Zhai and Song *Journal of Big Data* (2022) 9:87
https://doi.org/10.1186/s40537-022-00640-0

Journal of Big Data

RESEARCH Open Access

Optimal instance subset selection from big data using genetic algorithm and open source framework

Junhai Zhai* and Dandan Song

*Correspondence:
mczjh@126.com

Hebei Key Laboratory of Machine Learning and Computational Intelligence, College of Mathematics and Information Science, Hebei University, Baoding, China

Abstract

Data is accumulating at an incredible rate, and the era of big data has arrived. Big data brings great challenges to traditional machine learning algorithms, it is difficult for learning tasks in big data scenario to be completed on stand-alone. Data reduction is an effective way to solve this problem. Data reduction includes attribute reduction and instance reduction. In this study, we focus on instance reduction also called instance selection, and view the instance selection as an optimal instance subset selection problem. Inspired by the ideas of cross validation and divide and conquer, we defined a novel criterion called combined information entropy with respect to a set of classifiers to measure the importance of an instance subset, the criterion uses multiple independent classifiers trained on different subsets to measure the optimality of an instance subset. Based on the criterion, we proposed an approach which uses genetic algorithm and open source framework to select optimal instance subset from big data. The proposed algorithm is implemented on two open source big data platforms Hadoop and Spark, the conducted experiments on four artificial data sets demonstrate the feasibility of the proposed algorithm and visualize the distribution of selected instances, and the conducted experiments on four real data sets compared with three closely related methods on test accuracy and compression ratio demonstrate the effectiveness of the proposed algorithm. Furthermore, the two implementations on Hadoop and Spark are also experimentally compared. The experimental results show that the proposed algorithm provides excellent performance and outperforms the three methods.

Keywords: Big data, Instance selection, Cross-selection, Genetic algorithm, Open source platforms

Introduction

In machine learning, instance selection is to select a subset from a training set such that there is little or no performance degradation training a learning system with the selected subset. The condensed nearest neighbor (CNN) [1] proposed by Hart is the first instance selection algorithm to reduce the computational complexity of 1-nearest neighbor (1-NN). The core concept of CNN is consistent subset, which can correctly classify all instances in the training set with 1-NN. The goal of CNN is to find the

Springer Open

© The Author(s) 2022. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

minimal consistent subset of the training set. However, the consistent subset found by CNN may not be the smallest. To deal with this drawback, Gates [2] proposed reduced nearest neighbor (RNN). Based on the relative significance of the instances in training set, Dasarathy [3] proposed another algorithm to find the minimal consistent subset of the training set. In addition, CNN is sensitive to noise. To this end, Wilson and Martinez [4] proposed edited nearest neighbor (ENN). Based on CNN, the researchers also proposed some other improved algorithms. For instance, Brighton and Mellish [5] introduced the concepts of reachable set and coverage set into CNN and proposed an iterative case filtering algorithm. Angiulli [6] introduced the idea of Voronoi partition into CNN and proposed fast CNN. Li and Maguire [7] proposed a critical pattern selection algorithm by considering local geometrical and statistical information, which selects both border and edge instances from training set. Hernandez-Leal et al. [8] introduced an instance ranking per class using borders, and proposed an instance selection algorithm using the ranking information. Cavalcanti and Soares [9] also proposed a ranking based instance selection algorithm. Different from [8], the algorithm calculates a score for each instance given its relations with other instances in the training set, and then selects instances according to the score.

The algorithms mentioned above are all *k*-nearest neighbor based methods, and the researchers have also proposed instance selection methods based on other learning algorithms. Liu et al. [10] proposed an efficient self-adaption instance selection algorithm reconstructing training set for support vector machine from the viewpoint of geometry. Aslani and Seipel [11] adopted locality-sensitive hashing for developing an instance selection method which rests on rapidly finding similar and redundant training instances and excluding them from the training set. Based on clustering technique, Chen et al. [12] proposed an instance selection algorithm for speeding up support vector machines. Ant colony optimization (ACO) algorithm performs two primary functions: boundary detection and boundary instance selection. Based on ACO, Akinyelu et al. [13] proposed an instance selection algorithm for SVM speed optimization. Shao et al. [14] combined instance and feature selection, and proposed an uniform sparse primal and dual LSSVM model. Furthermore, Du et al. [15] found that if the feature and instance selection are addressed separately, the irrelevant features may mislead the process of instance selection. In order to deal with this problem, they proposed a unified framework, which selects instances and features simultaneously. Liaw [16] proposed a framework for cooperative evolutionary learning and instance selection in an adaptive manner, and an effective data evaluation of representativeness of promising solution for evolutionary instance selection. Chen et al. [17] proposed a sample selection method based on genetic algorithm (GA). Arnaiz-González et al. [18] proposed a new technique for instance selection and noise filtering for regression, the technique uses instance selection for classification after output value discretization, and demonstrates well robustness to noise. In addition, Arnaiz-González et al. [19] also proposed fusion of instance selection algorithms for regression tasks to improve the selection performance. Malhat et al. [20] proposed a sample selection algorithm based on global probability density and correlation functions. Czarnowski [21] proposed a sample selection algorithm based on cluster analysis.

Most of the above instance selection algorithms are only applicable to small and medium data sets since the entire data set needs to be loaded into memory when selecting instances from a data set. However, these algorithms may become infeasible for big data, when the size of the training set far exceeds the memory capacity of the computer. There have been few studies on instance selection for big data, and only a few researchers have explored this topic. Based on the locally sensitive hashing technique, Arnaiz-González et al. [22] proposed an instance selection algorithm for large data with linear computational time complexity. In addition, they extended the instance selection algorithm based on democratic distance to the big data environment and implement the algorithm with MapReduce [23]. In the framework of instance reduction, Triguero et al. [24] proposed a MapReduce based instance reduction method, which can conduct instance reduction on big data, thus realizing the classification of big data. Based *k*-nearest neighbor graph, Mall et al. [25] propose a method to select representative subsets from big data set to realize the learning of big data. Based on random mutation hill climbing and MapReduce, Si et al. [26] proposed a big data instance selection algorithm. Inspired by the idea of cross validation and divide and conquer, we proposed an approach to select optimal instance subset from big data using genetic algorithm and open source framework. The main contributions of this paper lie in the following three folds.

1. We defined a novel criterion which combines information entropies with respect to multiple classifiers to measure the importance of an instance subset. Because multiple independent classifiers are used to evaluate the optimality of an instance subset, the evaluation is consistent with human cognition and is more reasonable.
2. Based on the criterion, and inspired by the ideas of cross validation and divide and conquer, we proposed an approach to select optimal instance subset from big data using genetic algorithm and open source framework.
3. We implemented the proposed algorithm on two open source big data frameworks, Hadoop[1] and Spark.[2] Experiments on four artificial data sets demonstrate the feasibility of the proposed algorithm and visualize the distribution of selected instances. Experiments on four real data sets compared with three closely related methods on test accuracy and compression ratio demonstrate the effectiveness of the proposed algorithm.

The rest of this paper is organized as follows. In "Extreme learning machine" section, we present the preliminaries used in this paper. In "The proposed algorithm" section, we describe the details of the proposed approach. In "Experimental results and analysis" section, the experiments are conducted to demonstrate the feasibility and effectiveness of the proposed approach. At last, we conclude our work in "Conclusions" section.

[1] https://hadoop.apache.org/.

[2] http://spark.apache.org/.

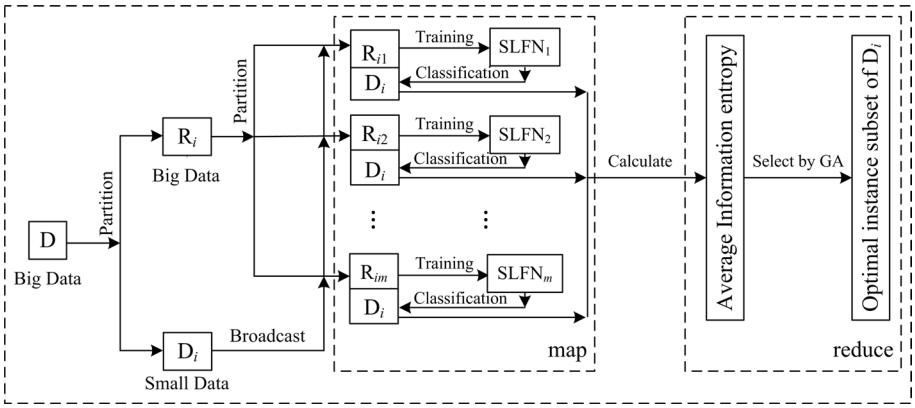

Fig. 1 The technical route of the *i*th round for selecting instances

Extreme learning machine

In this section, we will briefly review the extreme learning machine (ELM) [27] that will be used in the proposed method.

ELM is an algorithm for training random weighted network that is a single hidden layer feedforward neural network (SLFN) with a special architecture. The particularity of the architecture of SLFN is that the activation function of all nodes in the input layer is linear function $y = x$, that of all nodes in the output layer is also linear function $y = x$, and that of all nodes in the hidden layer is sigmoid function $y = \frac{1}{1+e^{-x}}$. In addition, all nodes of the hidden layer have bias, while all nodes of the output layer have no bias. The weights $\boldsymbol{w}_i$ between the input layer and the hidden layer and the biases b_i of the nodes of the hidden layer are randomly assigned, where $1 \leq i \leq m$ and m is the number of the nodes of the hidden layer. The weights $\boldsymbol{\beta}_i$ between the hidden layer and the output layer are obtained by solving the following optimization problem:

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{H}\boldsymbol{\beta} - \boldsymbol{Y}\| \tag{1}$$

where $\boldsymbol{Y}$ is the expected output matrix, $\boldsymbol{H}$ is the output matrix of hidden layer.

Given a training set $S = \{(\boldsymbol{x}_i, y_i) | 1 \leq i \leq n\}$, the smallest norm least-squares solution of Eq. (1) can be obtained by Eq. (2).

$$\boldsymbol{\beta} = \boldsymbol{H}^{\dagger}\boldsymbol{Y} \tag{2}$$

where $\boldsymbol{H}^{\dagger}$ is the Moore–Penrose generalized inverse of $\boldsymbol{H}$, and $\boldsymbol{H}^{\dagger} = \left(\boldsymbol{H}\boldsymbol{H}^{T}\right)^{-1}\boldsymbol{H}$.

Since the training of SLFN with ELM does not need to iteratively adjust weight parameters, its learning speed is very fast. In addition, ELM has good generalization performance [27], it has been widely used in classification and regression [28–31], pattern recognition [32–34], and data reduction [35–40].

The proposed algorithm

In this section, we present the details of the proposed algorithm. An overview of the proposed method is shown in Fig. 1. First, the big data set is partitioned into k subsets, and k-1 subsets are used to train classifiers via MapReduce. Then, the trained classifiers are used to classify instances from the remaining subset. Finally, genetic

algorithm is used to select informative instances that are more likely being misclassified by these classifiers from the remaining subset. This section is organized as follows, details of classifier training and calculating information entropy of subset are introduced in "Training classifiers and calculating information entropy of subset" section, Optimal subset selection using genetic algorithm is introduced in "Optimal subset selection via genetic algorithm" section, and computational time complexity is analyzed in "Computational time complexity" section.

Training classifiers and calculating information entropy of subset

For the convenience of description, we present the details of the proposed algorithm in the framework of Hadoop MapReduce. Since the algorithm is inspired by the idea of k-fold cross validation, it is a k-round iterative algorithm. Similar to the method of k-fold cross validation, we first partition a big data set D into k subsets, $D_1, D_2, \ldots, D_k$, and then we use the genetic algorithm to select informative instances from a subset $D_i (1 \leq i \leq k)$.

In the following, we present the details of ith round for selecting informative instances from D_i with genetic algorithm. Let $R_i = D - D_i$. Obviously, R_i is a big data set. In the ith round, R_i is automatically partitioned into m splits by map mechanism of MapReduce, and the m splits are deployed to m map computing nodes, where m is the number of map nodes in a big data computing platform. In addition, the subset D_i is broadcast to m map nodes.

On the m map nodes, complete the following works in parallel:

1. Training SLFN classifier by ELM

   The m SLFN classifiers are trained in parallel at m nodes on the local splits $R_{ij} (1 \leq j \leq m)$ by ELM algorithm. The reason why we choose ELM algorithm to train SLFN is that it has a very fast learning speed and excellent generalization ability [27]. The m SLFN classifiers denoted by $\mathrm{SLFN}_1, \mathrm{SLFN}_2, \ldots, \mathrm{SLFN}_m$ form a committee.
2. Classifying the instances in D_i by the trained SLFNs
   Given an instance $\boldsymbol{x} \in D_i$, the classification result of $\boldsymbol{x}$ by a SLFN on a map node can be transformed into a posterior probability distribution by the Eq. (3).

$$p(\omega_l | \boldsymbol{x}; \mathrm{SLFN}_j) = \frac{e^{y_i}}{\sum_{l=1}^{L} e^{y_l}} \tag{3}$$

   In Eq. (3), ω_l stands for the lth class, L is the number of classes.
3. Calculating the information entropy of the instances in D_i
   The information entropy of instance $\boldsymbol{x}$ with respect to classifier SLFN_j is defined by Eq. (4).

$$E(\boldsymbol{x}; \mathrm{SLFN}_j) = -\sum_{l=1}^{L} p(\omega_l | \boldsymbol{x}; \mathrm{SLFN}_j) \log_2 p(\omega_l | \boldsymbol{x}; \mathrm{SLFN}_j) \tag{4}$$

On a reduce node, complete the following works:

1. Calculating the average information entropy of the instances in D_i with respect to the committee.

   The average information entropy of instance $\boldsymbol{x}$ in D_i with respect to the committee is defined by Eq. (5).

$$\begin{aligned} AVE(\boldsymbol{x}) &= \frac{1}{m}\sum_{j=1}^{m} E(\boldsymbol{x}; \mathrm{SLFN}_j) \\ &= -\frac{1}{m}\sum_{j=1}^{m}\sum_{l=1}^{L} p(\omega_l|\boldsymbol{x}; \mathrm{SLFN}_j)\log_2 p(\omega_l|\boldsymbol{x}; \mathrm{SLFN}_j) \end{aligned} \tag{5}$$

2. Calculating the information entropy of a subset of D_i
   Given a subset $Q \subseteq D_i$, the information entropy of Q is defined by Eq. (6).

$$\begin{aligned} E(Q) &= \frac{1}{|Q|}\sum_{\boldsymbol{x}\in Q} AVE(\boldsymbol{x}) \\ &= -\frac{1}{m|Q|}\sum_{\boldsymbol{x}\in Q}\sum_{j=1}^{m}\sum_{l=1}^{L} p(\omega_l|\boldsymbol{x}; \mathrm{SLFN}_j)\log_2 p(\omega_l|\boldsymbol{x}; \mathrm{SLFN}_j) \end{aligned} \tag{6}$$

3. Selecting the optimal subset of D_i by genetic algorithm. The details of selecting optimal subset using genetic algorithm with Eq. (6) is presented in the next section.

Optimal subset selection via genetic algorithm

In this section, we demonstrate how the optimal instance subset is selected. The optimal instance subset consists of a number of informative instances. It is believed that hard instances that are more likely being misclassified by the classifiers are more informative to the classifiers. The reason is that for classifiers, easy instances that are more likely being correctly classified by the classifiers cannot contribute much to the losses of the classifiers, therefore do not contribute to the training of the classifiers. The misclassified instances are more informative and contribute more to the losses, and the classifiers learn most from the incorrectly classified instances and in turn adjust the decision boundaries.

A naive approach is to select hard instances using hard thresholds, e.g. selecting instances with scores above or below a threshold or selecting top-k instances. However, it is very hard to find a global hard threshold which works well for all subset partitions. Therefore, we propose to search dynamic thresholds using genetic algorithm (GA) [41–43]. GA uses population search strategy to find the optimal solution of the addressed problem. A population consist of some individuals, and an individual is a candidate solution which is usually encoded as a character string. There are three key issues for instance selection using genetic algorithm, which are (1) the encoding of candidate solutions of the addressed problem, (2) the design of fitness function and (3) genetic manipulations.

Regarding to the issue (1). Given an individual $Q \in P$, namely, a subset of D_i, P is a population. It is suitable to encode Q by 0 and 1. If $|D_i| = n_i (1 \leq i \leq k)$, then it is obvious that the length of binary strings used for representing Q is n_i.

Regarding to the issue (2). Given an individual $Q \in P$, we use Eq. (6) as the fitness function to evaluate the goodness of Q. As discussed before, the goal is to select hard instances which are more likely being misclassified by the classifier. And Eq. (6) can naturally measure how confident the classifier is or how likely each instance is being misclassified by the classifier. Larger entropy values indicate less confidence and more errors. To sum up, the larger $E(Q)$ is, the harder Q is, therefore the more important Q is.

In genetic algorithm, the genetic manipulations include selection, crossover and mutation. We use roulette wheel method to select two parent individuals from a population according to their fitness (the better fitness, the bigger chance to be selected). The roulette wheel method include the following four steps:

1. Calculate selection probability. For each individual Q_i in the population P, let $|P| = N$. The selection probability of Q_i is calculated by $p_i = \frac{E(Q_i)}{\sum_{j=1}^{N} E(Q_j)}$. Obviously, it is true that $\sum_{i=1}^{N} p_i = 1$.
2. Calculate cumulative probability. For each individual $Q_i \in P$, the cumulative probability of Q_i is calculated by $q_i = \sum_{j=1}^{i} p_j$.
3. Partition [0, 1] into N intervals, such that the length of ith interval is $p_i (1 \leq i \leq N)$.
4. Select N individuals. Repeat the following operations N times: generate a random number $r \in [0, 1]$, if $r \leq q_1$, then select the individual Q_1; Otherwise, if $q_{i-1} \leq r \leq q_i$, then select the individual Q_i.

We use one-point crossover with a crossover probability p_c to cross over the parents to form new offsprings, and we randomly select a position in an individual with a mutation probability p_m to mutate (i.e. change 1 to 0, or change 0 to 1) to form a new individual. The pseudo-code of the algorithm for instance selection by genetic algorithm is given in algorithm 1.

Algorithm 1: The algorithm for instance selection by genetic algorithm

Input: D_i, a subset of D; N, the population size; p_s, selection probability; p_c, crossover probability; p_m, mutation probability.
Output: $S_i \subseteq D_i$.

```
Randomly initialize a population of size N;
repeat
    for (i = 1; i ≤ N; i = i + 1) do
        Calculate the fitness of Q_i by (6);
    end
    for (i = 1; i ≤ N; i = i + 1) do
        Generate a random number r ∈ [0, 1];
        if (r ≤ q_1) then
            Select Q_1;
        end
        else
            if (q_{i-1} ≤ r ≤ q_i) then
                Select Q_i;
            end
        end
    end
    Select individuals by p_c, and conduct crossover;
    Select individuals by p_m, and conduct mutation;
until (the termination condition is met);
Output S_i.
```

Computational time complexity

In this section, we analyze the computational time complexity of the proposed algorithm. At each map node, the main operations include (1) Training a SLFN classifier by

ELM algorithm, (2) Classifying the instances in D_i by the trained SLFN classifier, and (3) Calculating the information entropy of the instances in D_i by Eq. (4). The computational time complexity of training SLFN by ELM is $O(m^2n)$ [44], where m is the number of hidden nodes of SLFN, n is the number of instances in training set. The computational time complexity of classifying the instances in D_i by the trained SLFN classifier is $O(n_i)$, where n_i is the number of instances in D_i. The computational time complexity of calculating the information entropy of the instances in D_i is $O(L \times n_i)$. Accordingly, the computational time complexity of operations at one map node is $O(m^2n) + O(n_i) + O(Ln_i)$. Since usually $L \ll n_i$ and $n_i \simeq \frac{1}{k}n$, we have $O(m^2n) + O(n_i) + O(Ln_i) = O(m^2n)$. Totally, the computational time complexity of operations at k map nodes is $O(km^2n)$.

At reduce node, the main operations include (1) calculating the average information entropy of the instances in D_i with respect to the committee by Eq. (5), (2) calculating the information entropy of a subset of D_i by Eq. (6), and (3) selecting the optimal subset of D_i by genetic algorithm. From formula Eq. (5) and Eq. (6), we can find that the computational time complexities of calculating the average information entropy of the instances in D_i and of calculating the information entropy of a subset of D_i are $O(Lm)$ and $O(Lmn_i)$, respectively. From the algorithm 1, we can find that the computational time complexities of selecting the optimal subset of D_i by genetic algorithm mainly determined by the first for loop, its computational time complexity is $O(NLmn_i)$. Accordingly, the computational time complexity of operations at reduce node is $O(Lm) + O(Lmn_i) + O(NLmn_i)$. Obviously, $O(Lm) + O(Lmn_i) + O(NLmn_i) = O(mn)$. Hence, the computational time complexity of the proposed algorithm is $O(km^2n) + O(mn) = O(km^2n)$.

Experimental results and analysis

We conducted two experiments to demonstrate the feasibility and effectiveness of the proposed approach respectively.

Experiment 1: demonstrating the feasibility of the proposed algorithm

In this section, we conduct experiments on four artificial data sets and visualize the selected instances to verify the feasibility of the proposed algorithm. The first and the second artificial data sets have clear classification boundaries between different categories. While the third and fourth artificial data sets do not have clear classification boundaries between different categories, i.e., there is overlap between the instances belonging to different classes.

The first artificial data set Circle is a two-dimensional data set, which consists of 1000 data points belonging to two classes, 500 data points per class. The points of the class 1 are uniformly distributed into a circle of radius 0.3 centered on (0.5, 0.5). The points of the class 2 are uniformly distributed into a ring centered on (0.5, 0.5) with internal and external radii equal to 0.3 and 0.5, respectively. The distribution of the instances in the artificial data set Circle, and the distribution of the instances selected from the data set Circle by the proposed algorithm are shown on the left and right of Fig. 2 respectively.

The second artificial data set Square consists of four 2000 data points belonging to four classes. The points of the class 1 are uniformly distributed in a square centered on (0.5, 0.5) and with length 1.0; The points of the class 2 are uniformly distributed in a square centered on $(-0.5, 0.5)$ and with length 1.0; The points of the class 3 are

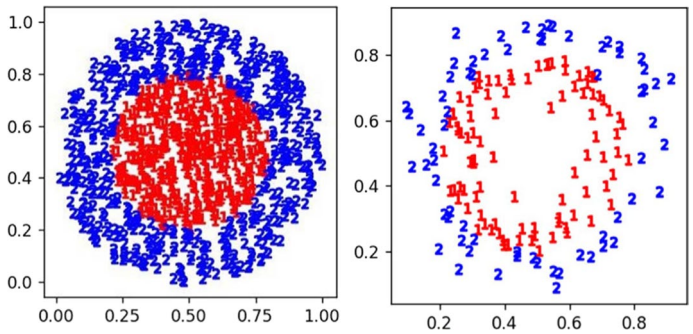

Fig. 2 The distribution of the instances in the artificial data set Circle (left), and the distribution of the instances selected from the artificial data set Circle (right)

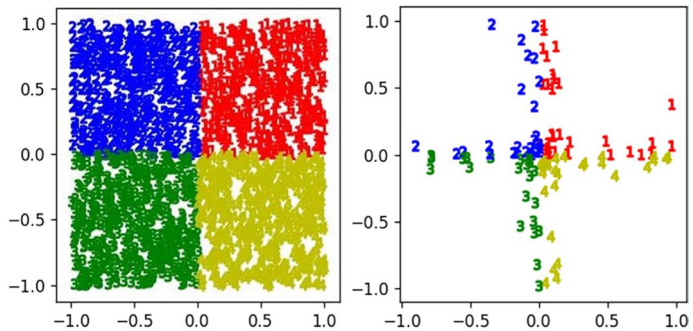

Fig. 3 The distribution of the instances in the artificial data set Square (left), and the distribution of the instances selected from the artificial data set Square (right)

Table 1 The mean vectors and covariance matrices of three Gaussian distributions in Gaussian1

i	μ_i	Σ_i
1	$(0.0, 0.0)^T$	$\begin{bmatrix} 0.7 & 0.0 \\ 0.0 & 0.7 \end{bmatrix}$
2	$(1.0, 1.0)^T$	$\begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$
3	$(-1.0, 1.0)^T$	$\begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$

uniformly distributed in a square centered on $(-0.5, -0.5)$ and with length 1.0; The points of the class 4 are uniformly distributed in a square centered on $(0.5, -0.5)$ and with length 1.0. The distribution of the instances in the artificial data set Square, and the distribution of the instances selected from the data set Square by the proposed algorithm are shown on the left and right of Fig. 3 respectively.

