Taylor & Francis
Taylor & Francis Group

Journal of Applied Statistics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/cjas20

Feature selection based on distance correlation: a filter algorithm

Hongwei Tan , Guodong Wang , Wendong Wang & Zili Zhang

To cite this article: Hongwei Tan , Guodong Wang , Wendong Wang & Zili Zhang (2020): Feature selection based on distance correlation: a filter algorithm, Journal of Applied Statistics, DOI: 10.1080/02664763.2020.1815672

To link to this article: https://doi.org/10.1080/02664763.2020.1815672

Published online: 07 Sep 2020.

JOURNAL OF APPLIED STATISTICS
https://doi.org/10.1080/02664763.2020.1815672

Taylor & Francis
Taylor & Francis Group

Feature selection based on distance correlation: a filter algorithm

Hongwei Tan[a,b], Guodong Wang[a], Wendong Wang[a] and Zili Zhang[a,c]

[a]School of Computer and Information Science, Southwest University, Chongqing, People's Republic of China; [b]School of Mathematics and Statistics, GuiZhou University of Finance and Economics, Guiyang, People's Republic of China; [c]School of Information Technology, Deakin University, Geelong, Australia

ABSTRACT

Feature selection (FS) is one of the most powerful techniques to cope with the curse of dimensionality. In the study, a new filter approach to feature selection based on distance correlation is presented (DCFS, for short), which keeps the model-free advantage without any pre-specified parameters. Our method consists of two steps: hard step (forward selection) and soft step (backward selection). In the hard step, two types of associations, between univariate feature and the classes and between group feature and the classes, are involved to pick out the most relevant features with respect to the target classes. Due to the strict screening condition in the first step, some of the useful features are likely removed. Therefore, in the soft step, a feature-relationship gain (like feature score) based on the distance correlation is introduced, which is concerned with five kinds of associations. We sort the feature gain values and implement the backward selection procedure until the errors stop declining. The simulation results show that our method becomes more competitive on several datasets compared with some of the representative feature selection methods based on several classification models.

ARTICLE HISTORY
Received 25 July 2019
Accepted 21 August 2020

KEYWORDS
Feature selection; distance correlation; *S*-correlation; feature relevance; feature redundancy

1. Introduction

Feature selection has been widely investigated in machine learning and data mining community, which aims at selecting an optimal feature subset from original feature set and improving the performance of final model. Feature selection algorithms can be partitioned into three classes: embedded methods [16,22], which merge the selection process into the learning of the classifier; wrapper methods [1,8], which leverage classifiers to achieve a given subset of features; filter methods [3,4,11,12], which analyze the intrinsic properties of data, ignoring the classifier. In this paper, we focus on the last one.

There exists several recent work on the filter feature algorithms, which are generally categorized into two groups. One consists of mutual information-based algorithms that

CONTACT Zili Zhang zhangzl@swu.edu.cn School of Computer and Information Science, Southwest University, Chongqing 400715, People's Republic of China School of Information Technology, Deakin University, Locked Bag 20000, Geelong, VIC 3220, Australia

leverage the mutual information (MI, a measure of dependence between two random variables) to craft an effective filter for eliminating the unwanted features. The algorithms mRMR [12], NMIFS [4], and F-score [19] are known to belong to this group. [2,10,18] comprehensively investigated the type of algorithms. Typically, the MI-based algorithms are not concerned with the relationship between group features since MI between group features is hardly estimated. While our proposed method appropriately integrates the relationship into the screening procedure to improve the performance of the final model. Moreover, due to the fact that MI between continuous and discrete features cannot be theoretically estimated, thus the type of algorithms performs slightly bad in the mix-type feature setting. In our experiments, the results in Tables 7 and 8 verify the viewpoint.

The other consists of distance measure-based algorithms. When a measure represents the distance between a feature set and the target feature set in some metric space, we call it distance measure-based. The distance measures used in feature selection range from Euclidean distance to more complex distance, such as Minkowski distance [6], Manhattan distance [7], and distance correlation [11,14,17,20]. Recently, the distance correlation-based feature selection methods have received a lot of attention, where DCSIS [11], DisCoMax [17] are two representative algorithms. The screening condition of DCSIS is simple yet effective ($dcor(f_i, y)^2 > \alpha n^{-\kappa}$). However, the algorithm requires two pre-specified parameters (α, κ) that may cause the instable performance of the final model and also does not involve the relationship between group features. For the DisCoMax, it needs to learn a low-dimensional feature representation z, which maximizes the distance correlations between the feature z and covariates x, and also between the feature z and response y. Similarly, DisCoMax also requires two pre-specified parameters (α, γ) in the optimization process. By contrast, our proposed method does not specify any hyper parameters.

In this study, we propose a new filter feature selection method based on the distance correlation [15] which is free of distributional assumptions. Furthermore, the method does not require any pre-specified parameters, neither does it implement optimization problem. These advantages enable the method to work effectively in feature selection. Our method is performed by two steps: hard step, which executes the forward selection by considering two types of relationships between univariate feature and the classes, and between group feature and the classes; soft step, which further detects the useful features in the removed features of the hard step by the backward selection. In order to improve the efficiency of the backward selection procedure, a feature-relationship gain (like feature score) is introduced, which comprehensively takes into account five types of associations between: (1) univariate feature and the classes; (2) group feature and the classes; (3) the selected feature subset and the set of all features; (4) the eliminated feature subset and the set of all features; (5) the selected feature subset and the eliminated feature subset. The experimental results testify that our method is more competitive compared with some of the representative feature selection methods such as mRMR, NMIFS, DCSIS and DisCoMax.

The rest of this paper is structured as follows: Section 2 briefly introduces the distance correlation. Section 3 develops some theoretical underpinnings for our algorithm and proposes a new filter algorithm for feature selection. Section 4 examines the performance of the algorithm via the experiments on synthetic and benchmark datasets. Finally, conclusion is drawn in Section 5.

