APPLIED GEOPHYSICS, Vol. 17, No. 1 (March 2020), P. 13—25, 13 Figures.
DOI:10.1007/s11770-020-0806-x

Adaptive multi-resolution graph-based clustering algorithm for electrofacies analysis*

Wu Hongliang[1], Wang Chen[3], Feng Zhou[1], Yuan Ye[1], Wang Hua-Feng♦[2,3], and Xu Bin-Sen[1]

Abstract: Logging facies analysis is a significant aspect of reservoir description. In particular, as a commonly used method for logging facies identification, Multi-Resolution Graph-based Clustering (MRGC) can perform depth analysis on multidimensional logging curves to predict logging facies. However, this method is very time-consuming and highly dependent on the initial parameters in the propagation process, which limits the practical application effect of the method. In this paper, an Adaptive Multi-Resolution Graph-based Clustering (AMRGC) is proposed, which is capable of both improving the efficiency of calculation process and achieving a stable propagation result. More specifically, the proposed method, 1) presents a light kernel representative index (LKRI) algorithm which is proved to need less calculation resource than those kernel selection methods in the literature by exclusively considering those "free attractor" points; 2) builds a Multi-Layer Perceptron (MLP) network with back propagation algorithm (BP) so as to avoid the uncertain results brought by uncertain parameter initializations which often happened by only using the K nearest neighbors (KNN) method. Compared with those clustering methods often used in image-based sedimentary phase analysis, such as Self Organizing Map (SOM), Dynamic Clustering (DYN) and Ascendant Hierarchical Clustering (AHC), etc., the AMRGC performs much better without the prior knowledge of data structure. Eventually, the experimental results illustrate that the proposed method also outperformed the original MRGC method on the task of clustering and propagation prediction, with a higher efficiency and stability.

Keywords: MRGC, AMRGC, MLP, logging facies analysis

Introduction

Logging facies analysis is an important prerequisite for identifying reservoir sedimentary characteristics, and the automatic clustering method is usually required in the analysis process. However, due to the high feature dimensions extracted on the dataset, many clustering algorithms are not well suited to practical situations, (i.e. log space is not equivalent to geological space, and two points that are close to each other in log space may not be similar geologically). At an earlier stage, Rogers proposed a BP neural network aiming at solving this problem point wisely, which requires few advanced statistical knowledge or strong log interpretation skills (Rogers, 1992). However, the BP neural network has

Manuscript received by the Editor September 28, 2019; revised manuscript received February 07, 2020.
*This work was sponsored by the Science and Technology Project of CNPC (No. 2018D-5010-16 and 2019D- 3808)

1. Research Institute of Petroleum Exploration & Development, PetroChina, Beijing 100083, China.
2. School of Computer Science and Technology, North China University of Technology, Beijing 100144, China.
3. College of Software, Beihang University, Beijing 100191, China.
♦Corresponding Author: Wang Hua-Feng (Email: wanghuafeng@ncut.edu.cn)

not achieved good results in the practical applications because of slow learning speed and low generalization capacity. Since then, Ye (Ye, 2000) proposed a multi-resolution graph-based clustering method (MRGC), and this algorithm has been developed based on some lithofacies analyzing software (Sutadiwirya, 2008; Pabakhsh, 2012; Khoshbakht and Mohammadnia, 2000; Pabakhsh, 2012; Nouri-Taleghani, 2015; Tian, 2016). Compared with the traditional clustering algorithms (TCAs) such as the Self Organizing Map (SOM)(Kohonen, 1990; Tian, 2016), Dynamic Clustering (DYN) (Diday, 1971; Mourot, 1993), Ascendant Hierachical Clustering (AHC) (Lance and Williams, 1967; Lukasová, 1979) and Artificial Neutral Network (ANN) (Tang, 2011), the MRGC (Ye, 2000) is proved to be a better option, because those TCAs have been thought to have following limitations: (1) need to know the number of clusters beforehand; (2) be sensitive to initial conditions and variations of parameter values; (3) not robust to the data variation in practical application (Mourot and Bousghiri, 1993).

For all clustering methods, determining the optimal number of clusters is one of the most important tasks of a clustering algorithm. For instance, MRGC is characterized by the nonparametric K-nearest neighbor, approach and a graph data representation. In practice, the MRGC is a good tool which analyzes the structure of the complex data and partition natural data groups into different shapes, sizes, and densities. However, due to the large amount of calculation of the MRGC algorithm itself, and the use of the KNN method as an important step in the MRGC propagation prediction stage, the overall process has seriously reduced the calculation efficiency of the MRGC algorithm. As mentioned earlier, MRGC needs to rely on initialized parameter values, which ultimately leads to unstable analysis results.

In summary, the BP neural network method and the MRGC method each have advantages and disadvantages, but there is a certain complementarity between the two. In this paper, inspired by MRGC and BP neural networks, we proposes a graph-based adaptive multi-resolution clustering analysis method (AMRGC). Compared with the MRGC, we mainly make two aspects of improvement in new proposed method: (1) a light kernel representative index (LKRI) is used instead of the KRI in MRGC algorithm; (2) Use the BP(MLP) instead of KNN in the prediction of the distribution of the new data stage (also named as the propagation stage). Consequently, the unstable results of the traditional K-nearest neighbor algorithm due to random initialization parameters are effectively avoided.

Graph-based adaptive multi-resolution cluster analysis method

In general, the MRGC at least consists of five steps (Ye, 2000): (1) Calculating the Neighbor Index(NI) which will help estimate the probability density function (PDF); (2) Calculating the KNN attractions based on the calculated NI values in the first step; (3) Calculating the Kernel Representative Index (KRI) which is necessary for generating a proper range in determining the number of clusters; (4) Merging those initial small groups to form final clusters by the Nearest Neighbors' Attraction Power; (5) Predicting the distribution of the new data. As shown in Figure 1, the difference between the traditional MRGC and AMRGC mainly lies in (i) calculating the Light-Kernel Representative Index (LKRI); and (ii) using LPA based on BP network to predict the distribution of new data.

Data normalization

In order to eliminate the inconsistency of the actual data dimension of the original log and the fluctuation of the value range, the acquired raw log data is usually cleaned and being normalized prior to further processing (Dodge, 2003). Practically, the raw log data is

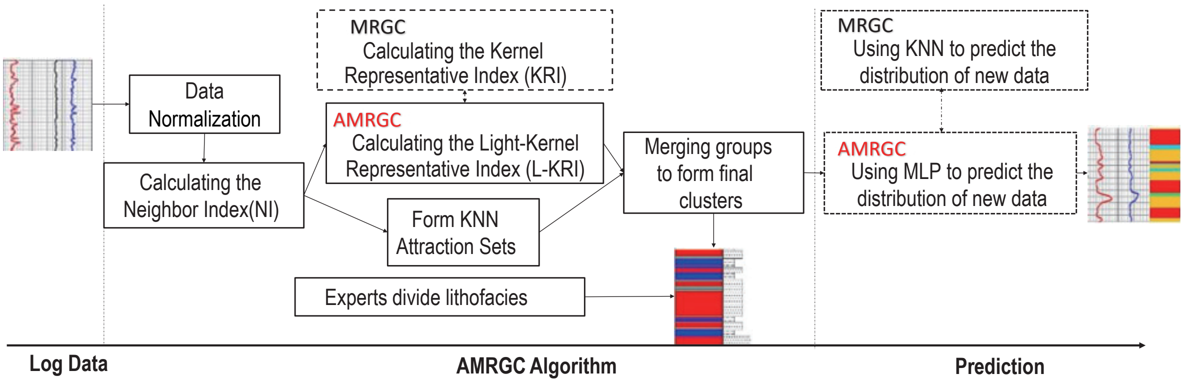

MRGC
Calculating the Kernel Representative Index (KRI)
MRGC
Using KNN to predict the distribution of new data
Data Normalization
AMRGC
Calculating the Light-Kernel Representative Index (L-KRI)
Merging groups to form final clusters
AMRGC
Using MLP to predict the distribution of new data
Calculating the Neighbor Index(NI)
Form KNN Attraction Sets
Experts divide lithofacies
Log Data
AMRGC Algorithm
Prediction

Fig.1 The pipeline of the AMRGC algorithm which also indicates the difference between the MRGC and AMRGC.

normalized to a range and converted into dimensionless numbers. As for the AMRGC algorithm, we also adopted a strategy of normalization on the original log data obtained. Suppose dataset D has n objects, that is, D={x_1, x_2, x_3, ..., x_n}, every object has the same dimension. every object x_i can be seen as a vector, where $\overrightarrow{x_i}$={x_{i1}, x_{i2}, x_{i3}, ..., x_{im}}, $1 \le i \le n$, and is the dimension for $\overrightarrow{x_i}$, x_{ij} is the j th attribute value of $\overrightarrow{x_i}$. Prior to the normalization, the mean and variance ($\vec{\mu}$ and $\overrightarrow{\sigma^2}$) of given sample over all dimensions are calculated respectively. The mean $\overline{\mu_j}$ and variance $\overline{\sigma_j^2}$ of the attribute, as well as the mean $\overline{\mu}$ and variance σ^2 of all the samples, are calculated as equations (1) to (7), which is the data normalization process in Figure 1.

$$\overline{\mu_j} = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \quad (j = 1, 2, \cdots, m), \tag{1}$$

$$\overline{\sigma_j^2} = \frac{1}{n-1}\sum_{i=1}^{n} (x_{ij} - \overline{\mu_j})^2 \quad (j = 1, 2, \cdots, m), \tag{2}$$

$$\overline{\vec{\mu}} = (\overline{\mu_1}, \overline{\mu_2}, \cdots, \overline{\mu_m}) \quad (j = 1, 2, \cdots, m), \tag{3}$$

$$\overline{\overline{\sigma^2}} = (\overline{\sigma_1^2}, \overline{\sigma_2^2}, \cdots, \overline{\sigma_m^2}) \quad (j = 1, 2, \cdots, m), \tag{4}$$

where $\overrightarrow{x_i}'$ represents the vector of $\overrightarrow{x_i}$ after the normalization, such that,

$$x_{ij}' = \frac{x_{ij} - \overline{\mu_j}}{\sqrt{\overline{\sigma_j^2}}} \quad (j = 1, 2, \cdots, m), \tag{5}$$

$$\overrightarrow{x_i}' = (x_{i1}', x_{i2}', \cdots, x_{ij}', \cdots, x_{im}') \quad (j = 1, 2, \cdots, m), \tag{6}$$

Thus, we have the Euclidean distance between any given two data points (x_i, x_j) as:

$$D(\overrightarrow{x_i}, \overrightarrow{x_j}) = \sqrt{\sum_{j=1}^{m} (x_{ij}' - x_{jj}')^2}, \tag{7}$$

Next, we define a matrix $U=(u_{ij})_{n\times n}$, and $U[i][j]=u_{ij}$ which is the index value of the j_{th} nearest neighbor(NN) of measurement of vector $\overrightarrow{x_i}$ based on the calculation of $D(\overrightarrow{x_i}, \overrightarrow{x_j})$, i.e.$\overrightarrow{x_j}$'s j_{th} nearest neighbor is $\overrightarrow{x_{u_{ij}}}$. For convenience, $\overrightarrow{x_i}$ is considered to be the n_{th} nearest neighbor of itself. Once U is determined, a sorting algorithm can be applied to identify the rank matrix $W=(w_{ij})_{n\times n}$ in which w_{ij} means $W_{[i][j]}$ and W indicates that how much $\overrightarrow{x_i}$ is relevant to its' j_{th} nearest neighbor. Meantime, a companion matrix $V=(v_{ij})_{n\times n}$ is initialized with $v_j=j$ (j=1, 2, ..., n), which indicates that $\overrightarrow{x_j}$ is the $v_{ij}th$ nearest neighbor of $\overrightarrow{x_i}$. According to the relationship described above, finally, $W[i][j]=V[U[i][j]][i]$.

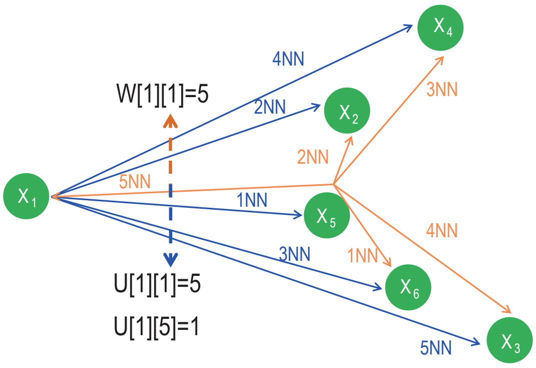


Fig.2 Illustration of the relationship among *U*, *V*, and *W*.

As shown in Figure 2, 1NN of x_1 is x_5, so we can get U[1][1]=5, which means 1NN of x_1 is x_5, V[1][5]=1, which means x_5 is 1NN of x_1, and x_1 is x_5's 5NN, so there is W[1][1]=5, which means that x_1 is 5NN of its 1NN. Among them, $W[i][j]$ can be calculated from the matrix U, V, and further we have $W[i][j]$ according to $V[U[i][j]][i]$.

Calculation of the neighbor index(NI)

Let the measurement of a given point x be an element of a set of measurement points $D = \{x_1, x_2, x_3, ..., x_n\}$, and let point y be measurement point *x's th* Nearest Neighbor (NN) in the set of measurement points S, $n \le K$. The "limited rank" τ_n of measurement point x with respect to its nth NN(y) is defined to be:

$$\tau_c(\overrightarrow{x_i}) = \exp(-m/\alpha), \tag{8}$$

where α is greater than zero and $c \le n$-1. It is noted that α is insensitive to the size of the data set and may be initialized once for all data set. In practice, α will be set to 10.

In fact, in this paper, the limited rank τ_i is defined for calculating x_i's K nearest neighbors. The sum of the limited ranks for each point x is expressed as,

$$s(\overrightarrow{x_i}) = \sum_{i=1}^{n-1} \tau_c(\overrightarrow{x_i}), \tag{9}$$

Secondly, the smallest value S_{min} and largest value Smax of rank sums are calculated by

$$S_{\min}(\vec{x}) = Min\{s(\overrightarrow{x_i})\}, (i = 1, 2, \cdots, n-1), \tag{10}$$

$$S_{\max}(\vec{x}) = Max\{s(\overrightarrow{x_i})\}, (i = 1, 2, \cdots, n-1), \tag{11}$$

Thus, the ***NI*** calculation formula can be expressed as,

$$NI(\vec{x_i}) = \frac{s(\vec{x_i}) - S_{\min}(\vec{x})}{S_{\max}(\vec{x}) - S_{\min}(\vec{x})}, (i = 1, 2, \cdots, n-1). \qquad (12)$$

The K nearest neighbors' attraction calculation method

After calculating the NI of all objects, we can use a multidimensional KNN point-to-point attraction algorithm. And it will help us calculate adherence points' attraction for every measurement of point and determine the small data group center. $Attr_{\vec{x_i}}(\vec{x_j})$ is $\vec{x_i}$'s K_{th} nearest neighbor $\vec{x_j}$'s attraction to $\vec{x_i}$, which can be calculated by,

$$Attr_{\vec{x_i}}(\vec{x_j}) = NI(\vec{x_j}) V_{\vec{x_i}}(\vec{x_j}) - NI(\vec{x_i}). \qquad (13)$$

It is acknowledged that a better data clustering results could be obtained even with a larger amount of data and a higher data dimension when the *K* value is ranged from 4 to 12 (Hastie, 2009). Meantime, it is also realized that the probability of misclassification with *K* value in this interval could be minimized. In view of this reason, the *K* value is configured by 4 for the following experiments. The adherence function $V_{\vec{x_i}}(\vec{x_j})$ is defined as ,

$$V_{\vec{x_i}}(\vec{x_j}) = \begin{cases} 1 & (\text{if } \vec{x_j} \text{ is one of } K \text{ nearest neighbor points of } \vec{x_i}), \\ 0 & \text{otherwise.} \end{cases} \qquad (14)$$