The third artificial data set Gaussian1 is a two-dimensional data set with three classes followed three Gaussian distributions, it contains 1500 data points, 500 data points per class. The mean vectors and covariance matrices of the three Gaussian distributions are given in Table 1. The distribution of the instances in the artificial data set Gaussian1, and the distribution of the instances selected from the data

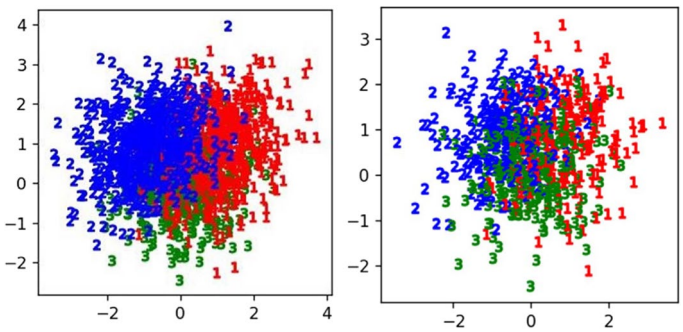

Fig. 4 The distribution of the instances in the artificial data set Gaussian1 (left), and the distribution of the instances selected from the artificial data set Gaussian1 (right)

Table 2 The mean vectors and covariance matrices of three Gaussian distributions in Gaussian2

i	μ_i	Σ_i
1	$(0.0, 0.0, 0.0)^T$	$\begin{bmatrix} 3.0 & 0.0 & 0.0 \\ 0.0 & 5.0 & 0.0 \\ 0.0 & 0.0 & 2.0 \end{bmatrix}$
2	$(1.0, 5.0, -3.0)^T$	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 4.0 & 1.0 \\ 0.0 & 1.0 & 6.0 \end{bmatrix}$
3	$(0.0, 0.0, 0.0)^T$	$\begin{bmatrix} 10.0 & 0.0 & 0.0 \\ 0.0 & 10.0 & 0.0 \\ 0.0 & 0.0 & 10.0 \end{bmatrix}$

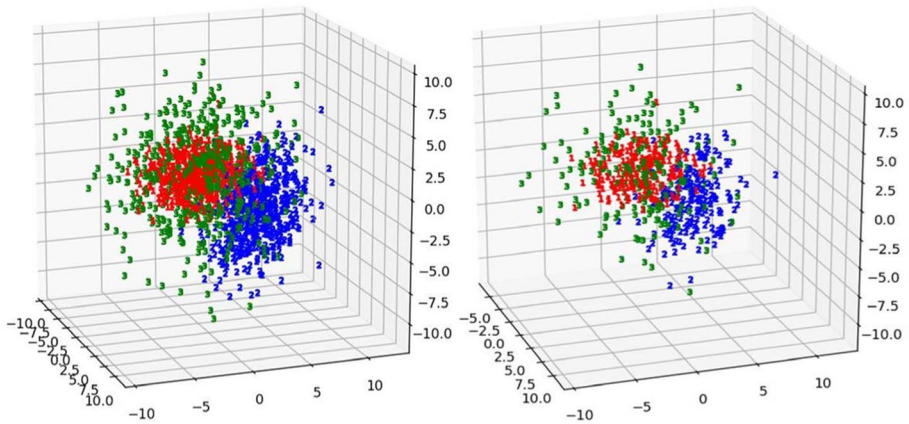

Fig. 5 The distribution of the instances in the artificial data set Gaussian2 (left), and the distribution of the instances selected from the artificial data set Gaussian2 (right)

set Gaussian1 by the proposed algorithm are shown on the left and right of Fig. 4 respectively.

The fourth artificial data set Gaussian2 is a three-dimensional data set with three classes followed three Gaussian distributions, it contains 1500 data points, 500 data points per class. The mean vectors and covariance matrices of the three Gaussian

Table 3 The experimental comparison of the test accuracies of the classifiers trained on the four original artificial data sets and on the four selected subsets

Data sets	Test accuracy1	Test accuracy2
Circle	**0.972**	0.960
Square	**0.986**	0.973
Gaussian1	0.701	**0.733**
Gaussian2	0.733	**0.793**

distributions are given in Table 2. The distribution of the instances in the artificial data set Gaussian2, and the distribution of the instances selected from the data set Gaussian2 by the proposed algorithm are shown on the left and right of Fig. 5 respectively.

From the visualized results presented in Figs. 2 and 3, it is found that the data points selected from artificial data sets Circle and Square by the proposed algorithm are distributed near the classification boundary. While from the visualized results presented in Figs. 4 and 5, it is found that the data points selected from artificial data sets Gaussian1 and Gaussian2 by the proposed algorithm are distributed in overlapping areas of different classes. The experimental results are consistent with the instance selection criteria or heuristic given in formula Eq. (6).

In addition, we also compared the test accuracies of the classifiers trained on the four original artificial data sets with that trained on the selected instance subsets, the test accuracies are denoted by test accuracy1 and test accuracy2 respectively. The classifier is a single hidden layer feedforward neural network trained by extreme learning machine. The results of the experimental comparison are listed in Table 3. In Table 3, the bold values indicate the maximum value of Test accuracy1 and Test accuracy2.

From the experimental results listed in Table 3, we can find that the test accuracies of the classifiers trained on the selected instance subsets of Circle and Square are slightly lower than test accuracies of the classifiers trained on the two original data sets, Circle and Square. While the test accuracies of the classifiers trained on the selected instance subsets of Gaussian1 and Gaussian2 are higher than test accuracies of the classifiers trained on the two original data sets, Gaussian1 and Gaussian2. The experimental results verified that the algorithm proposed in this paper is feasible. In the second experiment, we will demonstrate the effectiveness of the proposed algorithm by comparing it with three closely related methods.

Experiment 2: demonstrating the effectiveness of the proposed algorithm

In this section, we demonstrate the effectiveness of the proposed method by comparing it with three closely related methods on test accuracy and compression ratio. The three compared methods are MapReduce based condensed nearest neighbor method (denoted by MR-CNN), Spark based condensed nearest neighbor method (denoted by S-CNN), and locality sensitive hashing based instance selection method (denoted by LCS-IS) [22]. In the following, we first introduce the experimental environment and data sets, and then introduce the parameter setting used, finally present the comparisons with three related methods. In addition, we also compared the two implementations of

Table 4 The configuration of the nodes in the computing platform

Items	Configuration
CPU	Inter Xeon E5-4603 with two cores, 2.0GZ
Memory	16GB
Network card	Broadcom 5720 QP 1Gb
Hard disk	1TB
Operating system	Ubuntu 13.04
Hadoop	Hadoop 2.7.3
Sprk	Spark 2.3.1
JDK	JDK 1.8

Table 5 The basic information of the 4 data sets

Data sets	Number of instances	Number of attributes	Number of classes
Shuttle	58,000	9	7
Poker	1,000,000	10	10
CovType	581,012	54	7
Skin	245,057	3	2

Table 6 The setting of parameters in genetic algorithm and in SLFN

Data sets	Genetic algorithm			SLFN
	N	p_c	p_m	Number of hidden nodes
Shuttle	70	0.70	0.05	50
Poker	130	0.85	0.08	40
CovType	110	0.80	0.06	190
Skin	90	0.75	0.05	25

the proposed algorithm on big data platform Hadoop and Spark in terms of testing accuracy, compression ratio and running time, and visualized the experimental results.