2. Notations and background

2.1. Notations

X and Y are p-dimension and q-dimension random vectors, where their characteristic function and joint characteristic function are denoted by $f_X(\cdot)$, $f_Y(\cdot)$ and $f_{X,Y}(\cdot,\cdot)$, respectively. The Euclidean norm of $\mathbf{x}$ in $\mathbb{R}^p$ is $\|\mathbf{x}\|_p$. In the feature selection procedure, the set of the target classes is denoted by C, and $C = \{c_1, c_2, \ldots, c_k\}$. The set of all features is denoted by F, and $F = \{f_1, f_2, \ldots, f_n\}$. S represents an arbitrary feature subset of F, that is, $S \subseteq F$, while the subset $S^{(+i)} = S \cup \{f_i\}$, $i = 1, 2, \ldots, n$, the univariate feature $f_i \notin S$, and $S^{(-j)} = S \setminus f_j$, $f_j \in S$. The selected feature subset in the hard step is denoted by S_{HS}, while the subset $S_{soft} = F \setminus S_{HS}$ is fed into the soft step. For two vectors $\mathbf{x}$, $\mathbf{y}$, $(\mathbf{x}, \mathbf{y})$ is a new vectors that the vector $\mathbf{x}$ concatenates with the vector $\mathbf{y}$, e.g. $\mathbf{x} = (1, 2)$, $\mathbf{y} = (3, 4, 5)$, then $(\mathbf{x}, \mathbf{y}) = (1, 2, 3, 4, 5)$. Note that we do not distinguish random variable from feature or the class, unless otherwise specified.

2.2. Background: distance correlation

Székely *et al.* [15] proposed distance correlation (*dCor*, for short) to measure all types of dependence between two random vectors X and Y in arbitrary dimension. The core idea of *dCor* leverages the characteristic function of random vector to establish dependence statistics. As we know, two random vectors X and Y are independent if and only if $f_{X,Y}(\mathbf{t}, \mathbf{s}) = f_X(\mathbf{t}) f_Y(\mathbf{s})$. Suppose $d = \|f_{X,Y}(\mathbf{t}, \mathbf{s}) - f_X(\mathbf{t}) f_Y(\mathbf{s})\|^2$, then d can be used to measure certain distance between X and Y. [15] defined the distance covariance (abbreviated to *dCov*) as a metric of dependence between X and Y, denoted by $dCov(X, Y)$, and its square was defined by

$$dCov^2(X, Y) = \int_{R^{p+q}} \|f_{X,Y}(\mathbf{t}, \mathbf{s}) - f_X(\mathbf{t}) f_Y(\mathbf{s})\|^2 w(\mathbf{t}, \mathbf{s})\, \mathrm{d}\mathbf{t}\, \mathrm{d}\mathbf{s}, \tag{1}$$

where $w(\mathbf{t}, \mathbf{s}) = (c_p c_q \|\mathbf{t}\|_{p+1}^{p} \|\mathbf{s}\|_{q+1}^{q})^{-1}$ is the weight function with $c_d = \pi^{(1+d)/2} / \Gamma((1 + d)/2)$. Correspondingly, the square of distance variance (abbreviated to *dV ar*) was defined as $dVar^2(X) = dCov^2(X, X)$. Analogous to Pearson correlation coefficient, *dCor* between random vectors X and Y was defined by

$$dCor(X, Y) = \begin{cases} \dfrac{dCov(X, Y)}{\sqrt{dVar(X) \cdot dVar(Y)}}, & dVar(X) \cdot dVar(Y) > 0, \\ 0, & dVar(X) \cdot dVar(Y) = 0. \end{cases} \tag{2}$$

Two attractive properties of *dCor* [13,15] motivate us to utilize it for designing the feature selection algorithm. The first one is that it can measure the linear or nonlinear relationship between two random vectors (or variables) in arbitrary dimension. This property implies that it is an effective tool to quantify the intrinsic relationships between univariate feature and the target classes, between two feature groups, and between the target classes and group feature. The second remarkable property of *dCor* is that *dCor* equals 0 if and only if two random vectors are independent. This property further makes it possible that *dCor* can be used as a dimension reduction tool, as we will see in the following sections.

3. Feature selection based on distance correlation

In this section, a filter algorithm for feature selection based on *dCor* is presented. To make our algorithm practical and more effective, some theoretical underpinnings on the algorithm are developed, such as proposing *S*-correlation, *G*-correlation, re-quantifying feature relevance and redundancy. The pseudo codes of the algorithm are shown in Algorithm 1.

3.1. Definitions based on distance correlation

As above mentioned, *dCor* is an effective tool that can measure arbitrary types of associations between two random vectors. In this view, we can also leverage it to measure the associations between features. For the sake of clarity, some basic notions with respect to feature selection will be redefined based on *dCor*.

Definition 3.1 (*f*-correlation, *C*-correlation, *S*-correlation, *G*-correlation): Let S_1, S_2 be two subsets of F, and f_i, $f_j \in F$, $i \neq j$, $i,j = 1,2,\ldots,n$, then $dCor(f_i,f_j)$, $dCor(C,f_i)$, $dCor(C,S)$ and $dCor(S_1,S_2)$ are called f-correlation, C-correlation, S-correlation and G-correlation.

Remark 3.1 ([21]): Proposed the notions of f-correlation and C-correlation based on MI. Similarly, we easily redefine the f-correlation and C-correlation based on *dCor* to quantify the relationships of feature–feature and feature–class. Furthermore, the notions of *S*-correlation and *G*-correlation are introduced, which quantify the relationships of group feature–class and group feature–group feature. Importantly, in practice, these quantitative indicators are easily computed and explained.

Definition 3.2 (Feature relevance): For a given feature f_i, $i = 1,2,\ldots,n$, if $dCor(C,f_i) > 0$, then called the feature f_i is a relevant feature with respect to C.

Remark 3.2: According to Definition 3.2, the feature relevance is quantified with $dCor(C,f_i)$, i.e. C-correlation. Clearly, the greater $dCor(C,f_i)$ is, the stronger the relevance is, and vice versa.

Definition 3.3 (Feature redundancy): Let S_{ij} be a subset of F that consists of the features f_i and f_j, that is, $S_{ij} = \{f_i,f_j\}$, if the relevance between features f_i and f_j is stronger, and $dCor(f_i,C) > dCor(S_{ij},C)$, then the feature f_j with respect to C is redundant.

Remark 3.3: Different from the traditional definitions [5,21], Definition 3.3 is simple yet practical. The first condition implies that the information of the feature f_j overlaps with the feature f_i, and the second condition illustrates that the feature f_j has no 'contribution' to the target task, therefore the feature f_j with respect to the target class C is redundant. In addition, if the relevance between features f_i and f_j is weakly, and $dCor(f_i,C) \leq dCor(S_{ij},C)$, then the feature f_j is weakly relevant but non-redundant. It can be seen that the weakly relevant but non-redundant features contribute to the target task, thus we suggest the kinds of features should be added into the final model.

3.2. Theoretical results

Our aim is to design an effective feature selection algorithm based on *dCor*. To ensure the validity of our algorithm theoretically, we prove Lemma 3.4, Lemma 3.5 and Theorem 3.6.