When $\vec{x_i}$ belongs to the maximum of $Attr_{\vec{x_i}}(\vec{x_j})$, $\vec{x_i}$ is attracted to the nearest neighbor $\vec{x_j}$. If all $\vec{x_j}$in the *K* nearest neighbor of $\vec{x_i}$ do not attract $\vec{x_i}$, and this implies that $\vec{x_i}$ is not adhered to any other points. We calculate the maximum of $Attr_{\vec{x_i}}(\vec{x_j})$ as the $Attr_{\vec{x_i}}$, that is,

$$Attr_{\vec{x_i}} = Max\{Attr_{\vec{x_i}}(\vec{x_j})\}. \qquad (15)$$

It should be noted that all objects are classified based on the three basic situations as follows,

(1) For *j*=1, 2,...,*K*(*j*≠*i*), the $Attr_{\vec{x_i}}(\vec{x_j})$ always true, which means $\vec{x_i}$ is not attracted by any points, even the *K* nearest neighbor, $\vec{x_i}$ is a "Free Attractor", which is similar to a kernel point of a local maximum of the probability distribution function (PDF).

(2) If $\vec{x_i}$ is directed to another point and at the same time attracts other points, $\vec{x_i}$ is a "Related Attractor". In this case, $\vec{x_i}$ is on the slope.

(3) If $\vec{x_i}$ is directed to another point but does not attract any other point. $\vec{x_i}$ is the "Pending Related" , which means $\vec{x_i}$ is most likely in the valley zone.

Also, at very start, we need to determine all objects' situations, then the attractions sets are formed. We have the attraction sets $S=\{S_i\}(i=1, 2, ..., N)$ (*N* is the number of kernel points) which is the initial group which can help cluster the delta.

Calculating the light kernel representative index (LKRI)

In order to further determine the cluster core, we need to calculate the KRI of each kernel point $\vec{x_i}$ (*i*=1, 2, ..., *n*), we set 1-st nearest neighbor with $\vec{x_j}$, whose $NI(\vec{x_j})$ > $NI(\vec{x_i})$, and $H(\vec{x_i}, \vec{x_j})$ = h , which means that $\vec{x_j}$ is the h_{th}NN of $\vec{x_i}$. Thus, the KRI of $\vec{x_i}$ can be calculated as follows,

$$KRI(\vec{x_i}) = NI(\vec{x_i})^a H(\vec{x_i}, \vec{x_j})^b D(\vec{x_i}, \vec{x_j})^c, \qquad (16)$$

where *a*, *b*, *c* are used to weight each corresponding function $NI(\vec{x_j})$, $H(\vec{x_i}, \vec{x_j})$, and $H(\vec{x_i}, \vec{x_j})$, $D(\vec{x_i}, \vec{x_j})$ under condition that $a = b = c = 1$.

Compared to MRGC, AMRGC only needs to calculate the KRI value according to the equation (16) based on the previously obtained free attractor nodes. In contrast, the MRGC will have to do the calculation on all the data points. Therefore, we call the new algorithm LKRI, which means that AMRGC is with a less computational load. In the experimental section, the benefits of replacing KRI with LKRI will be discussed in detail.

On the other hand, the reason why this algorithm can obtain suitable kernel point candidates is that by analyzing NI, as we know, the points with higher NI values are more likely to become the center of clustering. Actually, an NI value can be analogously attractive, and a higher NI value means that the selected point is much more attractive to other data points, the more likely it will be near the center of the cluster.

For example, as shown in Figure 3a, the data point with an NI value 0.9, i.e. the center point of figure 3a, and the rest of the points nearby whose NI values are mostly less than 0.9 formed an attraction set. Since x_1 is not attached to other data points, it is a free attractor. More likely, the same thing happened with the data point whose NI value is 0.95, i.e. the center point of figure 3b. As a result, the distance of x_2 from x_1 is larger than that of most points in x_1's attraction dataset, that is, only x_2 with an NI value of 0.95 satisfies the KRI calculation condition. Note, the method for calculating the KRI of x_1 point can refer to equation (16).

For a given point with a high NI value, it will be a difficult task to find another data point with an even higher NI in the vicinity. This also means a point whose NI value is larger than that of the known point tends to be a long way from the known point. That is, the free attractor nodes chosen by thresholding NI values of given points tend to have a high probability of being kernel point candidates. Later experiments also show that, compared with the results of previous kernel selection algorithms, AMRGC is good at selecting the cluster center points. Not only is its' clustering result quite consistent with that of the traditional MRGC when there are fewer center points needed to be determined, but also it outperformed the traditional MRGC when many center points occurred. In short, the new algorithm can greatly reduce the amount of process computation under the premise of ensuring that the correct cluster center is chosen.

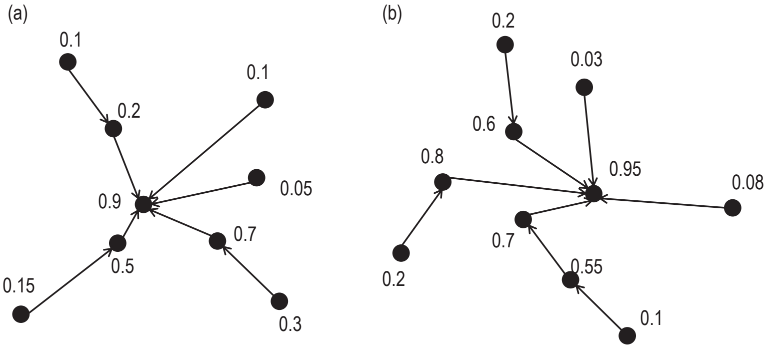


Fig.3 Attraction schematic diagram.

Finally, we arrange the LKRI in descending order, which will help us pick the final kernel points. And the merging stage will merge the obtained attraction sets and finally output the clustering result (Sutadiwirya, 2008). The details of the algorithm are described in Appendix.

LPA and MLP in AMRGC propagation stage (LPA+MLP)

In order to take full advantage of the data labels obtained by the AMRGC algorithm, we no longer use a simple KNN for propagation, but a KNN-based label propagation algorithm (LPA).

Using LPA based KNN to prediction

The KNN is a classical supervised learning method, for a new object $\overrightarrow{x_p}$, we use k training objects $\overrightarrow{x_i}$ (i=1, 2, ..., k) closest in distance d ($d \geq 0$) to $\overrightarrow{x_p}$, then classify $\overrightarrow{x_p}$ by using majority vote among the k neighbors. Usually, after data normalization, the distance d is calculated by two-norm based on different dimensions attributes, it can also be called "Euclidean distance". As previously claimed, the distance d is defined as,

$$\begin{aligned} d = D(\overrightarrow{x_p}, \overrightarrow{x_i}) &= \| \overrightarrow{x_i} - \overrightarrow{x_p} \|^2 \\ &= \sqrt{\textstyle\sum_{j=1}^{m} (x_{ij}{}' - x_{pj}{}')^2}, \qquad i = 1,\ 2, \cdots,\ k. \end{aligned} \tag{17}$$

Inspired by Zhu's method (Zhu, 2002), the main idea of our proposed LPA algorithm is to utilize the tag information of tagged nodes to predict the tag information of unlabeled nodes. In order to build a relational complete graph model based on given samples, we need to construct a graph structure first. The nodes of each graph are those known data points that contain the labeled data as well as the data to be labeled, and the edges between the nodes represent the similarities between them. In each graph, the labels of nodes are transmitted to other nodes according to their similarities. The greater the similarity of nodes, the easier the labels isare to spread. There are many ways to build this map such as RBF and KNN and so on. KNN graph was exploited in this study. In view of that, only will the K-neighbor weight of each node for this KNN graph be kept, while the rest will be set to 0. Hence, such a KNN graph can be regarded as a sparse similarity matrix. Suppose that all these graphs were fully connected, then the weight between any given node x_i and node x_j could be calculated by equation (18), and α, an $N \times N$ probability transfer matrix P is defined, where P_{ij} denotes the probability of node x_i transferring to node x_j. According to Zhu's definition (Zhu, 2002), we have,

$$\omega_{ij} = \exp(-\frac{\| x_i - x_j \|^2}{\alpha^2}), \tag{18}$$

$$P_{ij} = P(i \to j) = \frac{\omega_{ij}}{\sum_{k=1}^{K} \omega_{ik}}. \tag{19}$$

MLP to make new data prediction

As shown in Figure 4a, for inputted object $\vec{x} = (x_1, x_2, ..., x_i)$, its each dimension attribute is an input unit in input layer. The output vector is $\overrightarrow{y_L} = (y_1, y_2, \cdots, y_n)$ (l is the number of classes label which is predestinated for AMRGC). And $\vec{t} = (t_1, t_2, \cdots, t_l)$ is the target vector value.