The experimental environment and data sets

The experiment 2 is conducted on a big data platform with 8 computing nodes, the configuration of computing nodes in the big data platform is given in Table 4. It should be noted that in the big data platform, the configuration of the master node and the slave node are same.

The data sets used for experiments include four UCI data sets, the basic information of the four used data sets is given in Table 5.

The parameter settings

The parameter settings include the setting of the number of hidden layer nodes in SLFN, and the setting of parameters in genetic algorithm. First, an ablation study on the

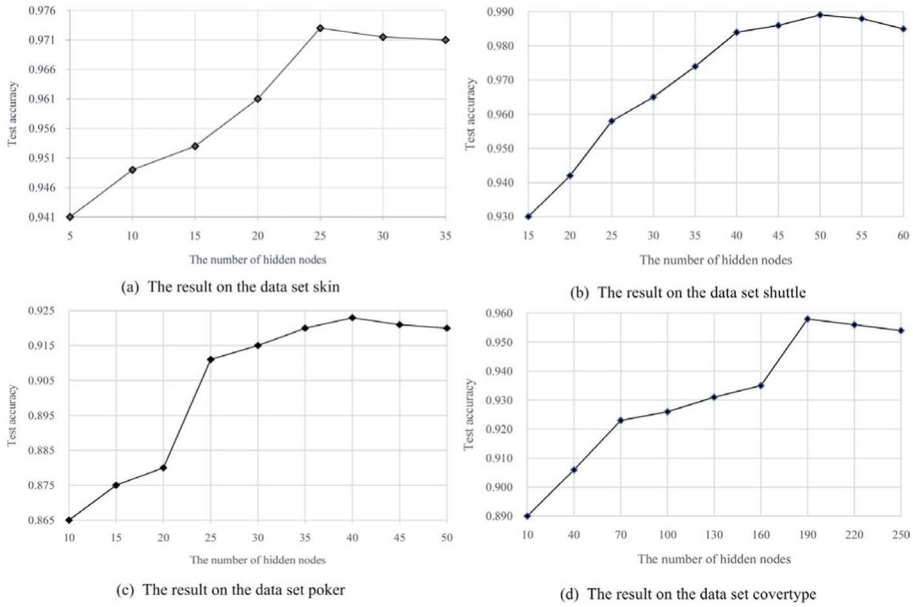

Fig. 6 The relationship between the number of hidden layer nodes and test accuracy

Table 7 The experimental results compared with three related methods on test accuracy

Data sets	MR-CNN	S-CNN	LCS-IS	GA-MR-IS	GA-S-IS
CovType	0.918	0.917	0.920	0.935	**0.941**
Poker	0.877	0.808	0.853	**0.923**	0.919
Shuttle	0.983	0.985	0.987	0.987	**0.989**
Skin	0.906	0.907	0.962	**0.973**	0.969

Table 8 The experimental results compared with three related methods on compression ratio

Data sets	MR-CNN	S-CNN	LCS-IS	GA-MR-IS	GA-S-IS
CovType	1.05	1.08	2.23	**5.07**	4.77
Poker	6.53	6.47	6.64	**8.77**	7.53
Shuttle	1.05	1.06	1.43	**1.83**	1.47
Skin	1.00	1.03	3.88	**5.77**	5.60

number of hidden layer nodes in SLFN is conducted. More specifically, we train SLFN on a subset randomly selected from big data set, and the size of the subset roughly equal to $\frac{n}{m}$, where n is the size of a big data set, and m is the number of nodes in a big data platform. We train different SLFNs with different number of hidden layer nodes, and record the test accuracy of corresponding SLFNs. The results of the ablation study on different data sets are given in Fig. 6. The setting of the number of hidden layer nodes in SLFN is listed in column 5 of the Table 6. Based on our prior knowledge on the parameters in genetic algorithm [45, 46], the setting of parameters in genetic algorithm is listed in the column 2–4 of Table 6. The number of iterations is not set as a parameter, the

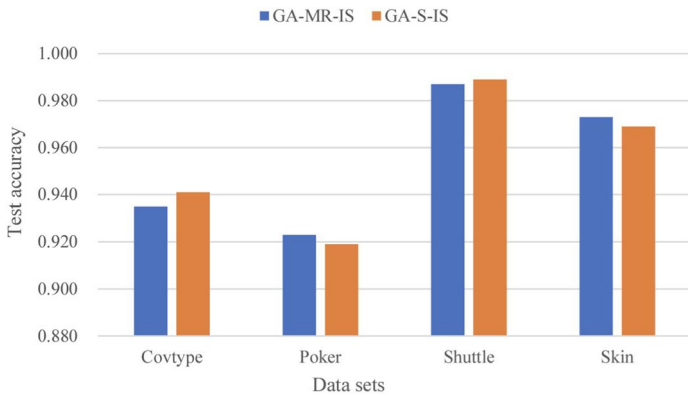

Fig. 7 The comparison of the test accuracy of two implementations

termination condition of the genetic algorithm is adaptive for different data sets. Specifically, when the value of fitness function does not increase, the algorithm will terminate.

The comparison with three related methods

We implemented the proposed approach on two open-source big data platforms, Hadoop and Spark, the two implementations are denoted by GA-MR-IS and GA-S-IS respectively. We experimentally compared GA-MR-IS and GA-S-IS with three related methods on test accuracy and compression ratio, the experimental results are given in Tables 7 and 8 respectively. In Tables 7 and 8, the bold values indicate the maximum accuracy of the five methods.

It can be seen from the experimental results listed in Tables 7 and 8 that, the proposed method is superior to the three compared methods in both the test accuracy and compression ratio. We think the reasons include the following three aspects:

1. The proposed method selects the optimal instance subset, and the optimality is given by an expert committee whose members are independent from the candidate instance set.
2. The proposed method can overcome the following three shortcomings of CNN, while MR-CNN and S-CNN can not.

