IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 28, 2018, accepted December 9, 2018, date of publication December 21, 2018, date of current version March 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2889122

Fast Data Reduction With Granulation-Based Instances Importance Labeling

XIAOYAN SUN[1], (Member, IEEE), LIAN LIU[1], CONG GENG[1], AND SHAOFENG YANG[2]
[1]Information and Control Engineering College, China University of Mining and Technology, Xuzhou 221116, China
[2]Asset Management Co., Ltd., China University of Mining and Technology, Xuzhou 221116, China

Corresponding author: Xiaoyan Sun (xysun78@126.com)

This work was supported by the National Natural Science Foundation of China under Grants 61473298 and 61876184.

ABSTRACT Data reduction has become greatly significant prior to applying instance-based machine learning algorithms in the Big Data era. Data reduction is used to reduce the size of data sets while retaining representative data. Existing algorithms, however, suffer from heavy computational cost and in having tradeoff in size reduction rate and learning accuracy. In this paper, we propose a fast data reduction approach by using granular computing to label important instances, i.e., instances with higher contributions to the learning task. The original data set is first granulated into K granules by applying K-means to a mapped lower-dimension space. Then, the importance of each instance in every granule is labeled based on its Hausdorff distance. Those instances whose importance values are lower than an experimentally tuned threshold are eliminated. The presented algorithm is applied to kNN classification tasks with eighteen different sizes of data sets from the UCI repository, and its outstanding performance in classification accuracy, size reduction rate, and runtime is illustrated by comparing with seven data reduction methods. The experimental results demonstrate that the proposed algorithm can greatly reduce the computational cost and achieve a higher classification accuracy when the reduction size is the same for all the compared algorithms.

INDEX TERMS Data reduction, granular computing, data importance label, kNN.

I. INTRODUCTION

With the explosive growth of data volume in industry and scientific domains, Big Data has attracted great attention from various applications due to its enormous potential value. However, the processing capacity of popular machine learning is struggling under this growth [1], [2]. Besides, storing large scale of instances can result in a great memory cost and a slow execution speed for instance-based learning algorithms. One alternative for tackling the problem is to perform data reduction as a preprocessing step of machine learning [3]. Data reduction is to remove the unimportant instances, e.g., noisy, redundant or less related ones, from the original datasets without greatly deteriorating the learners' performance. It is expected to enhance the performance of instance-based learning algorithms with smaller computational and storage cost [4].

Similar to the feature selection [5], [6], data reduction methods can be divided into wrapper and filter [7]. A wrapper method is usually carried out for classification tasks and performed based on the classification accuracy, i.e., instances with less contribution to the accuracy will be removed from the initial training set [8]. The filter algorithms of data reduction use a defined selection metric, e.g., clustering centers or marginal points of a cluster measured by distances variations before and after instances selection, for data reduction rather than the classification results. These existing data reduction methods can reduce the data size and improve the efficiency in data storing and mining, however, it is difficult to reach a tradeoff among classification accuracy, reduction ratio and lower computational cost. It is expected to make a great improvement by combining the wrapper and filter, i.e., both accuracy and filter distance metrics are concerned to label the important instances. As for the distance calculation, in the traditional filter methods, the instances of an entire dataset are usually compared and which will bring great computational cost when the size of a dataset is great. A ''divide and conquer'' strategy should be more efficient in such scenarios.

As is well known, in the field of feature selection, Granular Computing (GrC) methods have been successfully developed to obtain the important features based on the definition of information granules [9]–[11]. Its basic idea is to solve

problems at different levels of granularity, and establish an effective and user-centric concept to simplify people's cognition of the physical world. Such methods have good flexibility [12], and can improve efficiency and reduce costs. As we addressed before, the core of data reduction is to find the important or representative data from the original datasets with an acceptable performance maintaining on the instance-based learning and relative lower computational cost [13], [14]. However, Grc has not been sufficiently developed in the field of data reduction. It is feasible to introduce GrC into the data reduction by defining instances importance based on the data granules instead of traditional wrapper and filter who must be carried out by first applying classification on the entire dataset. Accordingly, the computational cost can be reduced. As in the filter-based methods, calculation on distances usually cannot be avoided in defining important instances, e.g., instances on the margin are more important than those in the internal of a class. Therefore, appropriate distance metric must be well developed in the data reduction based on the GrC granules. The Hausdorff distance defined for measuring the similarity between two datasets is the best choice for comparing the granules here.

Motivated by this, we here present a fast data reduction method, named as FDR-GIIL (Fast Data Reduction with Granulation based Instances Importance Labeling). It looks for important instances based on GrC and the Hausdorff distance on condition of less deteriorating the learner's performance. The GrC and K-means are first conducted to get the granules with different granularities, i.e., the entire dataset to be reduced is separated into K clusters. And then, the Hausdorff distance is adopted to label the importance of instances in each granule. Instances with smaller contributions are defined as unimportant ones and removed from its granule. This process is repeated until the reduction ratio is reached. Compared with traditional instance selection whose instances are deleted or selected by comparing all the instances in the entire datasets, the proposed FDR-GIIL algorithm adopts the "divide and conquer" strategy in this paper, which is expected to greatly reduce the computational cost and improve the performance of data reduction.

To summarize, our main contributions are as follows: (1) An improved GrC with lower computational cost is presented and applied to obtain the granules for first dividing the reduced datasets with different granularities. (2) The Hausdoff distance is introduced here to calculate the similarity between two granules with and without deleting instances, larger Hausdorff distance means greater importance of the selected instance to be deleted. (3) Besides, a crowding degree is defined for further selecting instances with same or similar Hausdoff distances. A data reduction metric is presented based on the Hausdoff distance and crowding degree. (4) The extensive experiments on Benchmark datasets demonstrate that the computational cost is greatly reduced with even improving the classification accuracy.

The rest of the paper is organized as follows. In Section 2, a survey of data reduction algorithms is presented. The proposed algorithm and related theoretical analysis, including data granulating and definition of data importance are presented in Section 3. The experiments and results are demonstrated and discussed in Section 4. The conclusions are finally followed.

II. RELATED WORK

A. WRAPPER ALGORITHMS

Wrapper algorithms are usually developed for classification problems, and the basis metric in most wrapper algorithms is the variation of the classification accuracy before and after data reduction. The Condensed Nearest Neighbor (CNN) proposed by Hart [15] is powerful for selecting instances by using the k-nearest neighbor. CNN picked out a consistent subset, i.e., a subset can correctly classify all instances from the original dataset, however, it cannot promise to find the smallest consistent subset and its performance is data sequence-dependent. Then some improved CNN algorithms were developed, e.g., generalized condensed nearest neighbor (GCNN) which selected an instance when differences between the instance distances to its nearest neighbors and its nearest enemies are higher than a given threshold [16]. Wilson [17] proposed edited nearest neighbor ENN) algorithm to deal with noisy instances by removing an instance p from the original dataset when the class of p was not consistent with the majority of its k nearest neighbors. In [18], the decremental reduction optimization procedures algorithms (DROPs) including DROP1, DROP2,..., DROP5 were further proposed by Wilson. The DROPs deleted the instance p when the instances in the reduced set can still be correctly classified without p. The provided experiments showed that the performance of DROP3 and DROP5 were optimal in the DROPs and also outperformed ENN and CNN. The local set-based smoother (LSSm) was proposed for removing the instances with a harmfulness greater than their usefulness [19]. The experiment result was excellent in accuracy, but lower in reduction rate. The combination of feature extraction and instance selection could reduce the large amount of computational time in training the classifiers. In [20], the Cuttlefish optimization algorithm was used for instance selection, while principal component analysis was used for feature extraction in [20]. The optimal extracted subset of data points and reduced feature space provided promising detection rate, accuracy rate, however, the fitness calculation was time consuming.