At each layer, we first compute the total input z for each unit, which is a weighted sum of the outputs of the units in the layer below. Then a non-linear function $f(\cdot)$ is applied to z to get the output of the unit. For simplicity, the bias terms isare omitted. The non-linear function used in neural networks is a rectified linear unit (*ReLU*) $f(z) = Max(0, z)$. The output y_l is classes' label vector which is predestinated for AMRGC. Such that y_l will be expressed as

$$y_l = f(z_l) = f(\sum \omega_{ml} y_m)$$

$$= f(\sum \omega_{ml} f(z_m)) = f(\sum \omega_{ml} f(\sum \omega_{jm} y_j))$$

$$= f(\sum \omega_{ml} f(\sum \omega_{jm} f(z_j))) = f(\sum \omega_{ml} f(\sum \omega_{jm} f(\sum \omega_{ij} x_i))), \quad (20)$$

In order to confirm the values of ω_{ml}, ω_{jm}, ω_{ij} in equation (20), the backward propagation algorithm is used. In Figure 4b, the equations are used for computing the backward propagation. The output error between output and target value can be describe as equation (21), which we also call it as "Loss function" and is shown as following:

$$E = \frac{1}{2} \| \vec{t} - \overrightarrow{y_L} \|_2^2 = \frac{1}{2} \sum_{l=1}^{l} (t_l - f(\sum \omega_{ml} f(\sum \omega_{jm} f(\sum \omega_{ij} x_i))))^2,$$
$$(m \in H_{2,} j \in H_1, i \in input, l \in output). \quad (21)$$

where the ω_{ml}, ω_{jm} and ω_{ij} can be referred to Figure 4. Once the $\partial E/\partial z_m$ is known, the error-derivative for the weight ω_{jm} on the connection from unit j in the layer below is perceived as $y_j \cdot \partial E/\partial z_m$. Then we can determine the relationship between E and input vectors by using the chain rule of derivatives.

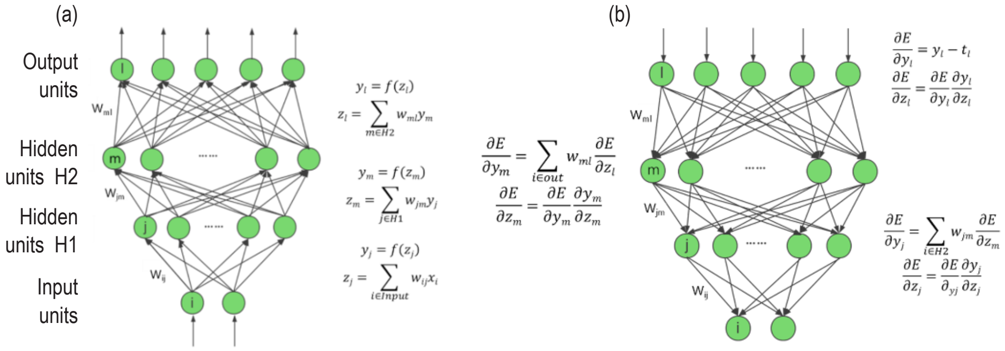

(a)
Output units
Hidden units H2
Hidden units H1
Input units
l
m
j
i
W_{ml}
W_{jm}
W_{ij}
......
$y_l = f(z_l)$
$z_l = \sum_{m \in H2} w_{ml} y_m$
$y_m = f(z_m)$
$z_m = \sum_{j \in H1} w_{jm} y_j$
$y_j = f(z_j)$
$z_j = \sum_{i \in Input} w_{ij} x_i$
(b)
$\frac{\partial E}{\partial y_l} = y_l - t_l$
$\frac{\partial E}{\partial z_l} = \frac{\partial E}{\partial y_l} \frac{\partial y_l}{\partial z_l}$
$\frac{\partial E}{\partial y_m} = \sum_{i \in out} w_{ml} \frac{\partial E}{\partial z_l}$
$\frac{\partial E}{\partial z_m} = \frac{\partial E}{\partial y_m} \frac{\partial y_m}{\partial z_m}$
$\frac{\partial E}{\partial y_j} = \sum_{i \in H2} w_{jm} \frac{\partial E}{\partial z_m}$
$\frac{\partial E}{\partial z_j} = \frac{\partial E}{\partial_{yj}} \frac{\partial y_j}{\partial z_j}$

Fig. 4 Propagation lgorithm workflow. (a) The four-layer forward propagation perceptron. (b) The four-layer backward propagation perceptron.

When we have new objects to train, we can use gradient descent algorithm to determine the unknown weight parameters and predict the test objects by corresponding weight parameters. In recent years, the neural network technology has made a breakthrough progress (Hinton, 2006) and received great attention. As aspect of our contribution, the new proposed method combines the neural network with the clustering method. Specifically, this method extracts features from the training data by neural network, and then uses the trained model to predict the propagation of the data without tags. The specific network parameters selected in the process will be described in detail in the experimental section in Section 3.

Experimental design and result

Clustering results of AMRGC and MRGC in comparison

As for the calculation process of attraction sets, the parameter K in this experiment was set to 4, and the merging parameter K' was set 8. In the experiment, we randomly selected a group of acquired logging data, which consists of 840 data points. After descending the LKRI, the result is shown in Figure 5. These points are arranged in descending order according to the LKRI value, and the free attraction points in the front more likely become kernel points, which are retained

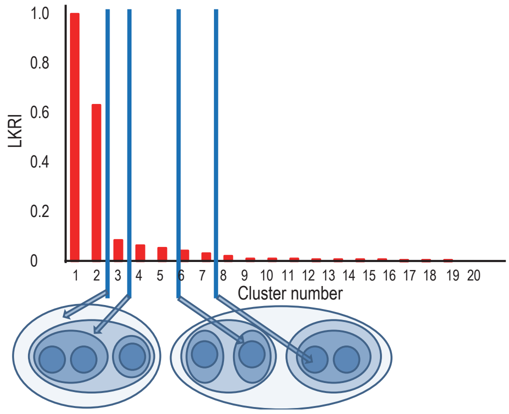


Fig. 5 Kernel points LKRI and cluster number choosing.

in the subsequent fusion process. The dark blue circle area indicates the number of corresponding point sets obtained by selecting different numbers of free attraction points. Wherein, each ellipse or circle represents a set of points formed by clustering.

In this experiment, five-dimensional data such as natural gamma-ray(GR), lithology density(DEN), compensated neutron (CNL), sonic interval transit time (AC), and deep resistivity (RD) were used for clustering experiments. Finally, the two-dimensional data of GR and CNL are drawn and displayed. In the figure, the abscissa axis is GR and the ordinate axis is CNL. It can be seen from these experiments that the K value has little effect on the clustering results and the clustering results are relatively stable, which also indicates the robustness of AMGRC. The effect of K value on clustering results is shown in Figure 6. According to the clustering results shown in Figure 6, it can be concluded that the initial value of the K value has little effect on the final clustering results. The colored balls in Figure 6 represent different clusters acquired after data points are processed by the clustering algorithm, and the colors represent different categories. The lithofacies label is represented by each color needs to be obtained after further processing.

