Feature Selection Based on Graph Structure

Zhiwei Hu[1], Zhaogong Zhang[1](✉), Zongchao Huang[1],
Dayuan Zheng[2](✉), and Ziliang Zhang[3]

[1] School of Computer Science and Technology, Heilongjiang University,
Harbin, China
zhaogong.zhang@qq.com
[2] School of Data Science and Technology, Heilongjiang University,
Harbin, China
1985022@hlju.edu.cn
[3] Department of Computer Science, College of Engineering,
The University of Alabama, Tuscaloosa, USA

Abstract. Feature selection is an important part of data preprocessing. Selecting effective feature subsets can effectively reduce feature redundancy and reduce irrelevant features, and reduce training costs. Based on the theory of feature clusters, this paper proposes a feature selection strategy based on the graph structure. Considering a feature as a node in the graph, using the idea of graph message propagation, integrating the first-order neighbor information of each node, and then selecting the key point of the local maximum score as the selected feature, this can effectively reduce the feature redundancy and reduce features that are not related to the label. Finally, in order to verify the anti-interference of this novel method, the noise dimension was added in the UCI data set, and the comparison test was again performed. The experimental results show that the proposed algorithm can effectively improve the classification accuracy in a specific data set, and the anti-interference is better than other feature selection algorithms.

Keywords: Feature cluster · Feature selection · MPNN · Supervised · Anti-interference

1 Introduction

In the field of data mining and machine learning research, feature dimension reduction is an important part of feature engineering. An effective feature dimension reduction method can accelerate the learning process of the training model and improve the performance of the model. Some studies have shown that, without losing the accuracy of classification, selecting effective feature subsets can suppress the over-fitting of the model, thus improving the universality of the model. In the popular deep learning field, the process of feature selection is hidden during neural network training. But when the

Z. Zhang and D. Zheng—This work was supported by the Natural Science Foundation of Heilongjiang Province (No. F2017024, No. F2017025).

Y. Li et al. (Eds.): COCOA 2019, LNCS 11949, pp. 289–302, 2019.
https://doi.org/10.1007/978-3-030-36412-0_23

amount of data is too large, it is not practical to put the data directly into neural network training. For example, in the analysis of biological genetic information, high-dimensional genetic data, 3 million features, neural network methods can not be directly calculated, at this time must feature selection.

Feature engineering is of great significance in many fields, such as the application of feature engineering in biological information field [1–4]. Feature dimension reduction is an important part of feature engineering. Feature dimension reduction is divided into feature selection and feature extraction. Feature selection selects M features from N features ($N > M$), and no new features are generated; and feature extraction extracts M features from the original N features, which will generate new features. The most typical feature extraction algorithm is the principal component analysis (PCA) algorithm, which selects the features corresponding to the largest M eigenvalues of the covariance matrix as the basis of the projection space, and finally projects the original features into the new space to generate new features. Feature extraction is only suitable for simple classification problems, and is not suitable for special tasks that finding some features which affects label. For example, in the field of bioinformatics, the goal is to find out some of the factors most relevant to a disease. In this case, feature extraction cannot be used as a method of feature dimension reduction.

According to the process of feature selection, feature selection can be divided into filtering, wrapping and embedding. Feature selection has two major tasks, eliminating redundancy between features and discarding features that are not related to label. The specific details are introduced in the next section.

This paper introduces a new graph-based filtering feature selection algorithm. The feature is regarded as the point of the graph structure. The edge between the point and the point is the correlation between the feature and the feature. The information in each node is the correlation between the feature and the label. And set a threshold and crop some edges with lower correlation to reduce the amount of calculation in the graph. The algorithm uses the idea of Message Passing Neural Nets (MPNN) [5] to integrate the information of each node's first-order neighbors with the designed integration function, and update the information of each node. Finally, we adopt the idea of SIFT (Scale-invariant feature transform) [6] to select the key point and select the local maximum node. The selected nodes are more stable and robust than other nodes, and the features corresponding to such nodes are considered to be selected features.