Lemma 3.4: *Let* $\widetilde{X}$ *and* $\widetilde{Y}$ *be two random vectors, if* $\widetilde{X}$ *and* $\widetilde{Y}$ *are independent, and then for arbitrary vectors* $\mathbf{t}$ *and* $\mathbf{s}$*, such that* $\exp\{i\langle\widetilde{X}, \mathbf{t}\rangle\}$ *and* $\exp\{i\langle\widetilde{Y}, \mathbf{s}\rangle\}$*are independent, where* i *denotes imaginary unit.*

Proof: See Appendix A.1. ■

Lemma 3.5 ([9]): *For arbitrary random vectors* X, Y*, suppose a random vector* (Y_1, Y_2) *is a partition of the random vectors* Y*, that is,* $Y_1 \cap Y_2 = \emptyset$, $Y_1 \cup Y_2 = Y$*, if* Y_2 *and* (X, Y_1) *are independent, and then* $dCov^2(X, Y) \leq dCov^2(X, Y_1)$.

Proof: See Appendix A.2. ■

Lemma 3.5 shows that if the irrelevant information is mixed into the random vector space (X, Y_1), and it cannot strengthen the relevance between random vectors X and Y_1. Inversely, it may weaken the relevance. In term of Lemma 3.5, we easily obtain Theorem 3.6.

Theorem 3.6: *Suppose* $S \subseteq F, f_i \in F$*, and* $S^{(+i)} = \{f_i\} \cup S, f_i \notin S, i = 1, \ldots, n$*, if the feature* f_i *and random vector* (C, S) *are independent, then* $dCor^2(C, S^{(+i)}) \leq dCor^2(C, S)$.

Proof: See Appendix A.3. ■

Note that here both the class C and the feature subset S are treated as a vector. Theorem 3.6 implies that if a feature f_i and the random vector (C, S) are not independent, then $dCor^2(C, S^{(+i)}) \geq dCor^2(C, S)$, that is, when a feature f_i is strongly dependent or relevant on the target class C, then the S-correlation of the subset $S^{(+i)}$ is larger than the S-correlation of the subset S. Therefore, the strongly relevant feature subset can be picked out. In our Algorithm 1, the hard step (forward selection) is crafted based on Theorem 3.6.

3.3. Filter algorithm for feature selection

Our method consists of two steps: hard step, which executes the forward selection by considering the C-correlation and S-correlation, that is, $dCor(f_i, C)$ and $dCor(S, C)$; soft step, which further detects the useful features in the removed features of the hard step by the backward selection. In order to improve the efficiency of the backward selection procedure, a feature-relationship gain (like feature score) is introduced, which takes into account three types of correlations: (1) C-correlation between univariate feature and the classes; (2) S-correlation between group feature and the classes; (3) G-correlations between the selected feature subset and the set of all features, between the eliminated feature subset and the set of all features, and between the selected feature subset and the eliminated feature subset. In the following, the method will be formally proposed, and its pseudo codes are shown in Algorithm 1.

Algorithm 1 DCFS Algorithm.

Require:

A training dataset with n features, $(f_1, \ldots, f_n)$ and the class C.

Ensure:

A selected feature subset, S_{opt}

1: **Hard Step:**

(1) Calculating $dCor(C, f_i)$ and initializing the S_{HS} with the largest $dCor(C, f_i)$;
(2) Repeating: if a feature f_j meets $dCor(S_{HS}^{(+j)}, C) > dCor(S_{HS}, C)$, then the feature f_j enters into S_{HS}, otherwise removed into S_{soft}.

2: **Soft Step:**

(1) Calculating FRG in S_{soft} and sorting it by the descending order;
(2) Repeating: these features are added into the classifier in turn;
(3) Until: the errors of the classifier stop declining;
(4) Return: an optimal subset $S_{opt\_softstep}$

3: **return** S_{opt}, and $S_{opt} = S_{HS} \cup S_{opt\_softstep}$

Hard Step. The goal of the step is to explore the strongly relevant feature subset, that is, S_{HS}. According to Theorem 3.6, the details of the step are described by

(1) computing the C-correlation, i.e. $dCor(f_i, C), i = 1, 2, \ldots, n$, and then using the largest $dCor(f_i, C)$ to initialize the strongly relevant feature subset S_{HS};
(2) screening procedure: for a feature f_j, $i \neq j = 1, 2, \ldots, n$, if $dCor(S_{HS}^{(+j)}, C) > dCor(S_{HS}, C)$, then the feature f_j enters into S_{HS}, otherwise removed.

It can be seen that the condition of the hard step is strict, which the selected features are required meeting the condition (2). Due to the hard condition, so the step was referred to as 'hard step'. Indeed, it is also a forward selection procedure. The drawback of the step is that it is likely to eliminate the subtle 'important' features. In order to overcome the drawback, the second step of our method, soft step, is implemented. First, a feature-relationship gain (*FRG*, for short) is introduced. For a arbitrary feature $f_m \in S_{soft}$, the feature-relationship gain is defined by

$$FRG = \frac{dCor(S_{HS}^{(+m)}, F) - dCor(S_{soft}^{(-m)}, F) + \exp\left[dCor(f_m, C)\right]}{dCor(S_{HS}^{(+m)}, S_{soft}^{(-m)}) + \exp\left[dCor(S_{HS}^{(-m)}, C) - dCor(S_{soft}^{(+m)}, C)\right]}. \tag{3}$$

Remark 3.4: From the mathematic perspective, the essence of the feature selection problem is to find out such a basis that represents the feature space and almost contains all information in the feature space, then the basis or feature group is the optimal feature subset. In the soft step, if a feature $f_m (\in S_{soft})$ is a useful feature, it should meet: (1) the feature f_m should increase the relationship between $S_{HS}^{(+m)}$ and F when it is added

into the strongly relevant feature subset S_{HS} along with cutting down the relationship between $S_{soft}^{(-m)}$ and F (this amounts to a kind of 'information transformation'), and its C-correlation is also taken into account. Thus the numerator of FRG, $dCor(S_{HS}^{(+m)}, F) - dCor(S_{soft}^{(-m)}, F) + \exp[dCor(f_m, C)]$, is required becoming large; (2) in the denominator of FRG, the value $dCor(S_{HS}^{(+m)}, S_{soft}^{(-m)})$ with respect to the feature f_m should become small, which its purpose is to make the selected features as independent as possible from the removed features, along with the small difference of S-correlation, that is, the value $dCor(S_{HS}^{(-m)}, C) - dCor(S_{soft}^{(+m)}, C)$ is required small. Therefore, the larger a feature's FRG is, the more important it is. Note that the inserted exponential $\exp(\cdot)$ of FRG has two purposes that: (1) guarantees the nonnegativity of FRG; (2) zooms up the feature importance exponentially, if the C-correlation of a feature f_m ($dCor(f_m, C)$) is larger along with the smaller difference of S-correlation ($dCor(S_{HS}^{(-m)}, C) - dCor(S_{soft}^{(+m)}, C)$), then the feature is probably important. The FRGs of all features in S_{soft} is sorted by the descending order, and then these features are fed into the model in turn until the error stop declining.