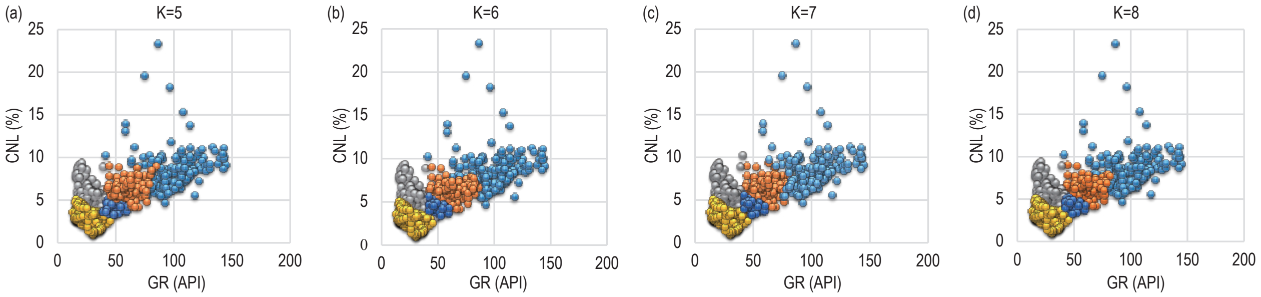


Fig. 6 Effects of different K values on clustering results.

Through the LKRI value, we can obtain multiple important indicators from those data points, which corresponds to the best cluster number at different resolutions. After obtaining the optimal number of clustering centers, we can still manually adjust the clusters. As for computing resource consumption, the AMRGC algorithm only needs to calculate the KRI value of a free attractor node (in this case, about 40 data points), while the traditional MRGC algorithm must calculate the KRI value of all data points (in this case Case, 840 data points). This indicates that the computational cost of the new KRI calculation method (LKRI) is reduced by 1/16 compared to what MRGC usually requires. In fact, as the amount of data increases, the proportion of all data points and free attractor nodes increases rapidly (see Figure 7).

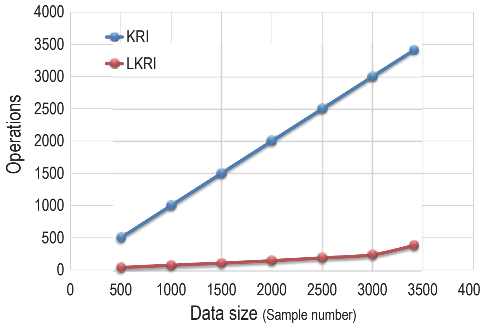


Fig. 7 The comparison of the calculation consumption between AMRGC and MRGC.

Therefore, the AMRGC algorithm can greatly reduce the amount of calculation. The experiment results depict that the AMRGC algorithm can still produce valid results although the amount of calculations decreases dramatically . In other words, the AMRGC algorithm can obtain the same kernel points as the MRGC algorithm, and can even obtain better clustering results than the MRGC algorithm. As shown in Figure 8, it illustrates different clustering results with different numbers of clusters: 3, 5, 7, and 10 (both the result of traditional MRGC and our proposed AMRGC algorithm are listed). According to the results, we can observe that when the number of clusters is small, LKRI and KRI will choose the same kernel and produce the same clustering results. Only when the number of clusters is large (for example, we found that it is 10 in this experiment), LKRI will select different kernel points from that the KRI select, and certainly the clustering results will be different. Also, the experimental results show that the kernel points selected by the LKRI method can obtain the same or even better clustering results.

In general, log interpretation experts understand the surface truth of lithofacies by referring to the information provided by coring wells. Based on coring well data, the number of clusters was determined. According to the recommendations of experts, in the following experiments, we used the cluster clustering model to classify the lithofacies. In addition, in order to compare the clustering effect with other algorithms, we also set the number of clusters of the SOM, AHC, and DYN algorithms to 5 (all of these methods need to know the number of clusters in advance). Figures 9 and 10 show the final clustering performance of each method.

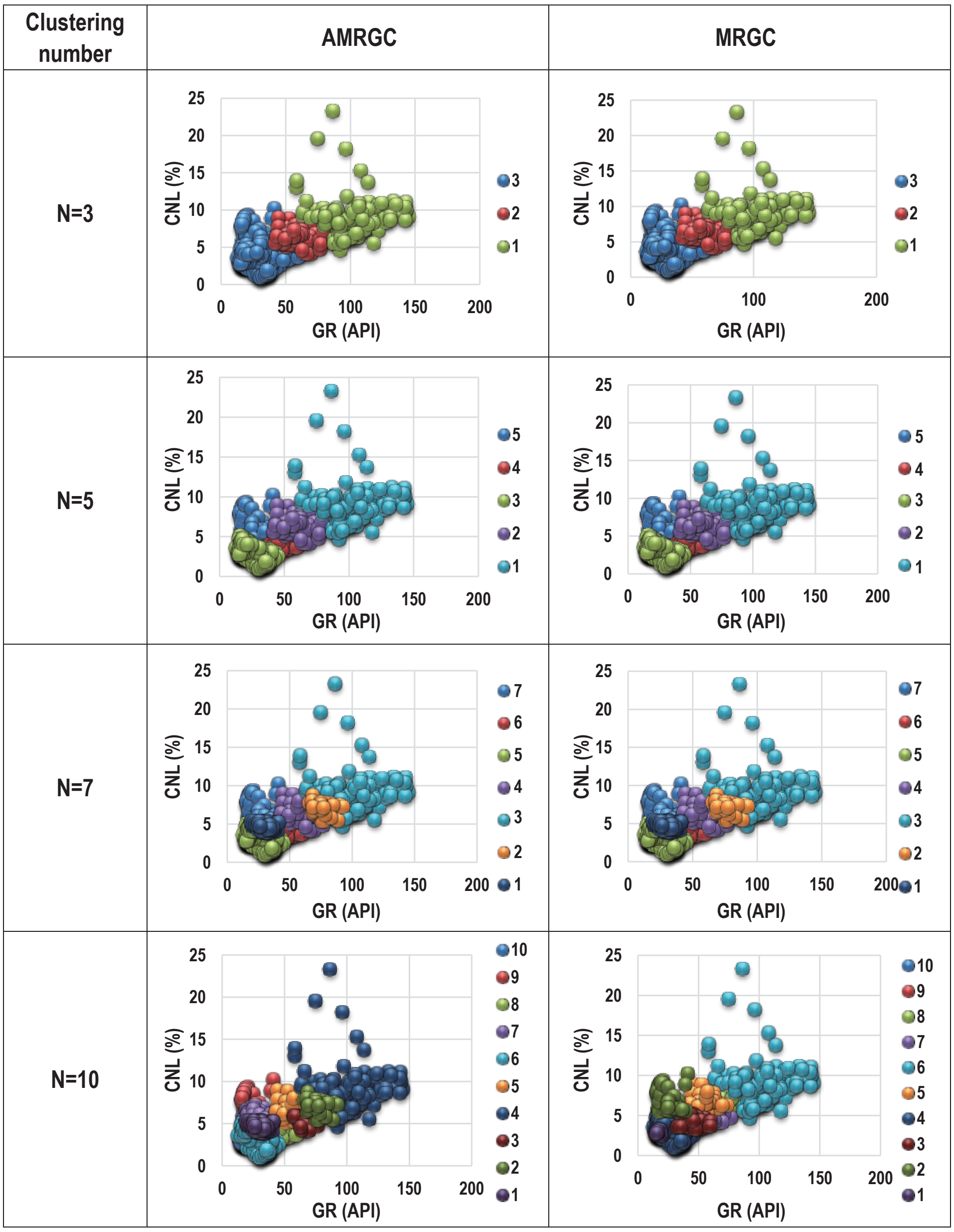


Fig. 8 The clustering results by using AMRGC and MRGC.