The rest of the paper is structured as follows. Overview of feature selection will be introduced in Sect. 2. MPNN and mutual information will be discussed in Sect. 3, and the proposed feature selection algorithm will be briefly described in Sect. 4. Finally, the experimental procedure and results of the comparison with the ReliefF and SFS algorithms will be described in Sect. 5. Finally, the conclusion of the work is offered in Sect. 6.

2 Overview of Feature Selection Method

The filter feature selection is independent of the subsequent classification (regression) model. Generally, the Pearson correlation index, mutual information, and maximum information coefficient are used to judge the relationship between features. The original filtering algorithm only considers the relationship between features or only considers the relationship between features and labels, so as to sort the features and select the optimal *M* features. But simply considering a single task of feature selection, the selected feature subset is not optimal. Caruana et al. proposed the SFS (Sequence Forward Search) algorithm [7], the algorithm is based on the idea of greed, and the optimal one is selected each time. The feature subset starts from the empty set, and each time the feature with the best fitness function is added to the feature subset until all the features are traversed. The disadvantage of the SFS algorithm is that it depends on the fitness function and can only be added to the feature subset, and cannot be eliminated. Therefore, there is still a case of feature redundancy. In 1992, Kira proposed the relief algorithm [8], which is only suitable for the feature selection of two-class tasks. A sample is randomly selected, and then *K* neighbor samples (for example, cosine similarity is used to calculate the similarity between samples) of the same classes as the selected sample are selected, and *K* neighbor samples of different classes from the selected sample are selected, and determine if the feature makes sense for the classification, and calculate weights based on this. Iterate multiple times, and finally sort according to the weight of each feature to select the appropriate feature. Later, in order to solve the multi-classification problem, Kononenko proposed the ReliefF algorithm [9]. The Relief series of algorithms are simple and efficient, and there are no restrictions on the data types. It belongs to a feature weighting algorithm. Therefore, features with a high correlation with the label will be given higher weights. The limitation of this algorithm is that it cannot effectively eliminate redundant features.

The wrapped feature selection takes the performance of the latter learning model as a reference, for example, the accuracy of the final model is used as a criterion for judging the quality of the feature subset. The main idea of the representative wrap feature selection algorithm is to regard feature selection as *01* problem, *0* for no selection, *1* for selection, and the feature selection problem to find the optimal solution in the solution space. For example, feature selection based on genetic algorithm (GAs) [10], feature selection based on particle swarm optimization algorithm PSO [11, 12] and gray wolf algorithm (GWO) [13] and particle swarm optimization algorithm combined Algorithms [14]. Of course, there are also some feature selection algorithms based on other optimization algorithms [15–18]. Because the wrap feature selection is the most reference to the performance of the model, the resulting solution will reach an approximate optimal solution as the number of iterations increases, with the accompanying calculation being particularly large.

The embedded feature selection is the same as the wrapper feature selection method, which is related to the training model, and the process of feature selection is embedded in the learning process of the model. Common embedded feature selection methods are generally applied to the regression task. The L_2 paradigm (Lasso) is embedded in the loss function of the learning model to achieve the compression factor effect. A feature that is not associated with a label, its coefficients are compressed to a small extent, and the feature coefficients associated with the label are amplified. In 2004, Efron proposed the Least Angle Regression Algorithm (LARS) [19], which treats the label *Y* as a vector and other features as vectors. The algorithm starts with all coefficients being zero, first finds the feature variable X_1 most relevant to the label Y, and proceeds on the solution path of the selected variable until there is another variable X_2, so that the two variables have the same correlation with the current residual. Then repeat this process, LARS guarantees that all the variables of the selected regression model will advance on the solution path, and the correlation coefficient with the current residual is the same. At the end of the article, the author proves that LARS and Lasso regression are equivalent. In addition, there are more traditional feature selection methods Bess (best subset selection) [20] for the feature selection of regression tasks. Consider the coefficient β_i of a feature as unknown, and use the loss function $l(\beta_i)$ to perform Taylor expansion, and calculate the difference between the minimum value of the expansion and $l(0)$. Sorting according to the difference, screening a part of the feature, the algorithm also embeds the process of updating the coefficient. Generally, the embedded feature selection algorithm is applied to the regression task, and the embedded feature selection method for the classification task is less.