B. FILTER ALGORITHMS

The filter algorithms of data reduction use a defined selection function for data reduction rather than the classification results. Lumini and Nanni [21] proposed a data reduction algorithm based on clustering (CLU). The algorithm divided the original dataset into clusters, then selected the center point of each cluster as the representative instance, and obtained

the final reduction dataset. The Prototype Selection Based Clustering (PSC) algorithm in [22] differed from the CLU algorithm in that the PSC deleted some internal instances in the class and retained class boundary instances. PSC retained not only the border instances but also some internal ones. Vallejo and Ortega [23] proposed an instance selection based on ranking (ISR). The ISR algorithm calculated the correlation of the instances in the training set, and then processed the instances in descending order according to their correlation. In [24], instance rank based borders for instance selection (IRB) algorithm was proposed. The algorithm used the ENN to denoise the dataset, and then sorted those instances locating on the classifier's boundary according to the defined sorting function. At the same time, the intra-class data was retained according to a certain proportion, and the reduced dataset was obtained. The experiments showed that the method achieved a tradeoff between classification accuracy and execution time, and had obvious advantages for processing large-scale datasets. In [25], local density-based instance selection (LDIS) was proposed. The algorithm evaluated the instances of each class separately and only kept the densest instances in a given neighborhood. This ensured a reasonably low time complexity.

The existing data reduction algorithms can reduce data size either based on the iterated classification or by defining instance selection functions, most of them are time consuming. For Big Data, such algorithms are difficult to be performed. Therefore, developing fast data reduction method is significant for keeping the pace of data increasing. Motivated by these, a fast data reduction (FDR-GIIL) by labeling important instances based on GrC and Hausdorff distance is designed here for effectively finding useful data since GrC has shown its power in feature selection.

III. IMPORTANT INSTANCES LABELING AND SELECTION

The "division and conquer" strategy is a good choice for large size data reduction, i.e., the dataset is first separated into smaller ones, and the data reduction is conducted in the relatively smaller size space. The computational cost is expected to be reduced with such a strategy. What's more, another purpose of our work is to give a more generalized method for the data size reduction without considering any priori information of the dataset (no matter for clustering, classification or regression tasks). Therefore, the latent relationships or distributions of the instances should be first discovered, and the important instances are then labeled and selected in this paper.

A. DATA GRANULATION

The granular computing is adopted here to perform the division due to its advantage in exploring the relationships among data features. The computational cost of the granulating process is expected to be greatly reduced for Big Data with large number of features. Accordingly, we here present a mapping based granulation, i.e., the instances with higher dimensional features are first mapped into a lower dimensional feature space and then granulated.

The dataset to be analyzed is denoted as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i = \left(x_{i_1}, x_{i_2}, \ldots, x_{i_c}, y_i\right)$ with x_{i_j} being the j-th ($j = 1, 2, \ldots, c$) attribute value of the i-th ($i = 1, 2, \ldots, n$) instance and y_i being the decision attribute (class label) of $\mathbf{X}$. The mapping function defined in [26] is applied here:

$$\boldsymbol{\tau}(\mathbf{x}_i) = \sqrt{\sum_{j=1}^{c} x_{ij}^2} \tag{1}$$

Clearly, with such a mapping, the original instance $\mathbf{x}_i$ with c-dimension features is mapped into a one dimensional feature space scaled by $\boldsymbol{\tau}(\mathbf{x}_i)$. The scope of $\boldsymbol{\tau}(\mathbf{x}_i)$ varies in the range of $[0, r]$, and

$$r = \sqrt{\sum_{j=1}^{c} \left(\max_{1 \le i \le n} \left| x_{i_j} \right| \right)^2} \tag{2}$$

In the mapped space, we then can use the clustering methods to fulfill the granulating according to the instances' similarities. Whether the mapping can well keep the similarities in the mapped space as that in the initial one should be further analyzed. Otherwise, the feasibility of the clustering based granulating cannot be guaranteed.

For two instances $\mathbf{x}_p$ and $\mathbf{x}_q$ in the original dataset $\mathbf{X}$ their distance d_{pq} can be calculated as follows :

$$\begin{aligned} d_{\mathrm{pq}} &= \sqrt{\sum_{j=1}^{c} \left(x_{p_j} - x_{q_j}\right)^2} \\ &= \sqrt{\sum_{j=1}^{c} x_{p_j}^2 + \sum_{j=1}^{c} x_{q_j}^2 - 2\sum_{j=1}^{c} x_{p_j} \cdot x_{q_j}} \end{aligned} \tag{3}$$

With the mapping, $\mathbf{x}_p$ and $\mathbf{x}_q$ are mapped to $\boldsymbol{\tau}(\mathbf{x}_p)$ and $\boldsymbol{\tau}(\mathbf{x}_q)$. The distance between $\boldsymbol{\tau}(\mathbf{x}_p)$ and $\boldsymbol{\tau}(\mathbf{x}_q)$ is

$$\begin{aligned} d'_{\mathrm{pq}} &= \boldsymbol{\tau}(\mathbf{x}_p) - \boldsymbol{\tau}(\mathbf{x}_q) \\ &= \sqrt{\sum_{j=1}^{c} x_{p_j}^2} - \sqrt{\sum_{j=1}^{c} x_{q_j}^2} \end{aligned} \tag{4}$$

The difference between d_{pq} and d'_{pq} can be obtained as in Eq. (5):

$$d_{pq} - d'_{pq} = \frac{2\boldsymbol{\tau}(\mathbf{x}_p)\boldsymbol{\tau}(\mathbf{x}_q)(1 - \cos\theta)}{d_{pq} + d'_{pq}} \tag{5}$$

where θ represents the angle between $\mathbf{x}_p$ and $\mathbf{x}_q$ in the initial dataset space. It can be seen from Eq. (5) that the distance difference of two instances in the different spaces is slight unless these two instances are quite different. In such scenario, they actually belong to two distinct clusters, and can also be correctly clustered in the mapped space.

Accordingly, the mapping can well keep the similarity measure and the granulating in the space can be performed by using a clustering method. Here, the K-means clustering [28] is applied to separate the space into K granules due to its

simplicity and lower computational cost. The procedure of K-means can be found in [28].

In summary, the granulation here is divided into two steps: (1) Granular mapping: mapping instances with high-dimension features into one-dimension granulation space using the granular mapping function refer to Eq. (1). (2) Clustering granulating: the data in the granulation space are clustered into K granules by using K-means clustering.