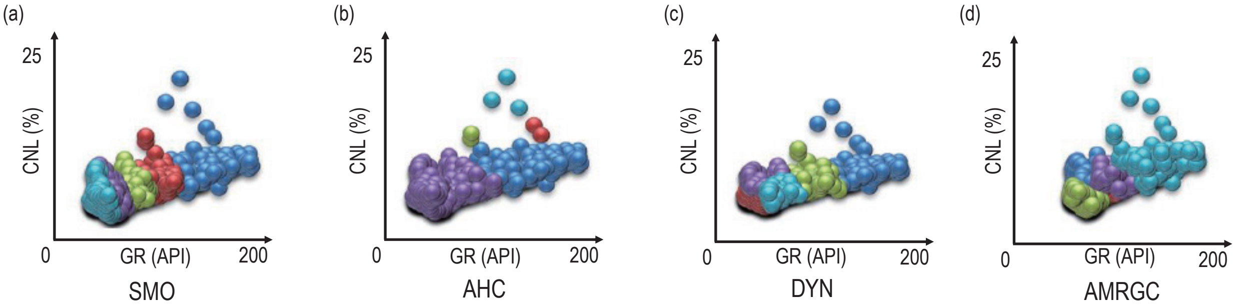


Fig. 9 the illustrations of clustering results produced by different methods.

As shown in Figure 10, it intuitively proves that the AMRGC algorithm performs better than other methods (such as SOM, AHC, DYN, etc.) in well facies prediction. The legends in Figure 8 and Figure10 are further illustrated in Figure 11. The point number column in Fig. 11 represents the number of sample points for each class in clustering result, one point is one target, and the target has two dimensional attribute values of GR and CNL.

In order to compare AMRGC quantitatively with other unsupervised clustering algorithms, we divided each well section of the experiment into 10 small

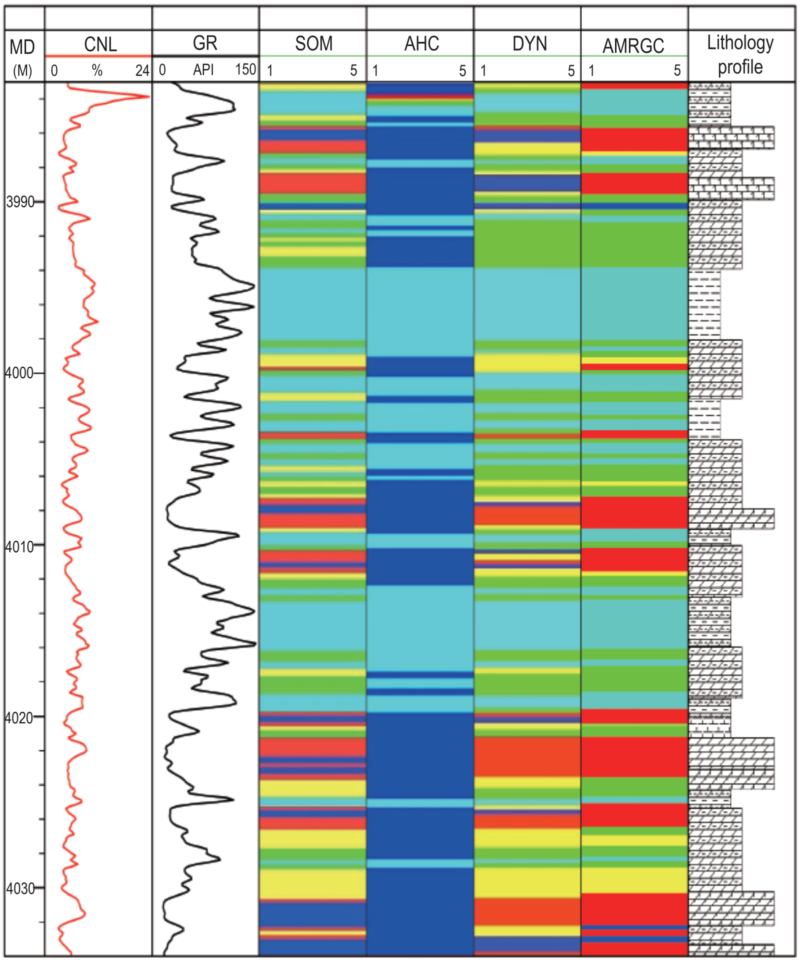


Fig. 10 The clustering results correspond to the special classification of lithofacies.

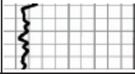
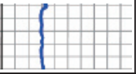
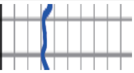
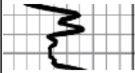
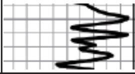
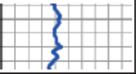
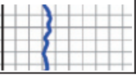
Facies Type		Classes id in AMRGC		GR(API) 0 150	CNL(%) -15 45	Lithology	Nums of Points	GR average (API)	CNL average(%)
Facies 1	🟥	5	●			Dolomite limestone	290	26.0	5.1
Facies 2	🟨	4	●			Argillaceous dolomite	58	46.0	4.2
Facies 3	🟩	3	●			Containing dolomite	130	63.6	6.4
Facies 4	🟦	2	●			Mudstone	132	103.0	9.0
Facise 5	🟦	1	●			Limy dolomite	231	25.7	3.0

Fig. 11 The correspondence of the facies types and the AMRGC clustering results.

sections to calculate the coincidence rate separately. The coincidence rate is calculated by comparing the sum of each actual length in the prediction result with the total length of the actual lithofacies. The final degree of overall consistency is measured by an average. From Figure 10 and Table 1, it can be seen that the AHC algorithm is easy to confuse the two types of lithology, which seriously affects the coincidence rate. For the AMRGC algorithm, it has a higher coincidence rate, so it has better discrimination ability for most lithologies.

Tab. 1 the conformity ratios given by methods

Algorithm	Conformity Ratio
AHC	30.2%
DYN	77.5%
SOM	81.2%
AMRGC	84.2%

Propagation prediction

In the experiment of propagation prediction, a 4-layer perceptual neural network was used. The specific hyperparameters used in the experiment are shown in Table 2. The specific steps are as follows: first, randomly select 80% of the data points from 840 data points and apply the AMRGC algorithm to obtain the clustering result; then, use the obtained output as input for training MLP; finally, apply the trained model to the remaining 20% of the data points are predicted. Experimental results show that when the accuracy of the training data reaches 87.5%, MLP can obtain better and more stable prediction ability than LPA. The results are shown in Figure 13. Next, for comparison, we use 20% of the data set to train the MLP, and use the generated training model to predict the labels for the remaining 80% of the data points. However, because the amount of training data at this time is much smaller than the amount of test data, it cannot have a good generalization ability for the test data. In this case, using semi-supervised LPA generally yields better results than using multi-layer perceptual neural networks with supervised learning. However, without loss of generality, that is, to ensure a sufficient amount of data for training, it is not difficult

Table 2 The MLP Architecture and parameters

Parameters	Value
Neuron in Input layers	2
Neuron in Output layers	5
Hidden layers	2
Neuron in each hidden layer	8,10
Iterations	500
Activation function in hidden and output layer	Re-LU
Learning rate base	0.003
Learning rate decay	0.8
Decay step	5000
Regularization rate	0.0001
Number of samples selected for one training iteration	8
Probability of each network connection being kept	0.75

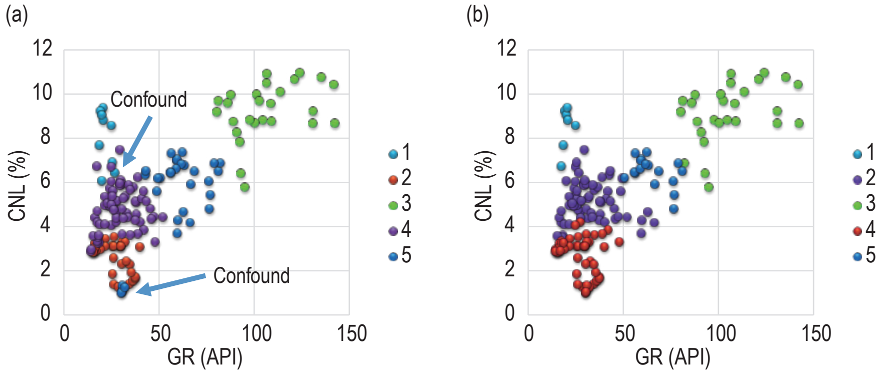


Fig. 12 Prediction results comparison based on Label propagation and MLP propagation.
(a) LPA based on KNN. (b) MLP based on BP.

to find that a multi-layered perceptual neural network that has iterated 500 times has better propagation performance than LPA. The corresponding comparison results are shown in Figure 14.