Note that our algorithm does not strictly discriminate the irrelevant features from the redundant features, that is because the algorithm can eliminate the unwanted features together and retain the useful features. In addition, computing the distance correlation is only computation burden in Algorithm 1, thus the time complexity of the algorithm is in an acceptable range. In Section 4, its performance will be tested by the extensive experiments.

4. Experiments

In this section, to verify the effectiveness of our algorithm, the empirical experiments on synthetic and benchmark datasets were implemented. In all experiments, we performed the fivefold cross validation on each dataset and reported the average root mean square error (RMSE) on the hold-out test set. The results demonstrate that our algorithm is more effective in classification tasks based on four classifiers: random forest, SVM, naive Bayes and AdaBoost. All results were obtained by R codes.

4.1. Results and discussion on synthetic datasets

In the experiments, three datasets, Monk1, Monk2 and Monk3, were picked out from UCI Repository,[1] where the class labels of Monk3 include 5% noise data. Each dataset with 6 features consists of a training set (432 samples) and a test set (169 samples). The Monk problems were the basis of first international comparison of learning algorithms, which the target tasks were all binary classifications. The class labels of each dataset are determined by three boolean functions as follows:

- Monk1: $C(f_1, \ldots, f_6) = (f_1 = f_2) \vee (f_5 = 1)$;
- Monk2: if$(f_i = 1) \wedge (f_j = 1)$, $i \neq j$, $i, j = 1, 2, \ldots, 6$, and then $C(f_1, \ldots, f_6) = 1$, else $C(f_1, \ldots, f_6) = 0$;
- Monk3: $C(f_1, \ldots, f_6) = (f_5 = 3 \wedge f_4 = 1) \vee (f_5 \neq 4 \wedge f_2 \neq 3)$.

Table 1. Testing the effectiveness of DCFS based random forest model in the hard step.

Title	Selected feature subset	FS_RMSE	AF_RMSE
Monk1	$\{f_1, f_2, f_5\}$	**0.1667**	0.3156
Monk2	$\{f_1, f_3, f_4, f_6\}$	0.6314	**0.6009**
Monk3 (Noise)[a]	$\{f_2, f_5\}$	**0.1674**	0.7201

Table 2. Testing the backward selection of the relevant feature in the soft step.

Datasets	*FRG*	Selected Feature Subset	RMSE
Monk1	$f_3 : 1.1538$	$\{f_1, f_2, f_5, f_3\}$	0.1700
	$f_4 : 1.4427$	$\{f_1, f_2, f_5, f_4\}$	0.1689
	$f_6 : 1.1293$	$\{f_1, f_2, f_5, f_6\}$	0.3087
Monk2	$f_2 : 0.9627$	$\{f_1, f_3, f_4, f_6, f_2\}$	0.5182
	$f_5 : 1.2820$	$\{f_1, f_3, f_4, f_6, f_5\}$	0.4481
Monk3	$f_1 : 1.1627$	$\{f_2, f_5, f_1\}$	0.1746
	$f_3 : 1.0039$	$\{f_2, f_5, f_3\}$	0.2042
	$f_4 : 1.1552$	$\{f_2, f_5, f_4\}$	0.1734
	$f_6 : 1.0509$	$\{f_2, f_5, f_6\}$	0.2152

Clearly, the relevant feature subsets of three datasets with respect to the target class C are that: (1) Monk1: $\{f_1, f_2, f_5\}$; (2) Monk2: $\{f_i, i = 1, 2, \ldots, 6\}$; (3) Monk3: $\{f_2, f_4, f_5\}$. The others are the irrelevant features.

Hard Step. The hard step was performed in our algorithm based on random forest model. The results are shown in Table 1, where FS_RMSE and AF_RMSE denote the average root mean square error on the selected feature subset and the set of all features, respectively. As shown in Table 1, the FS_RMSE (bold) in Monk1 and Monk3 are less than the corresponding AF_RMSE. Especially, in Monk1, all of the relevant features are selected by our method (in the hard step). Even if in the noise setting (in Monk3), it also works well. Only in Monk2, its FS_RMSE (0.6314) performs slightly than the corresponding AF_RMSE (0.6009). The results verify that our method is effective.

Soft Step. For the Monk2 dataset, the two relevant features f_2, f_5 are not selected out in the hard step. Also, the relevant feature f_4 in Monk3 is rejected in this step. In order to more precisely pick out the other relevant features, the soft step, backward selection, was implemented. The *FRG* of each rejected features in the hard step including the corresponding RMSE value was computed in Table 2. In Monk1, the RMSEs of three combinations are greater than the RMSE of the feature subset $\{f_1, f_2, f_5\}$ in Table 1 (0.1667). This implies that the features f_3, f_4 and f_6 are irrelevant ones with regard to the target classes. For the Monk2 dataset, the combination $\{f_1, f_3, f_4, f_6, f_5\}$ is an optimal feature subset from the perspective of RSME (0.4481). Interestingly, in Monk3, the RMSE value goes up instead (from 0.1674 to 0.1734) when the relevant feature f_4 was added into the feature subset $\{f_2, f_5\}$. This indicates the feature f_4 is likely a weakly relevant feature, which removing this feature doesn't make much difference.

The results on synthetic datasets demonstrate that DCFS is an effective feature selection algorithm. In the following, its effectiveness will be further verified by comparing with several representative algorithms on various benchmark datasets.

Table 3. Summary of five UCI benchmark datasets.

Title	Attribute types	Instances	Features	Classes
Dermatology	Discrete	366	34	6
Sonar	Continuous	208	60	2
Vehicle	Continuous	846	18	4
Ionosphere	Mix	351	34	2
QSAR Biodegradation	Mix	1055	41	2

Table 4. Comparison of the RMSE values with the different numbers of the selected features (SF) on Dermatology based on the random forest model. The baseline result (RMSE) on dermatology based on random forest model is 0.1547.