   - CNN is especially sensitive to noise, because noisy instances will usually be misclassified by their neighbors, and thus will be retained.
   - CNN is also sensitive to the order of instances presented to the algorithm to decide whether or not to select.
   - There are also redundant instances in subset selected by CNN.

3. In the proposed method, we defined a novel criterion to measure importance of instance subset, the measure integrates the wisdom of all the members of a classifier committee. While LCS-IS uses the generated hash function to measure the importance of instances, the measure has high uncertainty.

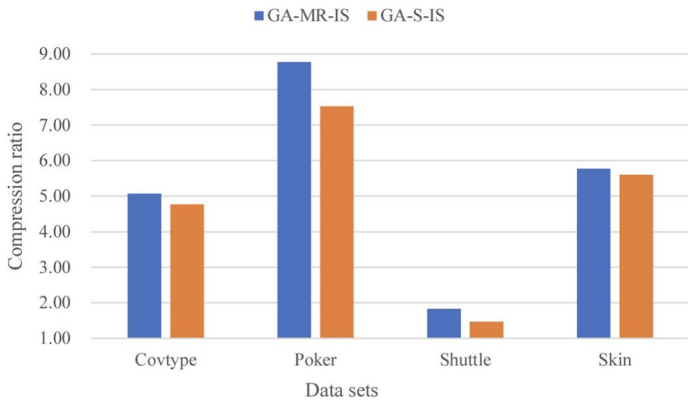

Fig. 8 The comparison of the compression ratio of two implementations

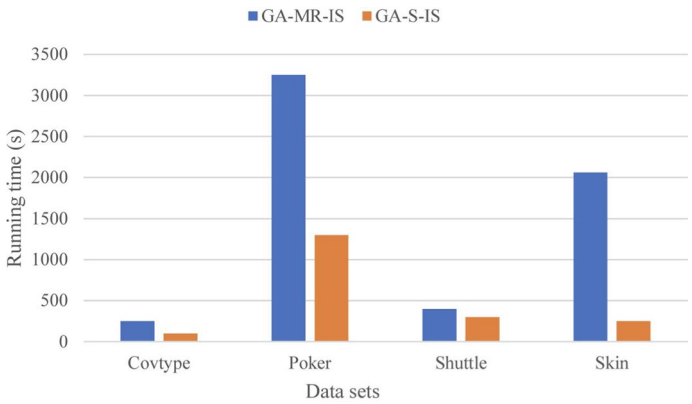

Fig. 9 The comparison of the running times of two implementations

The comparison of two implementations

We also conducted experimental comparison between the two implementations (i.e., GA-MR-IS and GA-S-IS) in terms of testing accuracy, compression ratio and running time, and visualized the experimental results, as shown in Figs. 7, 8 and 9. It can be seen intuitively from the visualization results shown in Figs. 7 and 8 that the two implementations have little difference in test accuracy and compression ratio. The reasons for this are easy to understand, since GA-MR-IS and GA-S-IS have the same mechanism for selecting optimal instance sets. But it can be seen from Fig. 9 that the running times of GA-MR-IS and GA-S-IS are significantly different, this is because MapReduce and Spark use different mechanisms to process big data. MapReduce uses batch processing mechanism to handle big data, while Spark uses memory computing mechanism to handle big data. In Zhai and Huang [47], we analyze the computational time complexity of the two implementations in detail, which can theoretically explain the reason for the significant difference.

Conclusions

Based on the ideas of cross validation and divide and conquer, we defined a novel criterion to measure the importance of an instance subset, and proposed a method based on genetic algorithm and open source framework to select optimal instance subset from big data. Genetic algorithm is adopted to select optimal subset, and the defined criterion is used as fitness function in genetic algorithm. The proposed approach has three advantages: (1) It uses the idea of cross validation to select the optimal instance subset, which can effectively reduce the inductive bias of the selected instance subset. (2) It has a high compression ratio compared with the related methods. (3) The classifier trained on the selected instance subset has competitive generalization performance. The promising future works of this study include (1) proving theoretically that the selected instance subset is optimal; (2) conducting more experiments on more big data sets.

Authors' information

Junhai Zhai is a Professor and Ph.D, Supervisor with College of Mathematics and Information Science, Hebei University, Baoding, China. His main research interests include big data processing, machine learning, and deep learning.Dandan Song is a M.S. candidate of College of Mathematics and Information Science, Hebei University, Baoding, China. Her main research interests include big data processing.

Acknowledgements

We would like to thank Hebei Key Laboratory of Machine Learning and Computational Intelligence for supporting big data computing platform, we also thank Wei Zhang, a graduate student in the Key Laboratory, for his help in this study.

Author contributions

JZ: conceptualization, methodology, resources, investigation, visualization, funding acquisition, supervision, project administration, writing—review and editing. DS: methodology, validation, data preprocessing, visualization. Both authors read and approved the final manuscript.

Funding

This research is supported by the key R &D program of science and technology foundation of Hebei Province (19210310D), and by the natural science foundation of Hebei Province (F2021201020).

Availability of data and materials

Not applicable.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 7 March 2022 Accepted: 27 June 2022
Published online: 05 July 2022

References

1. Hart P. The condensed nearest neighbor rule. IEEE Trans Inf Theory. 1967;14(5):515–6.
2. Gates GW. The reduced nearest neighbor rule. IEEE Trans Inf Theory. 1972;18(3):431–3.
3. Dasarathy BV. Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design. IEEE Trans Syst Man Cybern. 1994;24(1):511–7.
4. Wilson DR, Martinez TR. Reduction techniques for instance-based learning algorithms. Mach Learn. 2000;38(3):257–86.
5. Brighton B, Mellish C. Advances in instance selection for instance-based learning algorithms. Data Min Knowl Discov. 2002;6(2):153–72.
6. Angiulli F. Fast nearest neighbor condensation for large data sets classification. IEEE Trans Knowl Data Eng. 2007;19(11):1450–64.