3 Related Work

There are many criteria for the correlation between two variables. The most commonly used are Pearson correlation coefficients, cosine similarity, etc. All of the above criteria measure the value of continuous values, and mutual information has a better performance for discrete attributes. The proposed algorithm is a feature selection for the classification task and needs to measure the correlation between the feature and the label. Therefore, mutual information is used as the correlation criterion in the experiment.

3.1 Mutual Information

In 1948, Claude Shannon, the father of information theory, proposed the theory of information entropy. Information entropy represents the average information of multiple possibilities of one thing. The formula of information entropy is defined as follows:

$$H(X) = -\sum_{x \in X} p(x) \log(p(x)) \tag{1}$$

Mutual information is a very useful measure of information in information theory. It can be regarded as the amount of information about another random variable contained in a random variable, or it is the uncertainty of a random variable that is reduced due to the knowledge of another random variable.

Definition 1: Let the joint distribution of two random variables $(X;Y)$ be $p(x,y)$, the edge distribution be $p(x), p(y)$, and the mutual information $I(X;Y)$ be the relative entropy between the joint distribution $p(x,y)$ and the edge distribution $p(x)p(y)$, i.e.:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{2}$$

In addition, mutual information can also be expressed by the following formula:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) = H(X) - H(X|Y) \tag{3}$$

3.2 MPNN

MPNN is a method designed to extract the features of a topology. We describe MPNNs which operate on undirected graphs G with node features x_v and edge features e_{vw}. To enrich the relationship between each node and other nodes, MPNN allows each node to grasp the information from local to global through the spread of the message. The forward propagation of MPNN consists of two steps, a message-passing phase, and a readout phase. The message propagation process is updated by the information transfer function M_t and the node update function U_t. The hidden feature of the node after T update is defined as h_v^T, and initial state is defined as $h_v^0 = x_v$. Hidden features during messaging are defined according to the following formula:

$$m_v^{t+1} = \sum_{w \in N(v)} M_t\left(h_v^t, h_w^t, e_{vw}\right) \tag{4}$$

$$h_v^{t+1} = U_t\left(h_v^t, m_v^{t+1}\right) \tag{5}$$

Nv represents the first-order neighbor node of v. With the times of updates, the information of the node's features is more and more comprehensive, analogous to the convolution of the image. As the convolutional layer increases, the extracted information becomes more and more comprehensive. Finally, the prediction can be made based on the updated node information. The prediction function has the following definitions:

$$\hat{y} = R\left(\left\{h_v^T | v \in G\right\}\right) \tag{6}$$

The core idea of MPNN is to integrate the information of the neighbor nodes of a node onto the node. In this paper, the proposed algorithm is modified based on the idea of MPNN. The structure of the feature selection problem is regarded as the graph structure, and the feature is regarded as a node. The weight of the edge in the graph is the correlation between the feature and the label is stored in the node. The proposed algorithm sets the threshold and rejects the edges with small weights, then the graph changes from a complete graph to a normal graph. There are two reasons for this. First, the computational overhead is reduced. Second, the remaining edges are considered valid after the edges with small weights are discarded, because these edges are considered weakly correlated or irrelevant.

3.3 Node Information Update

The criterion for a good feature is that the feature is related to the label and is not related to other features. Assuming that mutual information is used as a measure of variable relevance, the greater the mutual information value, the greater the correlation between the variables. The two major tasks of feature selection can be integrated into one formula: the correlation between features and labels minus the redundancy of features and other features. The greater the difference, the better the feature, and the smaller the difference, the more redundant or irrelevant the feature is. Consider this difference as the score of the feature. The process of calculating the difference corresponds to the part of the MPNN node update information. In this paper, the node information is updated according to the following formula:

$$v_i = I(X_i, Y) - \sum_{j \in Ner(i)} I\left(X_i, X_j\right) \tag{7}$$

The node updated information is represented by v_i, and $Ner(i)$ represents the first-order neighbor node of node i.