Methods\ SF numbers	6	10	14	18	22
Ours	0.4863	**0.2833**	**0.2370**	**0.2055**	**0.1938**
DCSIS	0.4978	0.4358	0.3757	0.2362	0.2014
MVSIS	0.4865	0.4858	0.3945	0.2089	0.2227
mRMR	**0.4421**	0.4044	0.3612	0.3371	0.3367
NMIFS	0.4703	0.3860	0.3510	0.3136	0.2056
F-score	–	–	–	–	–
SPSA-FSR	0.6683	0.4758	0.5133	0.4203	0.3952
DisCoMax	0.6231	0.4864	0.4311	0.4078	0.2817

4.2. Results and discussion on benchmark datasets

In the series of experiments, five benchmark datasets were selected from UCI Repository and more details are shown in Table 3. Note that all of the target tasks are classification. Indeed, DCFS belongs to the supervised feature selection algorithm. We compared our algorithm with seven representative algorithms, including DCSIS [11], MVSIS [3], mRMR [12], NMIFS [4], *F*-score [19], SPSA-FSR (wrapper algorithm) [1] and DisCoMax [17]. Due to the space limitations, we only exhibit the results on random forest here and the results on the other classifiers can be found in Appendix 2.

Dermatology dataset contains 8 missing values with 366 instances, 34 features and 6 classes, where the missing values were filled by the mean imputation method. In the experiment, we picked out five different numbers of the selected features (SF numbers: 6, 10, 14, 18, 22) to compare the RMSE values with the other seven algorithms (Table 4). Note that the *F*-score algorithm can only deal with the binary classification problem. As shown in Table 4, clearly, our results (RMSEs) are better than almost all other methods, only mRMR (0.4421) performs slightly better than our approach (0.4863) when the SF numbers equals 6. Especially, the superiority is significant on the SF numbers 10 and 14. On the other hand, although DCSIS, DisCoMax and our method all belong to the algorithms based on *dCor*, our method completely outperforms the other two on all SF numbers.

For the two continuous datasets, Sonar (Table 5) and Vehicle (Table 6), our method still possesses some advantages. Especially on Sonar, our method is still superior to the other methods with the exception of the SF number 10. But the performance of our method lags behind mRMR with the SF numbers 5, 9 and DisCoMax with the SF number 13 on Vehicle (Table 6). It is the probable reason that mRMR is an algorithm based on mutual information, which mutual information can be estimated on the continuous datasets well. But on the mix type datasets, this advantage would not become remarkable. Also, according

Table 5. Comparison of the RMSE values with different numbers of the selected features (SF) on Sonar based on the random forest model. The baseline result (RMSE) on Sonar based on random forest model is 0.4049.

Methods\ SF numbers	10	20	30	40	50
Ours	0.5198	**0.4113**	**0.3828**	**0.3779**	**0.3617**
DCSIS	0.5248	0.4539	0.4549	0.4121	0.4204
MVSIS	0.5065	0.4582	0.4498	0.4432	0.4053
mRMR	**0.4709**	0.4237	0.4300	0.3962	0.4367
NMIFS	0.5276	0.4305	0.4214	0.3900	0.3809
F-score	0.5140	0.4562	0.4265	0.4393	0.4199
SPSA-FSR	0.5408	0.4319	0.3993	0.4079	0.3799
DisCoMax	0.6715	0.5433	0.5181	0.4430	0.4162

Table 6. Comparison of the RMSE values with different numbers of the selected features (SF) on vehicle based on the random forest model. The baseline result (RMSE) on vehicle based on random forest model is 0.5113.

Methods\ SF numbers	5	7	9	11	13
Ours	0.5755	**0.5395**	0.5332	**0.5126**	0.5094
DCSIS	0.5829	0.5615	0.5411	0.5295	0.5151
MVSIS	0.5820	0.5494	0.5373	0.5189	0.5111
mRMR	**0.5729**	0.5465	**0.5097**	0.5191	0.5086
NMIFS	0.6278	0.5821	0.5568	0.5270	0.5221
F-score	–	–	–	–	–
SPSA-FSR	0.5870	0.5541	0.5436	0.5288	0.5152
DisCoMax	0.6734	0.5908	0.5435	0.5276	**0.4924**

Table 7. Comparison of the RMSE values with different numbers of the selected features (SF) on Ionosphere based on the random forest model. The baseline result (RMSE) on Ionosphere based on random forest model is 0.2559.

Methods\ SF numbers	5	7	9	11	13
Ours	**0.2823**	**0.2604**	**0.2613**	**0.2561**	**0.2443**
DCSIS	0.2897	0.3012	0.2741	0.2718	0.2626
MVSIS	0.3263	0.3262	0.3118	0.2693	0.2538
mRMR	0.3020	0.2844	0.2837	0.2775	0.2611
NMIFS	0.3159	0.2972	0.2826	0.2772	0.2724
F-score	0.2973	0.2979	0.3284	0.2874	0.2720
SPSA-FSR	0.2876	0.2627	0.2625	0.2573	0.2668
DisCoMax	0.3901	0.3574	0.2990	0.3012	0.2847

to the results of Tables 4, 5 and 6, the DisCoMax algorithm performs better in the low-dimension feature dataset (Table 6) than in the high-dimension dataset (Tables 4 and 5).

On the two mix datasets, Ionosphere and QSAR Biodegradation, our method completely outperforms the other compared methods with all SF numbers in Tables 7 and 8. Furthermore, according to the results of Ionosphere (Table 7), it is superior to the baseline (0.2559, RMSR in the set of all features) when the SF number equals 13 (0.2443). Similarly, on QSAR Biodegradation, Table 8 shows that the performance of our method has outperformed the baseline (0.3657) when the SF number equals 15 (0.3605). These results testify that our method is more competitive on the mix datasets compared with the other methods based on the random forest model. Also, this further verifies that our method can pick out the useful features and eliminate the unwanted features.

Table 8. Comparison of the RMSE values with different numbers of the selected features (SF) on QSAR Biodegradation based on the random forest model. The baseline result (RMSE) on QSAR Biodegradation based on random forest model is 0.3657.