In this granulating procedure, three advantages can be achieved: (1) Separating a large scale dataset into K smaller ones without greatly deteriorating the original distributions due to the granulating operators. (2) Reducing the amount of data. From Eq. (1), it can be discovered that points on a sphere or a circle will be mapped into one point in the granulation space. Therefore, such points are naturally reduced. (3) Benefiting for reducing the calculation complexity due to the mapping and granulating, i.e., the computational cost of K-means in the granulation space will be greatly decreased compared with that in the initial one. Furthermore, within the smaller granules, the instance importance labeling can also be time saving. Therefore, the computational cost can be reduced comparing with non-granulating.

B. IMPORTANT INSTANCES LABELING

If the similarity of the dataset $\mathbf{X}$ and $\mathbf{X}/\mathbf{x}_i$ (instance $\mathbf{x}_i$ is removed from $\mathbf{X}$) is high enough, it is reasonable to conclude that the instance $\mathbf{x}_i$ is unimportant. A similarity metric is important for implementing this observation. In this paper, the Hausdorff distance [28] presented to measure how far two subsets of a metric space are from each other is chosen to measure the similarity of two datasets. For improving the computational efficient, the similarity of $\mathbf{X}^K$ and $\mathbf{X}^K/\mathbf{x}_i$ of each granule $\mathbf{X}^K$ obtained in subsection III.A is calculated to discover the important instances of this granule. And the data size can be reduced by removing those unimportant instances.

Given two compared datasets $\mathbf{X}$ and $\mathbf{X}/\mathbf{x}_i$, their Hausdorff distance is calculated according to the following equations:

$$H(\mathbf{X}, \mathbf{X}/\mathbf{x}_i) = max(h(\mathbf{X}, \mathbf{X}/\mathbf{x}_i), h(\mathbf{X}/\mathbf{x}_i, \mathbf{X})) \quad (6)$$

$$h(\mathbf{X}, \mathbf{X}/\mathbf{x}_i) = \max_{\mathbf{x}_p \in \mathbf{X}, \mathbf{x}_q \in \mathbf{X}/\mathbf{x}_i} \min \left\| \mathbf{x}_p - \mathbf{x}_q \right\| \quad (7)$$

$$h(\mathbf{X}/\mathbf{x}_i, \mathbf{X}) = \max_{\mathbf{x}_p \in \mathbf{X}/\mathbf{x}_i, \mathbf{x}_q \in \mathbf{X}} \min \left\| \mathbf{x}_p - \mathbf{x}_q \right\| \quad (8)$$

Where $\left\| \mathbf{x}_p - \mathbf{x}_q \right\|$ is the distance norm between two vectors $\mathbf{x}_p$ and $\mathbf{x}_q$, $h(\mathbf{X}, \mathbf{X}/\mathbf{x}_i)$ represents the directional Hausdorff distance from granule $\mathbf{X}$ to $\mathbf{X}/\mathbf{x}_i$, and $h(\mathbf{X}/\mathbf{x}_i, \mathbf{X})$ is the inverse Hausdorff distance from granule $\mathbf{X}/\mathbf{x}_i$ to $\mathbf{X}$. The more similar $\mathbf{X}$ and $\mathbf{X}/\mathbf{x}_i$ are, the smaller the Hausdorff distance between $\mathbf{X}$ and $\mathbf{X}/\mathbf{x}_i$ is. Motivated by this, we here present the following method to label the importance of instances.

For an randomly selected instance $\mathbf{x}_i$, supposing it is removed from the granulated dataset $\mathbf{X}$, and we can have $\mathbf{X}/\mathbf{x}_i$, the importance of $\mathbf{x}_i$ is defined as follows :

$$sig(\mathbf{x}_i) = H(\mathbf{X}, \mathbf{X}/\mathbf{x}_i) \quad (9)$$

If two instances in the dataset are closer or similar, the Hausdorff distance between $\mathbf{X}$ and $\mathbf{X}/\mathbf{x}_i$ is small, i.e., the removed instance $\mathbf{x}_i$ has smaller influence on the distributions of $\mathbf{X}$, indicating that the contour features of the original dataset $\mathbf{X}$ can be well maintained after deleting the $\mathbf{x}_i$, thus the importance of $\mathbf{x}_i$ is small. Conversely, the importance of the reduced instance is larger. Therefore, the Hausdorff distance is feasible for labeling the importance of the instances.

It seems that the calculation of $H(\mathbf{X}, \mathbf{X}/\mathbf{x}_i)$ is time consuming since all instances should be compared according to Eq. (6). In fact, after mapping and granulation, the calculation cost is greatly reduced when labeling data importance. For our importance labeling, we only need to calculate $\sum_{g=1}^{K} \frac{1}{2} n_g \left(n_g - 1\right)$ elements (n_g is the number of instances in the g-th granule) for obtaining the distance of each pair of instances and get a symmetric distance matrix with zero diagonal of the dataset $\mathbf{X}$. Then, the minimum value of column (row) i is the importance of $\mathbf{x}_i$. It is clear that such a labeling is obtained without any clustering or classification based iteration, therefore, the computational cost is further reduced.

Those instances whose importance are smaller than a given threshold μ will be removed from the original dataset, and we save these instances to be deleted in set $\mathbf{X}_\mu$. It is possible that more than one instances may have the same importance and can all be removed. To avoid such a case, a crowding degree is further used here to filter out more crowded ones from the same important instances. The crowding degree is defined in Eq. (10):

$$cd\left(\mathbf{x}_i\right) = \frac{1}{\sum\limits_{\mathbf{x}_j \in \mathbf{X}_\mu} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|}, \quad \mathbf{X}_\mu = \{\mathbf{x}_i \,|sig(\mathbf{x}_i) \leq \mu\} \quad (10)$$

The larger value of $cd(\mathbf{x}_i)$ indicates that many instances in $\mathbf{X}_\mu$ are very closer to $\mathbf{x}_i$ and $\mathbf{x}_i$ should be deleted.

An example of importance labeling and instances removing of $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$ is shown in Fig. 1 ($\mathbf{x}_1$(1, 2), $\mathbf{x}_2$(1.5, 2), $\mathbf{x}_3$(1.5, 2.5) and $\mathbf{x}_4$(2, 0.5)). The data importance and its crowding degree are labeled beside each point of $\mathbf{X}$ in Fig.1. When the data importance threshold μ is set in the range of (0.5, 1.581), $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ will be discarded at the same time if the crowding degree is ignored, which will lead

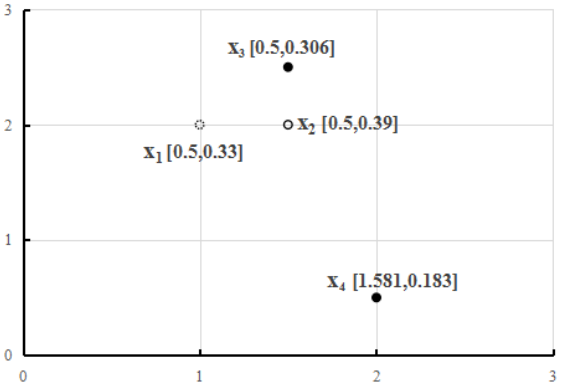


FIGURE 1. Instance importance labeling of $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$.