There are two extreme examples in (a) and (b) in Fig. 1, both of which are first-order neighbor graphs for the feature X_5. The X_5 in (a) is a redundant feature, because it has a great correlation with other features. After updating the value in the node by formula (7), it will be found that the value of the X_5 node is compressed very low; The node X_5 in (b) is a feature that has a high correlation with the label and low correlation with other features. Such a feature is an ideal feature. After the update of the formula (4), the value of X_5 will be higher than the other node values after the update. In the actual experiment, the edges with small weights are discarded, but the redundancy of related features and the redundancy of unrelated features can be used for reference.

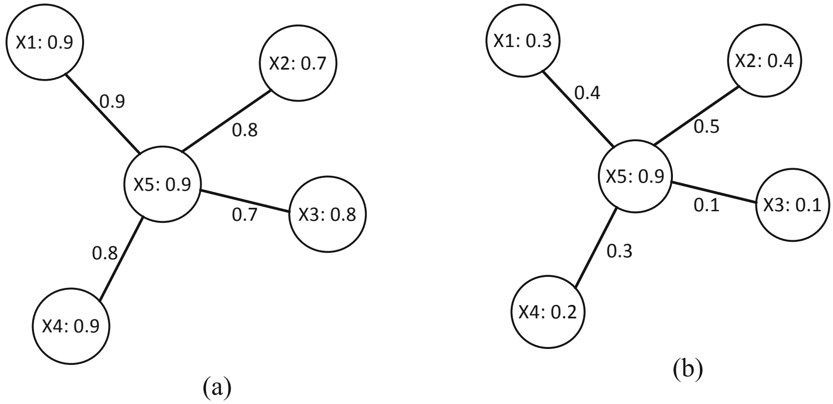

Fig. 1. Partial node diagram

4 Proposed Method

In order to solve the problem of feature selection, many researchers solve problems from different angles. The current popular research angle is to regard the problem as an NP-hard problem, and use the optimization algorithm to find the optimal solution in the solution space; Another angle is to treat feature selection as a clustering problem. Based on the hypothesis (1), features with redundancy are considered to be classified into the same cluster. The first one belongs to the wrapper type scheme, which is expensive to calculate, but the effect is good; the second type is filter type, and the calculation consumption is small, but the performance is not as good as the wrapper type, but it can process high-dimensional data.

Hypothesis (1): If the similarity between feature X_1 and feature X_2 is high, then the correlation of X_1 with the label approximates the relevance of X_2 to the label.

The proposed algorithm is based on the second solution. Many researchers have established different algorithms based on the theory of feature clusters. In [21], the algorithm proposed by Wang uses clustering algorithm to search clusters in different subspaces. According to the idea of KNN, features are added to existing clusters. And the author sets a threshold for the robustness of the cluster, and if the maximum distance between the feature and the cluster is greater than the threshold, a new cluster is established. Besides, in [22], the algorithm proposed by Song Q is also based on feature clusters, but instead of using clustering algorithms, the topology is used to construct the relationship between features and features. Use the Prim algorithm to generate the minimum spanning tree, and then turn it into a forest according to a certain strategy. Each tree in the forest corresponds to a cluster. The same point of the two algorithms is to find the feature cluster and then select the best feature from the feature cluster, but such a method will lose some features and the performance of redundant feature culling of high-dimensional data is not very good. The proposed algorithm does not need to find such a feature cluster. The graph propagated through the MPNNs network will integrate the information of the neighbor nodes. Assuming that the node is redundant, the score of the node is lowered, so that the node (feature) with a high local score is taken as a key node, and such a feature has higher correlation with the label and lower redundancy.

Based on the description of the above information, a novel feature selection algorithm GBFS (graph-based feature selection) is proposed. The algorithm first relies on correlation metrics to calculate the correlation between features and label, and then to filter the correlation between some features. In the experiment, the threshold was selected as the median of all the correlations, and the correlations less than the median were all discarded. The structure of the graph is constructed based on the remaining information. The specific construction process is as follows: The correlation between the feature and the feature is taken as the weight of the edge in the graph, and the correlation between the feature and the label is used as the information of the node. After that, the node information is updated according to the idea of the message propagation network (formula (7)). Only need to spread once, without having to propagate multiple times to get global information, because the graph before the filter edge is a complete graph, and the value less than the median is considered weakly correlated or irrelevant. Then, after screening, for a node, the remaining neighbor nodes are related, and are related to the node in the global scope. Finally, the best node is selected by the principle of local optimality. In theory, the score of a node is greater than that of all its neighbors. Such a node is considered to be the best, but in reality, such conditions are more rigorous, resulting in the sparse features selected. In the experiment, the selection condition is relaxed, and the node whose score is greater than half is considered to be critical. If the dimensions of the dataset are large, you can still limit the selection criteria.