Methods\ SF numbers	10	15	20	25	30
Ours	**0.3764**	**0.3605**	**0.3624**	**0.3613**	**0.3537**
DCSIS	0.4195	0.3916	0.3770	0.3673	0.3731
MVSIS	0.4219	0.3716	0.3830	0.3652	0.3753
mRMR	0.3841	0.3729	0.3655	0.3690	0.3754
NMIFS	0.3839	0.3736	0.3705	0.3668	0.3680
F-score	0.4202	0.3803	0.3639	0.3666	0.3622
SPSA-FSR	0.4051	0.3738	0.3675	0.3683	0.3539
DisCoMax	0.3946	0.4117	0.3913	0.3885	0.3699

5. Conclusion

In this study, we bridged the distance correlation (*dCor*) with feature selection and presented a filter algorithm, named DCFS, which solved the feature selection problem. The effectiveness of DCFS depended on two steps: (1) hard step, which implements the forward feature selection; (2) soft step, which executes the backward feature selection. Particularly, in the soft step, the feature-relationship gain (*FRG*) was introduced, which is the key element for the backward selection. The sorted features by *FRG* were fed into the model in turn. By doing so, this greatly reduced the time complexity of backward search. In addition, some notions based on *dCor* were proposed, such as *S*-correlation, *G*-correlation and feature redundancy. Under the definitions framework, we proved Theorem 3.6 that is the theoretical underpinnings of the hard step. The experimental results show that our method can pick out the useful features and eliminate the unwanted features on several datasets. In the future work, we would like to further dig into our ideas in more depth and develop the unsupervised or semi-supervised feature selection algorithms based on the distance correlation.

Note

1. http://mlr.cs.umass.edu/ml/datasets.html

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work is supported by the National Natural Science Foundation of China: Managing Uncertainty in Service Based Software with Intelligent Adaptive Architecture (No. 61732019).

References

[1] V. Aksakalli and M. Malekipirbazari, *Feature selection via binary simultaneous perturbation stochastic approximation*, Pattern. Recognit. Lett. 75 (2016), pp. 41–47.
[2] J. Cai, J. Luo, S. Wang, and S. Yang, *Feature selection in machine learning: A new perspective*, Neurocomputing 300 (2018), pp. 70–79.
[3] H. Cui, R. Li, and W. Zhong, *Model-free feature screening for ultrahigh dimensional discriminant analysis*, J. Am. Stat. Assoc. 110 (2015), pp. 630–641.

[4] P.A. Estévez, M. Tesmer, C.A. Perez, and J.M. Zurada, *Normalized mutual information feature selection*, IEEE Trans. Neural Networks 20 (2009), pp. 189–201.
[5] W. Gao, L. Hu, and P. Zhang, *Class-specific mutual information variation for feature selection*, Pattern. Recognit. 79 (2018), pp. 328–339.
[6] S. García, J. Luengo, and F. Herrera, *Data Preprocessing in Data Mining*, Springer, Cham, 2015.
[7] S. Happy, R. Mohanty, and A. Routray, *An effective feature selection method based on pair-wise feature proximity for high dimensional low sample size data*, *2017 25th European Signal Processing Conference (EUSIPCO)*, IEEE, 2017, pp. 1574–1578.
[8] M.M. Kabir, M.M. Islam, and K. Murase, *A new wrapper feature selection approach using neural network*, Neurocomputing 73 (2010), pp. 3273–3283.
[9] M. R. Kosorok, *Correction: discussion of Brownian distance covariance*, Ann. Appl. Stat. 7 (2013), pp. 1247–1247.
[10] Y. Li, T. Li, and H. Liu, *Recent advances in feature selection and its applications*, Knowl. Inf. Syst.53 (2017), pp. 551–577.
[11] R. Li, W. Zhong, and L. Zhu, *Feature screening via distance correlation learning*, J. Am. Stat. Assoc.107 (2012), pp. 1129–1139.
[12] H. Peng, F. Long, and C. Ding, *Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy*, IEEE Trans. Pattern. Anal. Mach. Intell. 27 (2005), pp. 1226–1238.
[13] M.L. Rizzo and G.J. Székely, *Energy distance*, Wiley Interdisc. Rev. Comput. Stat. 8 (2016), pp. 27–38.
[14] W. Sheng and X. Yin, *Sufficient dimension reduction via distance covariance*, J. Comput. Graph. Stat.25 (2016), pp. 91–104.
[15] G.J. Székely, M.L. Rizzo, and N.K. Bakirov, *Measuring and testing dependence by correlation of distances*, Ann. Stat. 35 (2007), pp. 2769–2794.
[16] R. Tibshirani, *Regression shrinkage and selection via the lasso: a retrospective*, J. Royal Stat. Soc. B Stat. Methodol. 73 (2011), pp. 273–282.
[17] P. Vepakomma, C. Tonde, and A. Elgammal, *Supervised dimensionality reduction via distance correlation maximization*, Electron. J. Stat. 12 (2018), pp. 960–984.
[18] J.R. Vergara and P.A. Estévez, *A review of feature selection methods based on mutual information*, Neural Comput. Appl. 24 (2014), pp. 175–186.
[19] D. Wang, Z. Zhang, R. Bai, and Y. Mao, *A hybrid system with filter approach and multiple population genetic algorithm for feature selection in credit scoring*, J. Comput. Appl. Math. 329 (2018), pp. 307–321.
[20] C.D. Yenigün and M.L. Rizzo, *Variable selection in regression using maximal correlation and distance correlation*, J. Stat. Comput. Simul. 85 (2015), pp. 1692–1705.
[21] L. Yu and H. Liu, *Efficient feature selection via analysis of relevance and redundancy*, J. Mach. Learn. Res. 5 (2004), pp. 1205–1224.
[22] J. Zhao, L. Chen, W. Pedrycz, and W. Wang, *Variational inference-based automatic relevance determination kernel for embedded feature selection of noisy industrial data*, IEEE Trans. Indust. Electron. 66 (2018), pp. 416–428.

Appendices

Appendix 1: Proofs

A.1 Proof of Lemma 3.4

Proof: Since $\widetilde{X}$ and $\widetilde{Y}$ are independent, and then

$$f_{\widetilde{X},\widetilde{Y}}(\mathbf{t},\mathbf{s}) = f_{\widetilde{X}}(\mathbf{t}) f_{\widetilde{Y}}(\mathbf{s}),$$

hence, we have

$$\begin{aligned} E(\exp\{i\langle\widetilde{X},\mathbf{t}\rangle + i\langle\widetilde{Y},\mathbf{s}\rangle\}) &= E(\exp\{i\langle\widetilde{X},\mathbf{t}\rangle\}\exp\{i\langle\widetilde{Y},\mathbf{s}\rangle\}) \\ &= E(\exp\{i\langle\widetilde{X},\mathbf{t}\rangle\})E(\exp\{i\langle\widetilde{Y},\mathbf{s}\rangle\}), \end{aligned}$$

where $E(\cdot)$ and $\langle,\rangle$ denote expectation and inner product operator, respectively. Thus $\exp\{i\langle\widetilde{X},\mathbf{t}\rangle\}$ and $\exp\{i\langle\widetilde{Y},\mathbf{s}\rangle\}$ are independent. ■

A.2 Proof of Lemma 3.5

Proof: Due to $Y_1 \cap Y_2 = \emptyset$ and $Y = Y_1 \cup Y_2$, then Y_1 and Y_2 are independent. Let $V(X,Y) = \|f_{X,Y}(\mathbf{t},\mathbf{s}) - f_X(\mathbf{t})f_Y(\mathbf{s})\|^2$, where $f_X(\cdot), f_Y(\cdot)$ and $f_{X,Y}(\cdot)$ denote the characteristic function of X,Y and their joint characteristic function, respectively. Since Y_2 and $X \cup Y_1$ are independent, according to Lemma 3.4, we have

$$\begin{aligned} V(X,Y) &= \|f_{X,Y}(\mathbf{t},\mathbf{s}) - f_X(\mathbf{t})f_Y(\mathbf{s})\|^2 \\ &= \|E(\exp\{i\langle X,\mathbf{t}\rangle + i\langle Y,\mathbf{s}\rangle\}) - E(\exp\{i\langle X,\mathbf{t}\rangle\})E(\exp\{i\langle Y,\mathbf{s}\rangle\})\|^2 \\ &= \|E(\exp\{i\langle X,\mathbf{t}\rangle + i\langle Y_1,\mathbf{s}_1\rangle + i\langle Y_2,\mathbf{s}_2\rangle\}) \\ &\quad - E(\exp\{i\langle X,\mathbf{t}\rangle\})E(\exp\{i\langle Y_1,\mathbf{s}_1\rangle + i\langle Y_2,\mathbf{s}_2\rangle\})\|^2 \\ &= \|E(\exp\{i\langle X,\mathbf{t}\rangle + i\langle Y_1,\mathbf{s}_1\rangle\})E(\exp\{i\langle Y_2,\mathbf{s}_2\rangle\}) \\ &\quad - E(\exp\{i\langle X,\mathbf{t}\rangle\})E(\exp\{i\langle Y_1,\mathbf{s}_1\rangle\})E(\exp\{i\langle Y_2,\mathbf{s}_2\rangle\})\|^2 \\ &= \|[E(\exp\{i\langle X,\mathbf{t}\rangle + i\langle Y_1,\mathbf{s}_1\rangle\}) - E(\exp\{i\langle X,\mathbf{t}\rangle\}) \\ &\quad \cdot E(\exp\{i\langle Y_1,\mathbf{s}_1\rangle\})]E(\exp\{i\langle Y_2,\mathbf{s}_2\rangle\})\|^2 \\ &= \|[f_{X,Y_1}(\mathbf{t}_1,\mathbf{s}_1) - f_X(\mathbf{t})f_{Y_1}(\mathbf{s}_1)]f_{Y_2}(\mathbf{s}_2)\|^2, \end{aligned}$$

According to Cauchy–Schwarz inequality and the property of characteristic function, we obtain

$$\begin{aligned} V(X,Y) &\le \|f_{X,Y_1}(\mathbf{t}_1,\mathbf{s}_1) - f_X(\mathbf{t})f_{Y_1}(\mathbf{s}_1)\|^2 \cdot \|f_{Y_2}(\mathbf{s}_2)\|^2 \\ &\le \|f_{X,Y_1}(\mathbf{t}_1,\mathbf{s}_1) - f_X(\mathbf{t})f_{Y_1}(\mathbf{s}_1)\|^2 = V(X,Y_1), \end{aligned}$$

namely have

$$V(X,Y) \le V(X,Y_1),$$

in terms of Equation (1) and integral isotonicity, and then have

$$dCov^2(X,Y) \le dCov^2(X,Y_1).$$ ■

A.3 Proof of Theorem 3.6

Proof: Let $S^{(+i)} = \{f_i\} \cup S, f_i \notin S, i = 1,\ldots,n$, and the feature f_i and the random vector (C,S) are independent. According to Lemma 3.5,

$$dCov^2(C,S^{(+i)}) \le dCov^2(C,S), \tag{A1}$$

due to $f_i \notin S$, then $\{f_i\} \cup S = (f_i,S)$, note that here the set $\{f_i\} \cup S$ treats as a vector. Clearly, f_i and (f_i,S) are not independent, and because $dVar^2(S^{(+i)}) = dCov^2(S^{(+i)},S^{(+i)}) = dCov^2((f_i,S),(f_i,S)))$, according to Lemma 3.5, we obtain

$$dVar^2(S^{(+i)}) \ge dVar^2(S), \tag{A2}$$

in terms of A1 and A2, and suppose that $dVar(C)$, $dVar(S)$ and $dVar(S^{(+i)})$ are both nonzero, then we have

$$dCor^2(C, S^{(+i)}) = \frac{dCov^2(C, S^{(+i)})}{dVar(C) \cdot dVar(S^{(+i)})} \leq \frac{dCov^2(C, S)}{dVar(C) \cdot dVar(S)} = dCor^2(C, S),$$

namely, $dCor^2(C, S^{(+i)}) \leq dCor^2(C, S)$. ■

Appendix 2: Comparison of the experimental results based on the classifiers SVM, naive Bayes and AdaBoost

To provide more evidence, a series of experiments were performed on the datasets in Table 3 based on the classifiers SVM, naive Bayes and AdaBoost. Note that the kernel function radical was used in SVM. For saving space, the results of three classifiers were compressed into one table.

The results in Tables A1–A5 show that although our method has no absolute advantage on all datasets, it possesses greatly superiority on these datasets, especially on the mix datasets (Tables A4 and A5). The results further verify that our method is more competitive method compared with the representative methods.

Table A1. Comparison of the RMSE values with the different numbers of the selected features (SF) on dermatology based on SVM, naive Bayes and AdaBoost.