Algorithm 1 Fast Data Reduction With Granulation Based Instances Importance Labeling (FDR-GIIL)

Input: the original dataset $\mathbf{X}$.

Output: the reduced datdaset $\mathbf{X}'$.

1: Normalize $\mathbf{X}$, and map it into the granulation space, denoted as $\mathbf{GX}$ with Eq.(1).
2: Use K-Means to granulate the $\boldsymbol{\tau}(\mathbf{x}_i)$ in the granulation space $\mathbf{GX}$ and obtain K granules $\{\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^K\}$.
3: Calculate the importance of $\boldsymbol{\tau}(\mathbf{x}_i)$ based on the granule $\mathbf{X}^g$ $(g = 1, 2, \ldots, K)$ and $\mathbf{X}^{g'} = \mathbf{X}^g - \{\boldsymbol{\tau}(\mathbf{x}_i)\}$ according to Eq. (9).
4: Deduplicate the instances in $\mathbf{X}$ that mapped into one $\boldsymbol{\tau}(\mathbf{x}_i)$ (instances distributed on a circle or a sphere).
5: Judge whether the data importance of $\boldsymbol{\tau}(\mathbf{x}_i)$ is greater than the threshold μ, if yes, execute step (6), otherwise $\boldsymbol{\tau}(\mathbf{x}_i) \in \mathbf{X}_\mu$ and go to step(7).
6: Retain the instances.
7: Calculate the crowding degree of instances with the same data importance in $\mathbf{X}_\mu$, and delete the instances with larger crowding degree from $\mathbf{X}_\mu$.
8: Count the ratio of reduction.
9: Output $\mathbf{X}'$ until the ratio meets the requirment.

to an obvious distribution change in $\mathbf{X}$. When considering the crowding, $\mathbf{x}_2$ with larger crowding degree is first removed from $\mathbf{X}$, then $\mathbf{x}_1$ is removed if necessary, until the data reduction ratio is satisfied.

C. PSEUDO OF THE PROPOSED ALGORITHM

IV. EXPERIMENTS

A. DATASETS

Eighteen datasets selected from the UCI Repository [29] with different sizes are reduced to demonstrate the effectiveness of the proposed algorithm. Among these datasets, there are eight small and medium ones including five Numeric (Glass, Iris, Liver, Vehicle, Wine) and three Mixed (Echocardiogram, Hepatitis, Zoo) datasets, and the specific attributes are listed in TABLE 1. The other Ten large datasets including four Numeric (Segmentation, Magic, Letter, Shuttle) and six Mixed (Chess, Poker 90k, Covertype, Census Income, Poker 350k, KDD Cup 800k) datasets are chosen and shown in TABLE 2.

TABLE 1. Attributes of small and medium datasets.

Dataset	# of instances	# of Attr-Num	# of Attr-Cat
Echocardiogram	132	7	2
Glass	214	10	0
Hepatitis	155	6	13
Iris	150	4	0
Liver	345	7	0
Vehicle	846	18	0
Wine	178	13	0
Zoo	101	0	16

TABLE 2. Attributes of large datasets.

Dataset	# of instances	# of Attr-Num	# of Attr-Cat
Segmentation	2100	19	0
Magic	19,020	10	0
Letter	20,000	16	0
Chess	28,056	0	6
Shuttle	58,000	9	0
Poker 90k	90,000	0	10
Covertype	250,000	10	44
Census Income	299,285	7	33
Poker 350k	350,000	0	10
KDD Cup 800k	800,000	34	7

Note: Attr-Num : numerical attributes ; Attr-Cat: categorical attributes

Four groups of experiments are conducted: (1) effectiveness of mapping-based granulation: the K-means clustering differences before and after mapping are compared to demonstrate the corresponding performance, (2) visualizing the data reduction to intuitively illustrate the performance of the presented algorithm, (3) experimental results on small and medium datasets and (4) experimental results on large datasets. The state-of-the-art data reduction methods are compared here.

B. PARAMETER SETTINGS

Three parameters are required for the proposed FDR-GIIL algorithms, that is, the number of granules K, data importance threshold μ and the reduction ratio. In this paper, K granules are obtained by K-means clustering, and K is set as the number of categories in the original dataset. In order to keep the same reduction ratio with other compared algorithms, the reduction ratio of each dataset is maintained at around 59% in this paper. Data importance threshold determines the reduction ratio of datasets. We here perform a series of experiments on each dataset to construct cubic spline interpolation function between data importance and data reduction ratio. Therefore, data importance can be estimated according to specific reduction ratio.

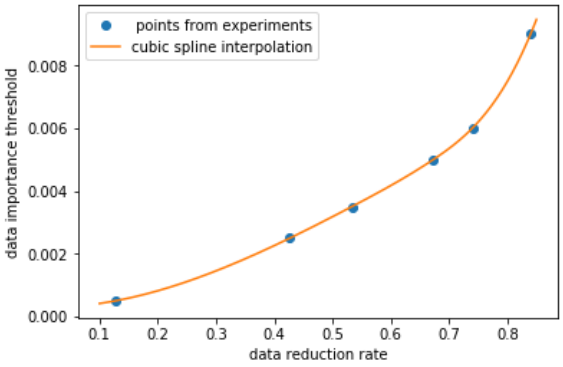


FIGURE 2. The approximated curve between data importance threshold and data reduction ratio of Iris.

An example of data importance threshold determining is presented as follows. The relationship between the data importance threshold and data reduction ratio of Iris is approximated by a spline interpolation and the curve is shown

in Fig.2. Given the data reduction rate 59% of Iris, data importance can be determined as 4.06×10^{-3}.

Moreover, in order to validate the efficiency of FDR-GIIL algorithm, two indicators are used in the experiments: relative classification accuracy and the algorithm runtime (in seconds).

The relative classification accuracy demonstrates the ability of the data reduction algorithms to maintain the classification performance of the datasets. It indicates that the data reduction algorithm can improve the classification ability of the dataset if relative classification accuracy is greater than 0. The runtime of the algorithm can reflect the efficiency of the algorithm reduction. A smaller value indicates a smaller computational cost of the data reduction algorithm.

C. EFFECTIVENESS OF MAPPING-BASED DATA GRANULATION

The purpose of mapping-based granulation is to reduce the computational cost with possibly maintaining the distribution of the instances of the original dataset. To show the variations on the instances' distributions of the mapped granules and the original datasets, the Iris (sepal length, petal length) is used to perform the experiments. The original dataset and the dataset after granular mapping are respectively performed to obtain $K(K=3)$ granule sets by clustering granulation. Fig.3 shows the instances distributions of the original dataset, the results of clustering on the original dataset and the mapping based granulated dataset are shown in Fig.4 and Fig.5, respectively.