In addition, we also compared the new method with the traditional unsupervised clustering method. Specifically, we compare the performance of AMRG with other traditional clustering algorithms based on various indicators(as shown in Table 3). Specific indicators include whether prior knowledge is needed, processing efficiency in large-scale data, and characteristics of the propagation stage.

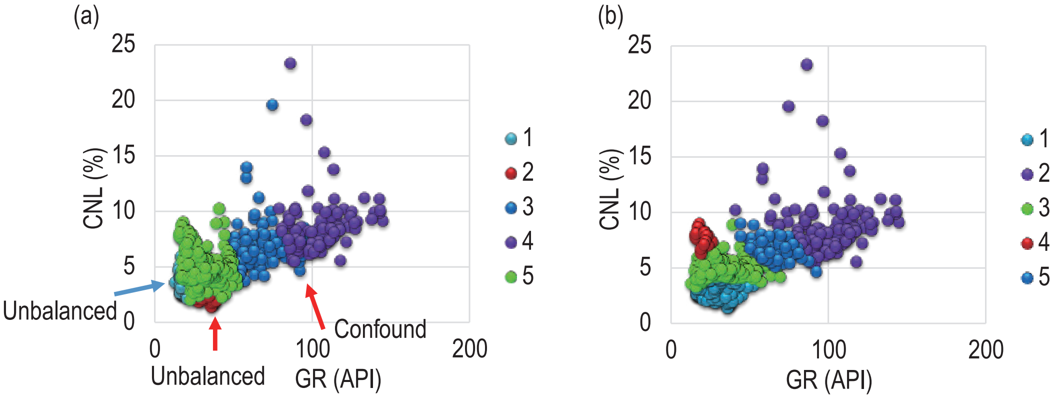


Fig. 13 Results for a crossing validation. (a) LPA based on KNN. (b) MLP based on BP.

Tab. 3 AMRGC compared with other unsupervised clustering algorithms

Clustering Methods	AMRGC	MRGC	SOM	DYN	AHC
Requires no prior knowledge of data structure	✓	✓	-	-	-
No Operator Bias	✓	✓	-	-	-
Faster against large datasets	✓	✓	-	✓	-
More magical propagation stage	✓	-	-	-	-

Conclusions

This study addresses the challenges of traditional log facies cluster analysis algorithms that require manually specifying the number of clustering kernels and the sensitivity of clustering results to initialization values. The new proposed method (AMRGC) exposit the Light Kernel Representative Index (LKRI) to determine the number of clustering kernels, and uses a combination of neural networks and clustering methods to predict the data propagation. In general, it provides a useful exploration of a combination of supervised and unsupervised methods for logging facies cluster analysis tasks. From the perspective of method implementation, this method is a multi-dimensional point pattern recognition method based on non-parametric KNN and graph clustering, which takes into account the advantages of the MRGC algorithm (such as making full use of high-dimensional information in geological data for cluster analysis), While avoiding the shortcomings of the MRGC method (such as computational efficiency). The experimental results show that the newly proposed method performs well in the task of clustering high-dimensional data or data with complex structures. This algorithm not only improves the operation efficiency of the traditional MRGC algorithm, but also can better predict the unlabeled data. In addition, without prior knowledge of the data, the new algorithm performs significantly better than other clustering methods commonly used in the industry, such as self-organizing mapping (SOM). Therefore, the new method will be helpful for the clustering analysis of logging facies with large-scale logging data as input. With the rapid development of the current deep learning field, this method still needs to explore new data propagation methods in subsequent research work, including the use of deep learning technology, in order to better meet the continuous data processing efficiency requirements.

References

Abrar, B., 2011, Integration of log data analysis and facies core to define electrofacies using multi resolution graph-based clustering: Iranlan Journal of Chemistry & Chemical Engineering- International English Edition, **30**(3),1−7.

Diday, E., 1971, Une nouvelle méthode en classification automatique et reconnaissance des formes la méthode des nuées dynamiques: Revue de statistique appliquée, **19**(2), 19−33.

Dodge, Y., and Commenges, D. 2006, The Oxford dictionary of statistical terms: Oxford University Press, England, 119-370.

Hastie, T., Tibshirani, R., & Friedman, J., 2009, The elements of statistical learning: data mining, inference, and prediction: Springer Science & Business Media. 467−468.

Hinton, G. E., Salakhutdinov, R. 2006, Reducing the dimensionality of data with neural networks: Science, **313**(5786), 504−507.

Khoshbakht, F., Mohammadnia, M., 2010, Assessment of clustering methods for predicting permeability in a heterogeneous carbonate reservoir: Journal of Petroleum ence & Technology, 2(2), 50−57.

Kohonen, T., 1990, The self-organizing map: Proceedings of the IEEE, **78**(9), 1464−1480.

Lance, G. N., and Williams, W. T., 1967, A general theory of classificatory sorting strategies Hierarchical systems: Computer Journal, **9**(4), 373−380.

Lukasová, A., 1979, Hierarchical agglomerative clustering procedure: Pattern Recognition, **11**(5−6), 365−381.

Mourot, G. 1993, Contribution au diagnostic des systèmes industriels par reconnaissance des formes (Doctoral dissertation): Doctorat del'Institut National Polytechnique de Lorraine, France.

Mourot, G., Bousghiri, S., and Ragot, J., 1993, Pattern recognition for diagnosis of technological systems: a review: International Conference on Systems IEEE, 275−281.

Nouri-Taleghani, M., Kadkhodaie-llkhchi, A., & Karimi-Khaledi, M., 2015, Determining hydraulic flow units using a hybrid neural network and multi-resolution graph-based clustering method case study from South Pars Gasfield, Iran: Journal of Petroleum Geology, **38**(2), 177−191

Pabakhsh, M., Ahmadi, K. and Riahi, M. A., and et al., 2012, Prediction of PEF and LITH logs using MRGC approach: Life Science Journal, **9**(4), 974−982.

Rogers, S. J., Fang, J. H., Karr, C. L., and et al., 1992, Determination of lithology from well logs using a neural network: AAPG Bulletin, **76**(5), 731−739.

Rosenblatt, F., 1958, The perceptron: a probabilistic model for information storage and organization in the brain: Psychological Review, **65**(6), 386.

Sutadiwirya, Y., Abrar, B., Henardi, D., and et al., 2008, Using MRGC (multi resolution graph-based clustering) method to integrate log data analysis and core facies to define electrofacies, in the Benua Field. In Central Sumatera Basin, Indonesia: International Gas Union Research Conference(IGRC), Paris. Currans Associates, Inc., 733−744.

Tang, H., Meddaugh, W. S., and Toomey, N., 2011, Using an artificial-neural-network method to predict carbonate well log facies successfully: SPE Reservoir Evaluation & Engineering, **14**(01), 35−44.

Tian, Y., Xu, H. Zhang, X. Y., et al., 2016, Multi-resolution graph-based clustering analysis for lithofacies identification from well log data: case study of intraplatform bank gas fields, Amu Darya Basin: Applied Geophysics, **13**(4), 598−607.

Tian, Y., Zhang, X. Y., Zhu, G. W., et al., Controlling effects of paleogeomorphology on intraplatform shoal reservoirs distribution and gas reservoirs characteristics: Taking intraplatform shoal gasfields of the Amu Darya basin as examples: Natural Gas Geoscience, 27(2), 320−329.

Ye, S. J., and Rabiller, P., 2000, A new tool for electro-facies analysis: multi-resolution graph-based clustering: SPWLA 41st Annual Logging Symposium, Dallas,Texas, USA,Jun 4−7.

Zhu, X. J., and Ghahramani, Z. B., 2002, Learning from labeled and unlabeled data with label propagation: Carnegie Mellon University CALD tech report CMU-CALD-02-107.

Appendix

Algorithm 1 The algorithm of getting the passage between attraction sets

```
Input: attraction sets ($S$), boundary points sets ($BPS$) , $K'$
Output: passage list ($L$)
for EACH $i$ in $[1, N]$ do
    for EACH $m$ in $[1, len(S_i)]$ do
        min$Dis \leftarrow Double\_MAX$
        $passage \leftarrow 0$
        if $P_m \in S_i$ and $P_m \in DPS$ then
            for Each $j$ in $[1, N]$ do
                for EACH $n$ in $[1, len(S_i)]$ do
                    if $P_n \in S_i$ and $P_n \in DPS$ then
                        if $P_m$ in the K'NN of $P_n$ or $P_n$ in the K'NN of $P_m$ then
                            if $D(P_m, P_n) <$ min$Dis$ then
                                min$Dis \leftarrow D(P_m, P_n)$
                                $passage \leftarrow$ M $in$ $(NI(P_m), NI(P_n))$
                            end if
                        end if
                    end if
            end for
    end for
end if
```

Algorithm 2 The algorithm of merging groups

```
Input: value list $L$, selected cluster kernel set($CKS$)
Output: merge results
    $L.sort$ $(P_1, P_2, passage)$by passage in decreasing order
    for items in $L$ do
        if set $S1$ corresponding to point $P1$ contains a point in $CKS$ then
            if set $S1$ corresponding to point $P1$ contain a point in $CKS$ then
                continue
            end if
        end if
    merge set $S1$ and $S2$
end for
```

Wu Hongliang, senior engineer, received a Ph.D in Earth exploration and information technology from Research Institute of Petroleum Exploration & Development in 2013. He is currently an enterprise expert of China Petroleum Exploration and Development Research Institute, devoted to the research on well logging processing and interpretation methods.

Email: wuhongliang@petrochina.com.cn

APPLIED GEOPHYSICS
（应用地球物理）

Vol.17, No.1 (March 2020)

中文摘要

各向异性地层随钻方位电磁波测井探测特性分析//Detection performance of azimuthal electromagnetic logging while drilling tool in anisotropic media, 巫振观[1,2,3]，王磊 [1,2,3]，范宜仁[1,2,3]，邓少贵[1,2,3]，黄瑞[4]，邢涛[1,2,3]，**APPLIED GEOPHYSICS**, 2020, **17**(1), P. 1–12. DOI: 10.1007/s11770-020-0804-z

(1.中国石油大学（华东）地球科学与技术学院，山东青岛 266580；2.海洋国家实验室海洋矿产资源评价与探测技术功能实验室，山东青岛 266071；3.中国石油大学CNPC测井重点实验室，山东青岛 266580；4.中国石油国际勘探开发有限公司, 北京 100034)

摘要：随钻方位电磁波测井仪器已广泛应用于大斜度井/水平井钻井，但地层各向异性使得地质导向及储层评价更为困难。为定量分析地层各向异性对随钻方位电磁波测井响应及资料处理影响，本文借助正反演技术，详细考察了测井响应对各向异性的敏感性及测井数据反演精度。首先，定义了敏感因子，定量分析了不同磁场分量、合成信号对各向异性的敏感性；然后，模拟了层状各向异性地层中的随钻方位电磁波测井仪器响应，对比分析了对称设计和非对称设计仪器的响应特征及对地层参数的敏感性；最后，探讨了反演模型的选取对各向异性地层界面反演精度的影响。数值模拟及反演结果表明：随钻方位电磁波测井响应受各向异性影响严重，通过对称补偿设计可降低地质信号对各向异性的敏感性，进而增强对地层界面的探测能力；各向异性地层中，视电阻率曲线出现分离及“犄角”等现象，界面附近方位信号则可能出现复杂非线性变化；反演中若不考虑各向异性的影响，难以获取准确的地层界面和电阻率信息，进而可能导致地质导向决策的失误。

关键词：各向异性，随钻方位电磁波测井，敏感性，反演

一种自适应多分辨率图聚类测井相分析方法//Adaptive multi-resolution graph-based clustering algorithm for electrofacies analysis, 武宏亮[1]，王晨[3]，冯周[1]，原野[1]，王华锋[2,3]，徐彬森[1]，**APPLIED GEOPHYSICS**, 2020, **17**(1), P. 13–25. DOI: 10.1007/s11770-020-0806-x

(1.中国石油勘探开发研究院，北京 100083；2.北方工业大学，北京 100144；3.北京航空航天大学，北京 100191)

摘要：测井相分析是通过自动聚类方法对多维测井曲线进行分析，进而进行相聚类与预测。基于图的多分辨率聚类(Multi-Resolution Graph-based Clustering，MRGC) 方法是一种常用的测井相分析方法，然而MRGC算法非常耗时，并且在传播过程中高度依赖初始参数，实际应用效益差。本文提出了一种自适应多分辨率图聚类 (Adaptive Multi Resolution Graph based Clustering，AMRGC) 分析方法。该方法不仅能提高测井相计算效率，而且能获得稳定的测井相传播结果。本文方法的两个核心算法是：1）轻核代表指数（L-KRI）算法只需计算少量“自由吸引”点，有效提高了计算效率；2）采用了反向传播算法（BP）与多层感知器（MLP）神经网络，有效避免了传统K 近邻算法因随机初始化参数导致的不稳定结果。实验结果表明，本文方法在聚类和传播预测任务上优于传统的MRGC方法，具有更高的运行效率和稳定性；同时，在没有数据先验知识的条件下效果明显优于自组织映射（SOM）、动态聚类（DYN）和自底向上的层次聚类(AHC)等其它常用聚类方法。

关键词：MRGC，AMRGC，MLP，测井相分析

多尺度多组分数字岩心构建与弹性模拟研究//Multi-scale and multi-component digital core construction and elastic property simulation, 崔利凯[1]，孙建孟[1]，闫伟超[1]，董怀民[1]，**APPLIED GEOPHYSICS**, 2020, **17**(1), P. 26–36. DOI: 10.1007/s11770-019-0789-7

(1.中国石油大学（华东）地球科学与技术学院, 山东青岛 266580)

摘要：常规数字岩心模型由于尺寸较小，对于非均质性较强的复杂岩心，难以具有代表性，模拟得到的岩石物理参数也难有参考性。以三组砂岩样品为例，本文提出了一种可行的构建多尺度多组分数字岩心的方法。利用CT 对柱塞岩心和毫米柱塞样品进行不同分辨率的扫描成像，并利用改进的图像配准方法进行扫描图像的精确配准。对于高分辨率扫描图像，利用常规图像分割方法，将岩心分割为孔隙和不同的矿物组分。基于配准关系，构造低分辨率图像的灰度与孔隙度、矿物组分含量的关系曲线。将构造的关系曲线应用到低分辨率扫描图像的图像分割过程。完成分割之后的数字岩心集合即构成了多尺度、多组分的数字岩心模型。基于多尺度多组分数字岩心模型，考虑了四种矿物模型，分别研究了细尺度和粗尺度下随着模型尺寸变化，三组岩心样品弹性模量的变化规律。结果表明：利用多尺度多组分数字岩心模型能够克服常规模型存在的代表性问题，实现岩心不同尺度孔隙、矿物的准确表征。在大尺度上计算的岩石弹性参数更具备代表性，同时岩石矿物组分考虑的越充分，模拟结果与实验结果越吻合。

关键词：弹性参数，数字岩心，多尺度，多组分

喀斯特地区煤田多孔联采3D-RVSP勘探实践：以中国五轮山为例//Multi-hole joint acquisition of a 3D-RVSP in a karst area: Case study in the Wulunshan Coal Field, China, 胡明顺[1,2]，潘冬明[◆1]，周福宝[2]，李娟娟[3]，王扬州[4]，陈圣恩[5]，许永忠[1]，**APPLIED GEOPHYSICS**, 2020, **17**(1), P. 37–53. DOI: 10.1007/s11770-020-0808-8

(1.中国矿业大学资源与地球科学学院，江苏徐州 221116；2.中