-KMeans are the same.
  - Comment (d): From comment (c), the output center set of the first loop is the same with the algorithms 2D-KMeans and MapReduce_2D-KMeans.
  - At the second loop: The input center set of the second loop is the output center set of the first loop, so two input center sets of the second loop, which are calculated by the algorithms 2D-KMeans and MapReduce_2D-KMeans, are the same.
  - With the same reasoning as in the first loop, the output center sets of the second loop are the same with the algorithms 2D-KMeans and MapReduce_2D-KMeans.
  - Reasoning as the second loop for each the next loop, we have the same results with the algorithms 2D-KMeans and MapReduce_2D-KMeans.
  - Thus, the clustering results after convergence are the same with the algorithms 2D-KMeans and MapReduce_2D-KMeans. That is thing which must prove.

4 EXPERIMENTS

We test the proposed algorithm MapReduce_2D-KMeans and compare to the algorithm 2D-KMeans. Data set used for experiments includes 3 types. The first, Landsat ETM+ images are taken in Hoa Binh area in 2001 (on 15/02/2001), including 11 pictures of districts and 1 picture of Hoa Binh province. The second, SPOT 4 images are about Hoa Binh and Son La areas with 21 pictures in 2003 and 14 pictures in 2008. The third, Quickbird images which are downloaded from model data on website: http://opticks.org. Because of the limited scope of the paper, the authors present experiments with different three input images shown in table 4.

In the experiment, we use the Spark tool to implement the algorithm MapReduce_2D-KMeans using the MapReduce model.

**Table 4: The input images in the experiments 1, 2 and 3.**

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Experiment 2</th>
<th>Experiment 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Image](415x389 to 475x449)</td>
<td>![Image](494x388 to 555x449)</td>
<td>![Image](533x389 to 593x449)</td>
</tr>
</tbody>
</table>

The test diagram is illustrated in Figure 4. Thus, two above algorithms have the same initialized center set.

**Fig. 4: The test diagram with the algorithms 2D-KMeans and MapReduce_2D-KMeans.**

4.1 Experiment 1

Input image is a Lansat image of Da Bac district, belong to Hoa binh province, with size 1596 x 1333. The clustered image is shown in table 4.
Table 4: The clustered images of the Lansat image.

<table>
<thead>
<tr>
<th>Cluster Number</th>
<th>2D-KMeans</th>
<th>MapReduce_2D-KMeans</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5 show statistics and compare the execution time of the algorithms 2D-KMeans vs MapReduce_2D-KMeans.

Table 5: The clustering time of Lansat image.

<table>
<thead>
<tr>
<th>Cluster Number</th>
<th>2D-KMeans</th>
<th>MapReduce_2D-KMeans</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>177275</td>
<td>100095</td>
</tr>
<tr>
<td>8</td>
<td>356641</td>
<td>103304</td>
</tr>
<tr>
<td>11</td>
<td>441790</td>
<td>206450</td>
</tr>
<tr>
<td>14</td>
<td>465280</td>
<td>247096</td>
</tr>
<tr>
<td>17</td>
<td>626141</td>
<td>267736</td>
</tr>
<tr>
<td>20</td>
<td>754192</td>
<td>291974</td>
</tr>
</tbody>
</table>

4.3 Experiment 3
Input image is a SPOT image with size 2201 x 2101. The clustered image is shown in table 8.

Table 8: The clustered images of SPOT image.

<table>
<thead>
<tr>
<th>Cluster Number</th>
<th>2D-KMeans</th>
<th>MapReduce_2D-KMeans</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7 show statistics and compare the execution time of the algorithms 2D-KMeans vs MapReduce_2D-KMeans.

Table 7: The clustered images of Quickbird image.

<table>
<thead>
<tr>
<th>Cluster Number</th>
<th>2D-KMeans</th>
<th>MapReduce_2D-KMeans</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.2 Experiment 2
Input image is a Quickbird image with size 2056 x 2065. The clustered image is shown in table 6.

Table 6: The clustered images of the Quickbird image.

<table>
<thead>
<tr>
<th>Cluster Number</th>
<th>2D-KMeans</th>
<th>MapReduce_2D-KMeans</th>
</tr>
</thead>
</table>

5 CONCLUSIONS
In this paper, the authors proposed the new image clustering algorithm MapReduce_2D-KMeans that uses the MapReduce model to improve the clustering speed of the algorithm 2D-KMeans. In addition, the article also presents formal representation and the detailed algorithm representations of the procedures map_2D-KMeans and reduce_2D-KMeans. The test results show that the algorithm MapReduce_2D-KMeans gives much better clustering time compared to the algorithm 2D-KMeans without reducing clustering quality. In the next study, we plan to apply the MapReduce model to other machine learning algorithms to be able to exploit, analyze and process big data efficiently. Acknowledgment. This work is partially supported by Department of Information Technology, Thuyloi university and Institute of Information Technology, Vietnamese Academy of Science and Technology. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES