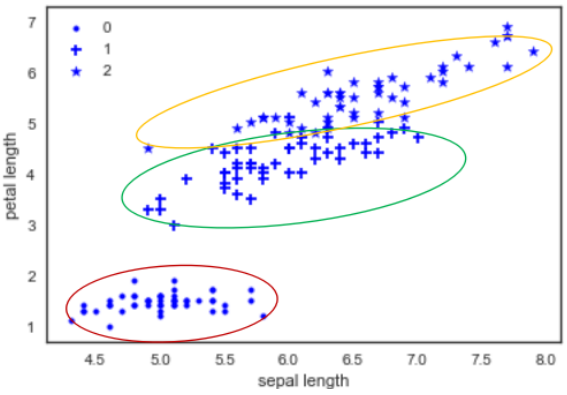


FIGURE 3. The class distribution of original dataset.

As can be seen that the differences of clustering between the original and the granulated dataset is small for the Iris dataset (the eight points of clustering variations due to granular mapping are marked with small red circle in Fig.5). Furthermore, an indicator Rand Index (RI) [28], [30] of clustering on the granulated mapping dataset is calculated according to Eq. (11) to sufficiently and numerically demonstrate the effectiveness of the granulation. Supposing that λ and λ^* are the cluster labels of granular mapped dataset and original dataset, respectively, n is the number of instances in $\mathbf{X}$, we pair the samples of $\mathbf{X}$ as $(\mathbf{x}_p, \mathbf{x}_q)$, and define $a = |\mathbf{SS}|$, $b = |\mathbf{DD}|$, where $\mathbf{SS} = \left\{(\mathbf{x}_p, \mathbf{x}_q)|\lambda_p = \lambda_q, \lambda_p^* = \lambda_q^*, p < q\right\}$,

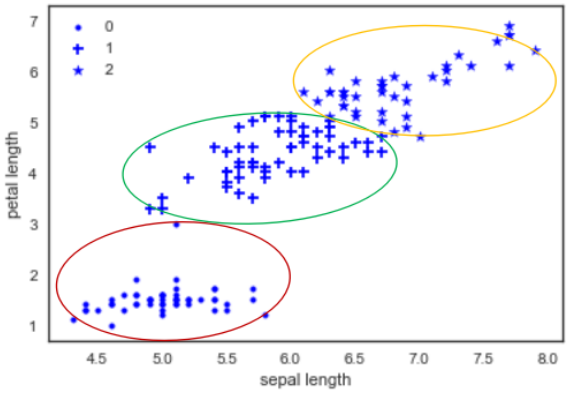


FIGURE 4. The clustering results on original dataset.

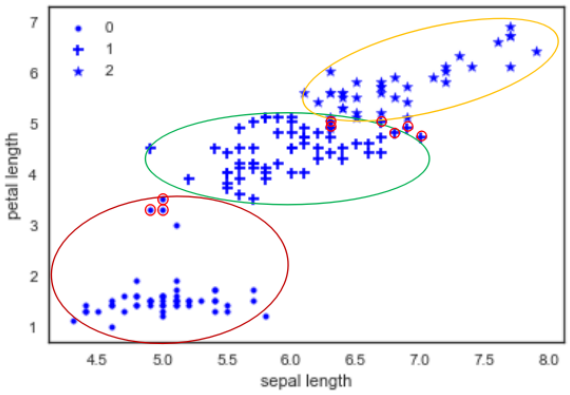


FIGURE 5. The clustering results on granular mapped dataset.

$$\mathbf{DD} = \left\{(\mathbf{x}_p, \mathbf{x}_q)|\lambda_p \neq \lambda_q, \lambda_p^* \neq \lambda_q^*, p < q\right\}.$$

$$RI = \frac{2(a+b)}{n(n-1)} \tag{11}$$

The larger the value of RI is, the smaller the variation of the instances' distribution is. The RI values for all the 18 compared datasets are calculated and listed in TABLE 3 (small to medium size datasets) and TABLE 4 (large scale datasets).

TABLE 3. The rand index (RI) of small and medium datasets after granular mapping.

Dataset	# of Class	RI
Echocardiogram	2	0.75
Glass	7	0.69
Hepatitis	2	0.61
Iris	3	0.91
Liver	2	0.80
Vehicle	3	0.82
Wine	3	0.92
Zoo	7	0.99

The following conclusions can be observed from Figures 3 to 5, and TABLE 3 to 4: (1) The RI of Iris, Wine, and Zoo is close to 1, indicating that the distribution of

TABLE 4. The rand index (RI) of large datasets after granular mapping.

Dataset	# of Class	RI
Segmentation	7	0.85
Magic	2	0.71
Letter	26	0.93
Chess	18	0.87
Shuttle	7	0.69
Poker 90k	10	0.76
Covertype	7	0.65
Census Income	2	0.86
Poker 350k	10	0.78
KDD Cup 800k	6	0.92

the datasets is well maintained after the granular mapping. (2) The RI values of Glass, Hepatitis and Shuttle (large scale) are relatively smaller, indicating that the original instances distributions of these dataset may be changed by using the mapping based granulation, however, such a change may be helpful for improving the classification or analysis of the original dataset, which can be further proved by the results shown in TABLE 5 in that the classification accuracy is improved after reducing number of instances of some granules. (3) For the large scale datasets, the total distributions of the selected datasets also can be well maintained, illustrating that the presented granulation may be more beneficial to the Big Data scenarios.

TABLE 5. Classification accuracy of small and medium datasets for FDR-GIIL with kNN($k = 3$).

Dataset	μ (10^{-3})	Orig	FDR-GIIL
Echocardiogram	4.60	**88.42%**	**91.60%** ↑
Glass	1.80	**68.85%**	**75.81%** ↑
Hepatitis	5.80	69.02%	68.26% ↓
Iris	4.06	95.93%	95.00% ↓
Liver	1.80	**63.90%**	**72.78%** ↑
Vehicle	2.40	**65.23%**	**73.72%** ↑
Wine	4.70	**71.12%**	**76.19%** ↑
Zoo	13.00	94.47%	88.15% ↓

In summary, the distributions or features of the most of the selected datasets are well maintained by using the mapping based granulation, and for those changed ones, the change is helpful for removing some unimportant instances and improve the classification accuracy (combining with the analysis of the results in TABLE 6 and 8).

D. VISUALIZATION OF DATA REDUCTION

For intuitively demonstrating the data reduction of the proposed algorithm, the Iris is further used to visually show the results of the data reduction. After data granulation of the Iris (Sepal Length, Petal Length) in Section IV.C, the instance importance is labeled in the granulation space, and the data size is reduced based on the labeled importance. We here set the removing threshold value as $\mu = 0.019$, and reserve about

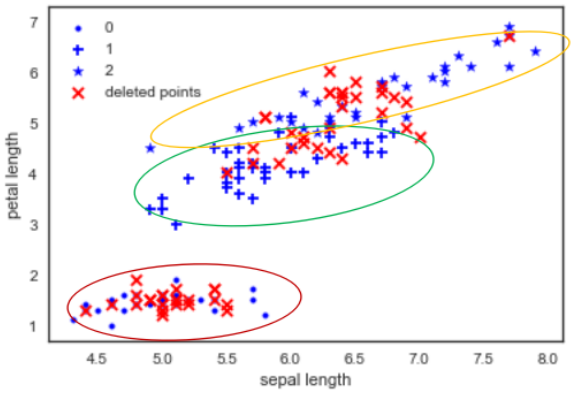


FIGURE 6. Result of data reduction for Iris.

half size of instances by further considering the crowding degree. The experimental result is shown in Fig.6. The points '×' in Fig.6 are the removed unimportant instances whose importance labels are less than μ. The reduction rate is 46% by further employing the crowding degree for those distances with the same importance.