From the description of the above algorithm, the running time of the proposed algorithm is mainly in calculating the correlation between any two features. In this step, the algorithm requires the calculation of C_n^2 times of mutual information. Therefore, the time complexity of the correlation between the features of the algorithm is calculated as $O(n^2 * T)$, and the time complexity of calculating the mutual information is assumed to be T. Since the adjacency matrix is used to store the information of the edge, the time complexity of updating the node information is $O(n^2)$, and the time complexity of finding the best node is also $O(n^2)$. Finally, the proposed algorithm has a time complexity of $O(n^2 * T)$. It can be seen that the time complexity is a quadratic term polynomial, and the algorithm execution efficiency is relatively simple and efficient. The specific algorithm pseudo-code is as follows:

Algorithm BGFS

```
Input: X ∈ $R^{nxm}$ , Y ∈ $R^{nx1}$
Output: the selected feature subset S
Init: Node = ∅， edge = ∅, S = ∅
For i to m do
    $v_i$ = I($X_i$, Y)
    Node = Node ∪ {$v_i$}
Endfor
For i=1 to m do
    For j=i+1 to m do
        $e_{ij} = I(X_i, X_j)$
        If $e_{ij}$ > θ then
            Edge = Edge ∪ {$e_{ij}$}
    Endfor
Endfor
Construct_graph(Node, Edge)
For $i$ to m do
    Update($v_i$) using formula (7)
End for
For $i$ to m do
    flag = 0
    neibigours = Ner($v_i$)
    For each $v_j$ ∈ nebigiours do
        If max($v_i$, $v_j$) == true then   //$v_i$ grater than $v_j$ return true
            flag = flag + 1
    Endfor
    If flag > β * len(neibigours) then
        S = S ∪ {$i$}
Endfor
Return S
```

5 Experimental Study and Result Analysis

To verify the performance of the GBFS algorithm, this paper mainly evaluates the classification accuracy of the selected feature subsets on the SVM classifier. In the experiment, the performance of GBFS algorithm and SFS and ReliefF two classic filtering feature selection algorithms are compared.

5.1 Data Set Description

Eight data sets from UCI were selected in the experiment. Table 1 lists the relevant parameters of the data set. Where samples represent the number of samples in the data set, attributes represent the number of features, Discrete attributes represent the number of discrete attributes, Continuous attributes represent the number of continuous attributes, and classes represent the number of categories.

Table 1. UCI data set.

Dataset	Samples	Attributes	Discrete attributes	Continuous attributes	Classes
Dermatology	366	34	33	1	6
Ionosphere	351	34	2	32	2
Sonar	208	60	0	60	2
Wdbc	569	30	0	30	2
Wine	178	13	0	13	3
Parkinsons	197	22	0	22	2
Lung	32	56	56	0	3
Hill valley	606	100	0	100	2

5.2 10-Fold Cross Validation

The experiment used a ten-fold cross-validation to estimate the performance of the model. The data set was divided into ten, and nine of them were taken as a training set and one was used as a test set. The corresponding accuracy is obtained for each experiment, and the average of the 10 results is used as an estimate of the performance of the algorithm. In the experiment, multiple ten-fold cross-validation was used, and the average value was calculated as an estimate of the accuracy of the algorithm. The reason for choosing to divide the data set into 10 is because a large number of researchers use a large number of data sets and use different algorithms to carry out continuous experiments, which shows that it is the best choice for obtaining the best error estimate, and there is also a theoretical proof.

5.3 Result Analysis

Because some UCI data sets are relatively neat, so before the experiment, the data set is disrupted, and then 10-fold cross-validation. The SFS algorithm and the ReliefF algorithm are sorting feature selection algorithms. Therefore, it is necessary to specify the number of features. In the experiment, the number of features selected according to the GBFS algorithm is selected, and the same number of features are specified in SFS and ReliefF. Compare the classification performance of the features selected by the three algorithms in the SVM classifier. Table 2 lists the experimental results of eight data sets, where SVM indicates that the original data set is directly trained using the SVM classifier, GBFS+SVM represents the feature selected by the GBFS feature, then put into the SVM trainer, and so on. And it can be seen that the number of features selected by the GBFS algorithm is about half of the original feature number, because the limit value of the algorithm in the experiment is the median, and the key point selection strategy is greater than half. If you want to continue to reduce features or increase features, you can adjust the parameters θ and β in the pseudo-code. The data in Table 2 shows in discrete data sets, the proposed algorithm selects features better than other algorithms. For data-regulated data sets, the algorithm also performs well. For example, the second data set ionosphere, whose received signals were processed using an autocorrelation function before making data sets. But in some data sets with continuous value attributes, the proposed algorithms don't perform well, even worse than them. Because mutual information measures the continuous value data set is not very good, resulting in the construction of the topology map does not represent the relationship between features well. In the case of small sample size, the effect is not very good, because the essence of mutual information is still through probability statistics. The less the data, the less accurate the statistical probability. Therefore, the proposed method is applicable to the case where the attribute value is discrete and the sample size is large. From the final average accuracy, the GBFS algorithm is a bit higher than ReliefF and smaller than SFS.

Table 2. Classification results of data sets.

Dataset	SVM	GBFS+SVM	RliefF+SVM	SFS+SVM	Features selected
Dermatology	0.911 ± 0.024	0.959 ± 0.012	0.859 ± 0.025	0.884 ± 0.029	18
Ionosphere	0.942 ± 0.024	0.980 ± 0.018	0.977 ± 0.019	0.897 ± 0.026	18
Sonar	0.559 ± 0.11	0.581 ± 0.079	0.523 ± 0.064	0.826 ± 0.040	31
Wdbc	0.642 ± 0.038	0.690 ± 0.028	0.708 ± 0.046	0.730 ± 0.040	16
Wine	0.410 ± 0.072	0.792 ± 0.057	0.410 ± 0.052	0.968 ± 0.018	7
Parkinsons	0.759 ± 0.041	0.767 ± 0.040	0.761 ± 0.039	0.763 ± 0.046	12
Lung	0.433 ± 0.12	0.443 ± 0.12	0.452 ± 0.17	0.476 ± 0.17	29
Hill valley	0.478 ± 0.024	0.480 ± 0.020	0.476 ± 0.020	0.484 ± 0.027	51
Average	0.642	0.712	0.646	0.754	–

To verify the anti-interference of the model, the original data set was modified and were added with noise. The purpose of adding the noise dimension is to interfere with the correlation between the features and the labels, because the generated noise dimension may be more relevant to labels than the original features. As the noise increases, some of the algorithm's flaws are amplified. Suppose we have a data set (n samples, m features), add noise data, and represent noise data in gray. The first mode of adding noise, as shown in Fig. 2. According to the number of the original features, the same number of noise features are added, and the noise features generated by the random numbers are weakly correlated with each other. The experimental results are shown in Table 3. The horizontal anti-interference of the three algorithms is similar. The second noise-adding mode, as shown in Fig. 3, first copies the original data set so that the sample size is twice (2n), and then adds the noise features of the same feature number (m). The experimental results are shown in Table 4. Longitudinal expansion is equivalent to data enhancement, so accuracy is generally improved. In this noise mode, the proposed algorithm stability is similar to other algorithms.

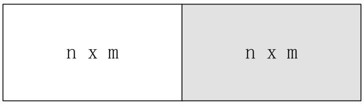

Fig. 2. Noise mode 1

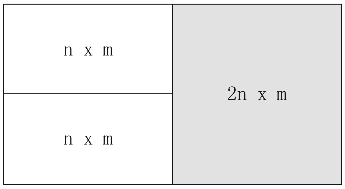

Fig. 3. Noise mode 2

Finally, the results of the original data set are averaged with the results of noise mode 1 and noise mode 2. As shown in Table 5, the anti-interference of GBFS is better than the other two algorithms. In other words, the accuracy of the various situations is combined and the proposed algorithm is more stable.

Table 3. The result of noise mode 1

Dataset	SVM	GBFS+SVM	RliefF+SVM	SFS+SVM	Features selected
Dermatology	0.911 ± 0.017	0.926 ± 0.022	0.909 ± 0.033	0.563 ± 0.064	35
Ionosphere	0.905 ± 0.036	0.939 ± 0.025	0.945 ± 0.025	0.895 ± 0.030	35
Sonar	0.523 ± 0.063	0.531 ± 0.055	0.537 ± 0.042	0.508 ± 0.083	61
Wdbc	0.627 ± 0.028	0.655 ± 0.030	0.655 ± 0.033	0.640 ± 0.029	31
Wine	0.531 ± 0.077	0.460 ± 0.068	0.411 ± 0.068	0.921 ± 0.044	14
Parkinsons	0.805 ± 0.048	0.785 ± 0.044	0.795 ± 0.036	0.835 ± 0.041	23
Lung	0.357 ± 0.14	0.381 ± 0.11	0.395 ± 0.12	0.343 ± 0.14	57
Hill valley	0.494 ± 0.045	0.498 ± 0.039	0.489 ± 0.036	0.491 ± 0.030	101
Average	0.644	0.647	0.642	0.650	–

Table 4. The result of noise mode 2

Dataset	SVM	GBFS+SVM	RliefF+SVM	SFS+SVM	Features selected
Dermatology	0.960 ± 0.011	0.967 ± 0.012	0.969 ± 0.010	0.918 ± 0.028	35
Ionosphere	0.976 ± 0.014	0.993 ± 0.0085	0.994 ± 0.012	0.953 ± 0.019	35
Sonar	0.587 ± 0.081	0.688 ± 0.033	0.690 ± 0.050	0.528 ± 0.029	61
Wdbc	0.884 ± 0.021	0.890 ± 0.019	0.891 ± 0.018	0.644 ± 0.027	31
Wine	0.873 ± 0.048	0.856 ± 0.049	0.826 ± 0.042	0.970 ± 0.022	14
Parkinsons	0.923 ± 0.015	0.931 ± 0.034	0.939 ± 0.028	0.901 ± 0.036	23
Lung	0.591 ± 0.12	0.729 ± 0.118	0.787 ± 0.085	0.388 ± 0.12	57
Hill valley	0.802 ± 0.019	0.790 ± 0.029	0.800 ± 0.024	0.511 ± 0.026	101
Average	0.824	0.856	0.862	0.727	–

Table 5. Anti-interference test results

Data	SVM	GBFS+SVM	ReliefF+SVM	SFS+SVM
Original data	0.642	0.712	0.646	0.754
Noise model 1	0.644	0.647	0.642	0.650
Noise model 2	0.824	0.856	0.862	0.727
Average	0.703	0.738	0.717	0.710

6 Conclusion

This paper proposes a new feature selection algorithm based on the feature cluster theory. Different from the traditional selection scheme based on feature cluster theory, the proposed algorithm does not need to find feature clusters. The selection process of feature clusters is embedded into these two steps through the MPNN message delivery mechanism and node information update formula in the third section. Compared with other sorting feature selection methods, such as SFS and ReliefF algorithm, the proposed algorithm does not need to specify the number of selected features. Besides, the advantage of the proposed algorithm is that it can be used to increase the computational speed with distributed system calculations. In the experiment, we use matrix to represent the structure of the graph, so we can use the distributed parallel computing of the matrix to speed up the operation. The disadvantage of the proposed algorithm is that as the data dimension increases, the amount of calculation of the graph becomes larger and larger, and the number of edges can only be reduced by increasing the threshold, but this will bring about loss of information. The proposed algorithm does not trade well between performance and amount of computation. Future work will improve the proposed algorithm to resolve the contradiction between high dimensional data and computational complexity. Secondly, there is no uniform standard for data preprocessing, and there is no good evaluation standard for continuous value discretization. However, the impact of continuous value discretization on the experiment is still very large. In the next research work, the correlation criteria between continuous value attribute and discrete value attribute will be also further explored and studied. Finally, to verify the anti-interference of the proposed model, noise data is added horizontally and vertically in the eight original UCI data sets. The anti-interference experiment results show that the proposed model is better robust than the other two.