7. Li YH, Maguire L. Selecting critical patterns based on local geometrical and statistical information. IEEE Trans Pattern Anal Mach Intell. 2011;33(6):1189–201.
8. Hernandez-Leal P, Carrasco-Ochoa JA, Martinez-Trinidad JF, et al. InstanceRank based on borders for instance selection. Pattern Recognit. 2013;46:365–75.
9. Cavalcanti GDC, Soares RJO. Ranking-based instance selection for pattern classification. Expert Syst Appl. 2020;150: 113269.
10. Liu C, Wang W, Wang M, et al. An efficient instance selection algorithm to reconstruct training set for support vector machine. Knowl Based Syst. 2017;116:58–73.
11. Aslani M, Seipel S. A fast instance selection method for support vector machines in building extraction. Appl Soft Comput. 2020;97(Part B):106716.
12. Chen J, Zhang C, Xue X, et al. Fast instance selection for speeding up support vector machines. Knowl Based Syst. 2013;45:1–7.
13. Akinyelu AA, Ezugwu AE. Nature inspired instance selection techniques for support vector machine speed optimization. IEEE Access. 2019;7:154581–99.
14. Shao Y, Li C, Huang L, et al. Joint sample and feature selection via sparse primal and dual LSSVM. Knowl Based Syst. 2019;185: 104915.
15. Du L, Ren X, Zhou P, et al. Unsupervised dual learning for feature and instance selection. IEEE Access. 2020;8:170248–60.
16. Liaw RT. A cooperative coevolution framework for evolutionary learning and instance selection. Swarm Evol Comput. 2021;62: 100840.
17. Chen ZY, Tsai CF, Eberle W, et al. Instance selection by genetic-based biological algorithm. Soft Comput. 2015;19:1269–82.
18. Arnaiz-González Á, Díez-Pastor JF, Rodríguez JJ, et al. Instance selection for regression by discretization. Expert Syst Appl. 2016;54:340–50.
19. Arnaiz-González Á, Blachnik M, Kordos M, et al. Fusion of instance selection methods in regression tasks. Inf Fusion. 2016;30:69–79.
20. Malhat M, Menshawy ME, Mousa H, et al. A new approach for instance selection: algorithms, evaluation, and comparisons. Expert Syst Appl. 2020;149: 113297.
21. Czarnowski I. Cluster-based instance selection for machine classification. Knowl Inf Syst. 2012;30:113–33.
22. Arnaiz-González Á, Díez-Pastor JF, Rodríguez JJ, et al. Instance selection of linear complexity for big data. Knowl Based Syst. 2016;107:83–95.
23. Arnaiz-González Á, González-Rogel Á, Díez-Pastor JF, et al. MR-DIS: democratic instance selection for big data by MapReduce. Progr Artif Intell. 2017;6(3):211–9.
24. Triguero I, Peralta D, Bacardit J, et al. MRPR: a MapReduce solution for prototype reduction in big data classification. Neurocomputing. 2015;150(Part A):331–45.
25. Mall R, Jumutc V, Langone R, et al. Representative Subsets for Big Data learning using K-NN graphs. In: IEEE international conference on big data, 27–30 Oct. Washington, DC; 2014. p. 37–42.
26. Si L, Yu J, Wu WY, et al. RMHC-MR: instance selection by random mutation hill climbing algorithm with MapReduce in big data. Procedia Comput Sci. 2017;111:252–9.
27. Huang GB, Zhu QY, Siew CK. Extreme learning machine: theory and applications. Neurocomputing. 2006;70:489–501.
28. Huang GB, Zhou HM, Ding XJ, et al. Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern Part B. 2012;42(2):513–29.
29. Xie Y, Li Y, Xia Z, et al. An improved forward regression variable selection algorithm for high-dimensional linear regression models. IEEE Access. 2020;8:129032–42.
30. Tsakiris MC, Peng L, Conca A, et al. An algebraic-geometric approach for linear regression without correspondences. IEEE Trans Inf Theory. 2020;66(8):5130–44.
31. Zhu H, Liu H, Fu A. Class-weighted neural network for monotonic imbalanced classification. Int J Mach Learn Cybern. 2021;12:1191–201.
32. Zhang C, Li H, Chen C, et al. Nonnegative representation based discriminant projection for face recognition. Int J Mach Learn Cybern. 2021;12:733–45.
33. Qin Y, Sun L, Xu Y. Exploring of alternative representations of facial images for face recognition. Int J Mach Learn Cybern. 2020;11:2289–95.
34. Wang Z, Zhou X, Wang W, et al. Emotion recognition using multimodal deep learning in multiple psychophysiological signals and video. Int J Mach Learn Cybern. 2020;11:923–34.
35. Kasun LLC, Yang Y, Huang G, et al. Dimension reduction with extreme learning machine. IEEE Trans Image Process. 2016;25(8):3906–18.
36. Wang CZ, Wang Y, Shao MW, et al. Fuzzy rough attribute reduction for categorical data. IEEE Trans Fuzzy Syst. 2020;28(5):818–30.
37. Wang CZ, Huang Y, Shao MW, et al. Feature selection based on neighborhood self-information. IEEE Trans Cybern. 2020;50(9):4031–42.
38. Wang CZ, Huang Y, Shao MW, et al. Fuzzy rough set-based attribute reduction using distance measures. Knowl Based Syst. 2019;164:205–12.
39. Ni P, Zhao SY, Wang XZ, et al. Incremental feature selection based on fuzzy rough sets. Inf Sci. 2020;536:185–204.
40. Ni P, Zhao SY, Wang XZ, et al. PARA: a positive-region based attribute reduction accelerator. Inf Sci. 2019;503:533–50.
41. Mitchell TM. Machine learning. New York: McGraw-Hill Companies, Inc; 2003.
42. Karagoz GN, Yazici A, Dokeroglu T, et al. A new framework of multi-objective evolutionary algorithms for feature selection and multi-label classification of video data. Int J Mach Learn Cybern. 2021;12:53–71.
43. Ge Y, Xin B, Zhou L, et al. Selecting park locations using a genetic algorithm and comprehensive satisfaction. Int J Mach Learn Cybern. 2020;11:1331–8.

44. Zhai JH, Shao QY, Wang XZ. Improvements for P-ELM1 and P-ELM2 pruning algorithms in extreme learning machines. Int J Uncertain Fuzziness Knowl Based Syst. 2016;24(3):327–45.
45. Zhai JH, Liu B, Zhang SF. Feature selection via evolutionary computation based on relative classification information entropy. Pattern Recognit Artif Intell. 2016;29(8):682–90.
46. Zhai JH, Wan B, Wang XZ. Probabilistic tolerance rough set model and its decision risk optimization. J Bioinform Intell Control. 2015;4(2):137–43.
47. Zhai JH, Huang YJ. Instance selection for big data based on locally sensitive hashing and double-voting mechanism. Adv Comput Intell. 2022;2(20):1–10. https://doi.org/10.1007/s43674-022-00033-z.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen journal and benefit from:

► Convenient online submission
► Rigorous peer review
► Open access: articles freely available online
► High visibility within the field
► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com