It can be seen from Fig.6 that the boundary points of each class are well preserved, which is conducive to maintain the classification performance of the original dataset. After data reduction, kNN ($k = 3$) is used to classify the original and reduced datasets respectively, and the classification accuracy of the reduced dataset is 94.82% which is larger than that of the original one 94.6%, indicating that the data reduction of our algorithm is helpful for improving the classification by removing unimportant instances.

E. EFFICIENCY OF DATA REDUCTION

Two groups of experiments are further conducted here by comparing our proposed algorithm (FDR-GIIL) with other popular data reduction methods on small-medium datasets and large scale ones, respectively. In these experiments, the compared algorithms usually remove 59% instances from the original datasets, accordingly, we here also set the reduction rate as 59% for our method.

1) EXPERIMENTS ON SMALL AND MEDIUM DATASETS

The datasets with relatively small size are reduced and then classified using kNN. (1) The classification accuracy on the original datasets and the reduced ones by using our FDR-GIIL are compared. In this experiment, we apply 10-fold cross validation over each dataset. The experimental results are shown in TABLE 5.

The 'Orig' and 'FDR-GIIL' represent the kNN classification accuracy obtained on the original dataset and the reduced ones respectively. '↑' and '↓' are labeled to illustrate the increase and decrease in the classification accuracy. (2) Other seven data reduction methods, IRB, ISR-k-nn, ISR, DROP3, DROP5, CLU, and PSC are further compared to show the performance of the proposed algorithm in TABLE 6. The reason of selecting these algorithms here is that IRB,

TABLE 6. Relative classification accuracy of small and medium-sized datasets for the compared algorithms, using kNN(k=3).

Dataset	FDR-GIIL	IRB	ISR-k-nn	ISR	DROP3	DROP5	CLU	PSC
Echocardiogram	**3.6%**	−1.51%*	−26.25%*	−3.02%*	−0.56%*	−3.39%*	−15.84%*	−20.38%*
Glass	**10.1%**	−12.28%*	−26.38%*	−14.78%*	−12.22%*	−14.03%*	−25.69%*	−18.52%*
Hepatitis	−1.1%	**6.53%**	−1.62%	−3.95%*	−0.81%	−4.94%*	−4.19%*	−0.73%
Iris	−0.97%	**0.01%**	−38.73%*	−0.7%	−2.1%	**0.01%**	−5.63%*	0%
Liver	**14.27%**	2.75%*	−8.16%*	−1.53%*	5.08%*	3.63%*	−14.67%*	−3.98%*
Vehicle	**13.02%**	−8.04%*	−26.1%*	−18.23%*	−15.26%*	−14.56%*	−41.14%*	−12.19%*
Wine	**7.13%**	−4.28%*	−19.42%*	3.64%*	1.83%*	1.25%*	−5.75%*	0.32%*
Zoo	−6.69%	−5.81%	−59.31%*	−2.32%*	−5.81%	**0.01%**	−2.32%*	−2.32%*
Average	**4.92%**	−2.83%	−25.75%	−5.11%	−3.73%	−4%	−13%	−7.2%

ISR-k-nn and ISR are effective methods that achieve a better tradeoff between the classification accuracy and reduction rate, the DROPs outperform other instance selection algorithms in accuracy, and CLU and PSC are two of the fastest instance selection algorithms. The relative variations on the classification accuracy defined in Eq. (12) of each algorithm on the original and reduced datasets are calculated and recorded.

$$\text{relativeaccuracy} = \frac{\text{reducedacc} - \text{originalacc}}{\text{originalacc}} \tag{12}$$

In Eq. (12), '*originalacc*' and '*reducedacc*' represent the kNN classification accuracy of the original dataset and the reduced dataset. If the relative classification accuracy is greater than 0, it indicates that the data reduction is conducive to improve the classification performance of the datasets, otherwise, the instance reduction will reduce the classification of the datasets. The classification variations of our algorithm are listed in TABLE 5. The experimental results of all the compared algorithms on the relative accuracy are given in TABLE 6.

From TABLE 5, we can conclude that the classification accuracy of five datasets among the eight ones are improved by reducing the data size with the proposed algorithm, especially for the Glass, whose classification accuracy is greatly enhanced even its instances distributions are changed by granulation. The accuracy of Zoo is reduced after data reduction even its instances distributions are well maintained by granulation. The results indicate that the granulation and importance based data reduction is feasible but should be finely designed.

For the compared algorithms, the non-parametric Wilcoxon Signed test with a confidence level $\alpha = 0.05$ is conducted to show the statistical performance. The symbol '*' is marked on the results if FDR-GIIL significantly outperforms the others. The highest relative classification accuracy of each dataset is also bolded. From TABLE 6, it can be observed that the relative classification accuracy obtained by FDR-GIIL algorithm is higher than the compared algorithms. The relative classification accuracy of IRB is lower than FDR-GIIL, then followed by DROP3 and DROP5 algorithms. The ISR-k-nn algorithm has the lowest relative classification accuracy. To sum up, our proposed FDR-GIIL outperforms the compared algorithms in most cases by improving the classification accuracy.

TABLE 7. Classification accuracy of large datasets for FDR-GIIL with kNN(k = 3).

Dataset	μ (10^{-5})	Orig	FDR-GIIL
Segmentation	66.00	95.3%	91.25% ↓
Magic	2.30	80.06%	78.21% ↓
Letter	114	95.57%	93.20% ↓
Chess	0.61	**56.65%**	**61.12%** ↑
Shuttle	1.80	99.82%	99.68% ↓
Poker 90k	1272	55.72%	52.29% ↓
Covertype	1.20	97.06%	95.15% ↓
Census Income	4.00	**89.96%**	**91.18%** ↑
Poker 350k	918	57.95%	55.44% ↓
KDD Cup 800k	6.20	98.80%	98.78% ↓

2) EXPERIMENTS ON LARGE DATASETS

The similar experiments of absolute accuracy and relative accuracy variations on large scale datasets are further conducted. Besides, the executed time of the compared algorithms for the large scale datasets is added to show the computational cost of the proposed algorithm. In this section, we compare our approach with IRB since IRB has a great compromise between runtime and accuracy. DROP3 is selected due to its high classification accuracy. CLU and PSC are also included because they are fast instance selection algorithms. However, ISR is not included in this experiment due to its high space requirements for obtaining the large datasets ranking, it is unfeasible for large datasets. The corresponding results are recorded in TABLE 7, 8 and 9.