References

1. Cai, Z., Goebel, R., Salavatipour, M., Lin, G.: Selecting dissimilar genes for multi-class classification, an application in cancer subtyping. BMC Bioinform. **8**, 206 (2007). (IF: 3.428)
2. Cai, Z., Zhang, T., Wan, X.: A computational framework for influenza antigenic cartography. PLoS Comput. Biol. **6**(10), e1000949 (2010)
3. Cai, Z., Xu, L., Shi, Y., Salavatipour, M., Goebel, R., Lin, G.: Using gene clustering to identify discriminatory genes with higher classification accuracy. In: IEEE the 6th Symposium on Bioinformatics and Bioengineering (BIBE 2006) (2006)
4. Yang, K., Cai, Z., Li, J., Lin, G.: A stable gene selection in microarray data analysis. BMC Bioinform. **7**, 228 (2006)
5. Gilmer, J., Schoenholz, S.S., Riley, P.F., et al.: Neural message passing for quantum chemistry (2017)
6. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (2004)
7. Caruana, R., De Sa, V.R.: Benefitting from the variables that variable selection discards. J. Mach. Learn. Res. **3**, 1245–1264 (2003)
8. Kira, K., Rendell, L.A.: A practical approach to feature selection. In: Proceedings of the Ninth International Workshop on Machine Learning, pp. 249–256 (1992)
9. Kononenko, I.: Estimating attributes: analysis and extensions of RELIEF. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-57868-4_57
10. Pal, S.K., Wang, P.P.: Genetic Algorithms for Pattern Recognition. CRC Press Inc., Boca Raton (1996)
11. Kennedy, J.: Particle swarm optimization. In: Sammut, C., Webb, G.I. (eds.) Encyclopedia of Machine Learning, pp. 760–766. Springer, Boston (2010). https://doi.org/10.1007/978-0-387-30164-8
12. Chuang, L.Y., Chang, H.W., Tu, C.J., et al.: Improved binary PSO for feature selection using gene expression data. Comput. Biol. Chem. **32**(1), 29–38 (2008)
13. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**(3), 46–61 (2014)
14. Emary, E., Zawbaa, H.M., Hassanien, A.E.: Binary grey wolf optimization approaches for feature selection. Neurocomputing **172**, 371–381 (2016)
15. Mafarja, M.M., Mirjalili, S.: Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection. Soft. Comput. **23**(15), 6249–6265 (2019)
16. Al-Tashi, Q., Kadir, S.J.A., Rais, H.M., et al.: Binary optimization using hybrid grey wolf optimization for feature selection. IEEE Access **7**, 39496–39508 (2019)
17. Mafarja, M., Aljarah, I., Faris, H., et al.: Binary grasshopper optimisation algorithm approaches for feature selection problems. Expert Syst. Appl. **117**, 267–286 (2019)
18. Li, W., Chao, X.Q.: Improved particle swarm optimization method for feature selection. J. Front. Comput. Sci. Technol. **13**(6), 990–1004 (2019)
19. Efron, B., Hastie, T., Johnstone, I., et al.: Least angle regression. Ann. Stat. **32**(2), 407–451 (2004)

20. Wen, C., Zhang, A., Quan, S., et al.: BeSS: an R package for best subset selection in linear, logistic and CoxPH models (2017)
21. Wang, L., Jiang, S.: Novel feature selection method based on feature clustering. Appl. Res. Comput. **32**(5), 1305–1308 (2015)
22. Song, Q., Ni, J., Wang, G.: A fast clustering-based feature subset selection algorithm for high-dimensional data. IEEE Trans. Knowl. Data Eng. **25**(1), 1–14 (2013)