It can be concluded from TABLE 7 that the absolute accuracy of eight datasets decreases by using our algorithm, and that of the other two datasets (Chess and Census Income) improves.

In TABLE 8, we show the number of instances and attributes of each dataset. The '−' sign indicates that the execution time of the algorithm is more than 100 h (360,000 seconds). The results show that DROP3 is the slowest algorithm. On the other hand, CLU and PSC execute fast, but slower when the dataset is a mixed one or its size

TABLE 8. Relative classification accuracy of large datasets for the comparison algorithm.

Dataset	FDR-GIIL	IRB	DROP3	CLU	PSC
Segmentation	-4.25%	-6%	-4.15%	−16.09%*	**-3.95%**
Magic	**-2.31%**	−3.17%	−3.47%	−19.15%*	−13.05%*
Letter	**-2.48%**	−7.82%*	−2.56%	−55.19%*	−17.01%*
Chess	**7.89%**	−9.51%*	−2.90%*	−46.74%*	−23.53%*
Shuttle	−0.14%	**-0.1%**	−0.14%	−4.90%*	−3.09%*
Poker 90k	**-6.16%**	−10.79%*	–	–	–
Covertype	**-1.97%**	−6.71%*	–	–	–
Census Income	**1.36%**	0.84%	–	–	–
Poker 350k	**-4.33%**	−16.82%*	–	–	–
KDD Cup 800k	**-0.02%**	–	–	–	–
Average	**-1.24%**	−7.27%	−2.64%	−25.41%	−12.13%

TABLE 9. Executing time of large datasets (seconds).

Dataset	# of instances	# of Attr-Num	# of Attr-Cat	FDR-GIIL	IRB	CLU	PSC	DROP3
Segmentation	2100	19	0	**3.9**	27.0	6.0	7.0	53.0
Magic	19,020	10	0	**88.6**	211.5	167.1	172.1	2555.6
Letter	20,000	16	0	**14.5**	500.8	217.2	226.2	4765.5
Chess	28,056	0	6	**17.3**	1651.4	3762.3	3862.8	7447.2
Shuttle	58,000	9	0	**30.9**	2407.7	277.4	288.4	123000.1
Poker 90k	90,000	0	10	**78.3**	20301.4	58813.1	60519.5	68834.1
Covertype	250,000	10	44	**649.7**	150139.8	-	-	-
Census Income	299,285	7	33	**21673.4**	230375.7	-	-	-
Poker 350k	350,000	0	10	**500.0**	313667.1	-	-	-
KDD Cup 800k	800,000	34	7	**70655.3**	-	-	-	-

is greater than 58,000. The DROP3, CLU and PSC cannot work on some datasets with big size due to their expensive computational cost. IRB is feasible to reduce the large scale dataset, but its runtime is far longer than our algorithm. And the runtime of IRB is more than 100 h when dataset size is 800,000. The computational cost of our algorithm is reduced about half of the fastest runtime of the compared algorithms. The execution time of our algorithm is much smaller than the compared algorithms, the computational cost of FDR-GIIL is further greatly reduced when the size of the datasets increases, indicating that FDR-GIIL is fast for dealing with large scale datasets as we addressed before.

The relative accuracy obtained by different data reduction algorithms for the large datasets is shown in TABLE 8. The average relative accuracy is also computed for each algorithm. DROP3 obtains the second best relative accuracy in average, but cannot work for the four largest datasets. PSC and CLU are worse than the other algorithms, and they are also unfeasible for the four largest datasets. The non-parametric Wilcoxon Signed test with a confidence level $\alpha = 0.05$ is also used to evaluate the statistical significance of FDR-GIIL, and the significant values are marked with '*' in TABLE 8. It can be concluded that our algorithm outperforms the most compared algorithms in the relative accuracy, i.e., the decreased absolute accuracy of the reduced datasets of our algorithm is smaller than the compared ones; for some datasets, e.g., Chess and Census Income, our algorithm has a distinct improvement on the classification accuracy after removing about 59% instances.

In addition, one large dataset is used to show the runtime rising tendency of each algorithm with the growth of the data size. The experiment is performed with the Covertype dataset (54 features, 7 classes, 250,000 instances), and 10 training sets are constructed (from 10,000 instances to 100,000 instances). The runtime results for the Covertype dataset are presented in TABLE 10 and Fig.7.

TABLE 10. Runtimes (sec) spent by each algorithm over different training sets size from the covertype dataset.

# of Instances	FDR-GIIL	IRB	DROP3	PSC	CLU
10,000	**25.6**	151.6	1336.0	853.4	833.2
20,000	**40.8**	626.0	7648.9	2709.7	2668.4
30,000	**68.7**	1421.9	27,600.4	6100.9	6132.3
40,000	**92.0**	2424.1	-	25,560.2	25,231.9
50,000	**121.9**	4448.6	-	-	-
60,000	**164.9**	6460.4	-	-	-
70,000	**214.6**	8804.3	-	-	-
80,000	**268.2**	11,170.6	-	-	-
90,000	**320.0**	13,710.0	-	-	-
100,000	**363.4**	15,650.1	-	-	-

From TABLE 10 and Fig.7, we can conclude that DROP3 runtime grows the most, the runtime growth rates

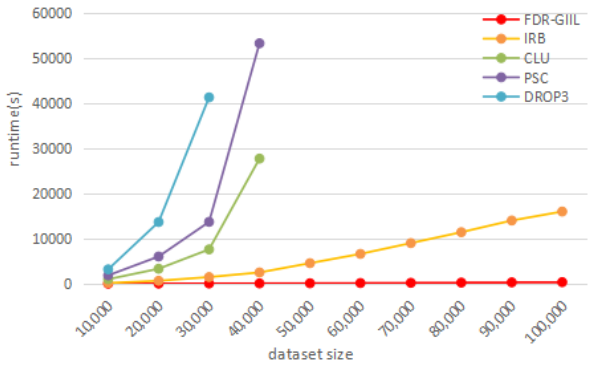


FIGURE 7. The runtime rising tendency of each algorithm with the growth of the data size.

of CLU and PSC are similar to DROP3, and all of them are unfeasible when the data size is larger than 40,000. As the dataset size increases, the runtime of IRB is growing more slowly than DROP3, CLU and PSC, but faster than our algorithm. The proposed algorithm FDR-GIIL is clearly the fastest data reduction algorithm.

To sum up, these experimental results show that the proposed data reduction method outperforms the current popular algorithms in smaller computational cost, and higher classification accuracy under the same reduction rate by using the mapping based granulation and importance labeling.

V. CONCLUSIONS

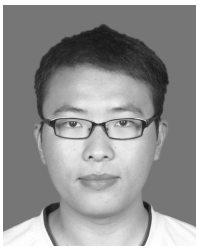
Data reduction is important for instance based learning on Big Data. Such reduction can enhance the efficiency of classification and reduce the storage requirement. In this paper, a fast data reduction with granulation based instances importance labeling is proposed. First, we granulate the original dataset based on a simple mapping and K-means, then label the instance importance in each granule based on the calculation of Hausdorff distance, and finally filter unimportant instances according to the importance and crowding degree to reduce the original data size. The superior performance of the proposed algorithm in maintaining the instances distribution, fast computing and keeping higher classification accuracy is experimentally demonstrated by using it to 18 datasets. Especially, the proposed algorithm is outstanding for removing unimportant instances of large scale datasets. In the future, feature reduction will be combined with the instance selection for improving the efficiency of classification.

REFERENCES

[1] M. R. Bendre and V. R. Thool, "Analytics, challenges and applications in big data environment: A survey," *J. Manage. Anal.*, vol. 3, no. 3, pp. 206–239, Jul. 2016.
[2] P. Guo, K. Wang, A-L. Luo, and M. Xue, "Computational intelligence for big data analysis: Current status and future prospect," *J. Softw.*, vol. 26, no. 11, pp. 3010–3025, Nov. 2015.
[3] F.-Z. Benjelloun, A. A. Lahcen, and S. Belfkih, "An overview of big data opportunities, applications and tools," in *Proc. IEEE ISCV Conf.*, Fez, Morocco, Mar. 2015, pp. 1–6.
[4] C. L. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Inf. Sci.*, vol. 275, pp. 314–347, Aug. 2014.
[5] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 97–107, Jan. 2014.
[6] P. R. Anukrishna and V. Paul, "A review on feature selection for high dimensional data," in *Proc. Int. Conf. Inventive Syst. Control*, Coimbatore, India, Jan. 2017, pp. 1–4.
[7] Á. Arnaiz-González, J.-F. Díez-Pastor, J. J. Rodríguez, and C. García-Osorio, "Instance selection of linear complexity for big data," *Knowl.-Based Syst.*, vol. 107, pp. 83–95, Sep. 2016.
[8] S. Ougiaroglou and G. Evangelidis, "RHC: A non-parametric cluster-based data reduction for efficient k-NN classification," *Pattern Anal. Appl.*, vol. 19, no. 1, pp. 93–109, Feb. 2016.
[9] J. Liang, Y. Qian, and D. Li, "Research development on granular computing theory and method for big data," *Big Data Res.*, vol. 2, no. 4, pp. 13–23, Jul. 2017.
[10] J. Niu, C. Huang, J. Li, and M. Fan, "Parallel computing techniques for concept-cognitive learning based on granular computing," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 11, pp. 1785–1805, Nov. 2018.
[11] X. Zhang, C. Mei, D. Chen, and Y. Yang, "A fuzzy rough set-based feature selection method using representative instances," *Knowl.-Based Syst.*, vol. 151, no. 1, pp. 216–229, Jul. 2018.
[12] J. Xu, G. Y. Wang, and H. Yu, "Review of big data processing based on granular computing," *Chin. J. Comput.*, vol. 38, no. 8, pp. 1497–1517, Aug. 2015.
[13] Y. Song, J. Liang, J. Lu, and X. Zhao, "An efficient instance selection algorithm for k nearest neighbor regression," *Neurocomputing*, vol. 251, pp. 26–34, Aug. 2017.
[14] J. Hamidzadeh, R. Monsefi, and H. S. Yazdi, "Large symmetric margin instance selection algorithm," *Int. J. Mach. Learn. Cybern.*, vol. 7, no. 1, pp. 25–45, Feb. 2016.
[15] P. Hart, "The condensed nearest neighbor rule," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 3, pp. 515–516, May 1968.
[16] C.-H. Chou, B.-H. Kuo, and F. Chang, "The generalized condensed nearest neighbor rule as a data reduction method," in *Proc. Int. Conf. Pattern Recognit.*, Hong Kong, Aug. 2006, pp. 556–559.
[17] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-2, no. 3, pp. 408–421, Jul. 1972.
[18] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," *Mach. Learn.*, vol. 38, no. 3, pp. 257–286, 2000.
[19] E. Leyva, A. González, and R. Pérez, "Three new instance selection methods based on local sets: A comparative study with several approaches from a bi-objective perspective," *Pattern Recognit.*, vol. 48, no. 4, pp. 1523–1537, Apr. 2015.
[20] M. Suganthi and V. Karunakaran, "Instance selection and feature extraction using cuttlefish optimization algorithm and principal component analysis using decision tree," *Cluster Comput.*, vol. 1, no. 2, pp. 1–13, Jan. 2018.
[21] A. Lumini and L. Nanni, "A clustering method for automatic biometric template selection," *Pattern Recognit.*, vol. 39, no. 3, pp. 495–497, Mar. 2006.
[22] J. A. Olvera-López, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A new fast prototype selection method based on clustering," *Pattern Anal. Appl.*, vol. 13, no. 2, pp. 131–141, 2010.
[23] C. G. Vallejo, J. A. Troyano, and F. J. Ortega, "InstanceRank: Bringing order to datasets," *Pattern Recognit. Lett.*, vol. 31, no. 2, pp. 133–142, Jan. 2010.
[24] P. Hernandez-Leal, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, and J. A. Olvera-Lopez, "InstanceRank based on borders for instance selection," *Pattern Recognit.*, vol. 46, no. 1, pp. 365–375, Jan. 2013.
[25] J. L. Carbonera and M. Abel, "A density-based approach for instance selection," in *Proc. IEEE Int. Conf. Tools Artif. Intell.*, Vietri sul Mare, Italy, Nov. 2015, pp. 768–774.
[26] F. Zhang, B. Liu, and H. Yan, "Rough decision rules extraction and reduction based on granular computing," *J. Commun.*, vol. 37, no. Z1, pp. 30–35, Oct. 2016.
[27] R. Sikora and S. Piramuthu, "Efficient genetic algorithm based data mining using feature selection with Hausdorff distance," *Inf. Technol. Manage.*, vol. 6, no. 4, pp. 315–331, Oct. 2005.
[28] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.
[29] D. Dua and E. K. Taniskidou, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[30] S. T. Hadjitodorov, L. I. Kuncheva, and L. P. Todorova, "Moderate diversity for better cluster ensembles," *Inf. Fusion*, vol. 7, no. 3, pp. 264–275, Sep. 2006.

XIAOYAN SUN received the Ph.D. degree in control theory and control engineering from the China University of Mining and Technology, Xuzhou, China, in 2009, where she is currently a Professor with the School of Information and Control Engineering. Her current research interests include interactive evolutionary computation, intelligent data processing, big data, and intelligence optimization.

LIAN LIU was born in 1992. She received the B.S. degree in automation from Changzhou University, in 2016. She is currently pursuing the M.S. degree in control engineering with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China. Her research interests include data reduction and machine learning.



CONG GENG was born in 1991. He received the B.S. degree in mathematics and applied mathematics from the China University of Mining and Technology, Xuzhou, China, in 2016, where he is currently pursuing the M.S. degree in control engineering with the School of Information and Control Engineering. His research interests include data reduction and machine learning.

SHAOFENG YANG was born in 1966. He is currently an Instructor with the China University of Mining and Technology. His research interests include mechanical and electrical integration and power transmission and control.