SVM: the baseline is 0.1536					
Methods\ SF numbers	6	10	14	18	22
Ours	0.4927	**0.2938**	**0.2595**	**0.1982**	**0.1705**
DCSIS	0.4892	0.4360	0.4257	0.2224	0.2205
MVSIS	0.5089	0.4760	0.4094	0.1996	0.1834
mRMR	**0.4502**	0.3788	0.3905	0.3661	0.3413
NMIFS	0.4979	0.3651	0.3312	0.3528	0.2105
F-score	–	–	–	–	–
SPSA-FSR	0.5602	0.4578	0.3481	0.2780	0.2603
DisComax	0.5914	0.4730	0.4423	0.3896	0.3663
naive Bayes: the baseline is 0.4138					
Ours	**0.5547**	0.5240	**0.4319**	**0.4146**	0.4387
DCSIS	0.6183	0.6725	0.6226	0.4814	0.4876
MVSIS	0.5891	0.5777	0.5531	0.4206	0.4155
mRMR	0.6278	0.6735	0.6329	0.6319	0.6193
NMIFS	0.5766	**0.4687**	0.4642	0.4417	**0.3770**
F-score	–	–	–	–	–
SPSA-FSR	0.5888	0.5929	0.4346	0.4497	0.4045
DisCoMax	0.6482	0.6106	0.5983	0.5170	0.4782
AdaBoost: the baseline is 0.1865					
Ours	0.5114	**0.3218**	**0.2733**	**0.2467**	**0.2417**
DCSIS	0.6315	0.5943	0.4733	0.2844	0.2523
MVSIS	0.6238	0.6223	0.3900	0.2819	0.2807
mRMR	0.6096	0.4266	0.3651	0.3662	0.3213
NMIFS	**0.4687**	0.3382	0.3790	0.3262	0.2590
F-score	–	–	–	–	–
SPSA-FSR	0.6134	0.5521	0.4237	0.3235	0.2987
DisCoMax	0.6002	0.5746	0.4988	0.4051	0.3724

Table A2. Comparison of the RMSE values with different numbers of the selected features (SF) on Sonar based on SVM, naive Bayes and AdaBoost.

SVM: the baseline is 0.4008					
Methods\ SF numbers	10	20	30	40	50
Ours	0.5077	**0.4232**	**0.3928**	**0.3829**	**0.3625**
DCSIS	0.5089	0.4572	0.4073	0.4223	0.4071
MVSIS	0.5188	0.4313	0.4461	0.4069	0.4037
mRMR	**0.4727**	0.4431	0.3943	0.4528	0.4321
NMIFS	0.5067	0.4454	0.4547	0.4660	0.3835
F-score	0.5240	0.4480	0.4655	0.4699	0.4114
SPSA-FSR	0.5157	0.5031	0.4536	0.4202	0.4404
DisCoMax	0.6643	0.5711	0.5074	0.4319	0.3867
naive Bayes: the baseline is 0.5596					
Ours	**0.5933**	**0.5677**	**0.5592**	**0.5588**	0.5703
DCSIS	0.6079	0.5811	0.5706	0.5658	0.5584
MVSIS	0.5975	0.5895	0.5840	0.5681	0.5705
mRMR	0.5977	0.5692	0.5633	0.5596	0.5831
NMIFS	0.5993	0.6079	0.5673	0.5593	**0.5436**
F-score	0.5941	0.5899	0.5881	0.5791	0.5765
SPSA-FSR	0.5958	0.5706	0.5598	0.5590	0.5699
DisCoMax	0.6782	0.6041	0.5811	0.5625	0.5584
AdaBoost: the baseline is 0.4840					
Ours	**0.4899**	**0.4589**	**0.4491**	**0.4415**	**0.4473**
DCSIS	0.5212	0.4621	0.4536	0.4593	0.4745
MVSIS	0.5477	0.4537	0.4499	0.4306	0.4485
mRMR	0.5117	0.4677	0.4829	0.4723	0.4581
NMIFS	0.5319	0.5160	0.4895	0.4819	0.4504
F-score	0.5615	0.4648	0.4575	0.4506	0.4724
SPSA-FSR	0.5340	0.5297	0.4982	0.4739	0.4547
DisCoMax	0.6117	0.6273	0.5834	0.4992	0.4655

Table A3. Comparison of the RMSE values with different numbers of the selected features (SF) on vehicle based on SVM, naive Bayes and AdaBoost.

SVM: the baseline is 0.4928					
Methods\ SF numbers	5	7	9	11	13
Ours	0.5706	0.5568	**0.5213**	0.5139	0.4799
DCSIS	0.6097	0.5878	0.5590	0.5602	0.4936
MVSIS	0.6087	0.5812	0.5711	0.5205	0.4959
mRMR	0.5922	0.5583	0.5286	0.5212	0.4860
NMIFS	0.6737	0.6137	0.5697	0.5381	0.5421
F-score	–	–	–	–	–
SPSA-FSR	**0.5600**	**0.5346**	0.5358	0.5185	0.5366
DisCoMax	0.6135	0.5783	0.5409	**0.4826**	**0.4693**
naive Bayes: the baseline is 0.7408					
Ours	**0.7420**	**0.7215**	0.7489	0.7553	0.7577
DCSIS	0.7536	0.7599	0.7660	0.7646	0.7578
MVSIS	0.7675	0.7657	0.7670	0.7424	0.7599
mRMR	0.7443	0.7399	**0.6823**	0.6930	**0.6729**
NMIFS	0.7509	0.7503	0.7482	0.7413	0.7390
F-score	–	–	–	–	–
SPSA-FSR	0.7636	0.7564	0.7458	0.7448	0.7452
DisCoMax	0.7738	0.7494	0.6949	**0.6845**	0.6820
AdaBoost: the baseline is 0.5235					
Ours	**0.5661**	**0.5582**	0.5533	**0.5325**	0.5196
DCSIS	0.5960	0.5790	0.5574	0.5396	0.5120
MVSIS	0.5842	0.5695	0.5634	0.5475	0.5257
mRMR	0.5774	0.5622	**0.5164**	0.5356	0.5241
NMIFS	0.6177	0.5959	0.5643	0.5488	0.5103
F-score	–	–	–	–	–
SPSA-FSR	0.6029	0.5751	0.5550	0.5521	0.5560
DisCoMax	0.6957	0.6452	0.5943	0.5593	**0.5003**

Table A4. Comparison of the RMSE values with different numbers of the selected features (SF) on ionosphere based on SVM, naive Bayes and AdaBoost.

SVM: the baseline is 0.2695					
Methods\ SF numbers	4	6	8	10	12
Ours	**0.2663**	**0.2695**	**0.2650**	**0.2529**	0.2893
DCSIS	0.2881	0.2765	0.2731	0.2819	0.3011
MVSIS	0.2912	0.2773	0.2680	0.2781	0.2983
mRMR	0.2818	0.2783	0.2678	0.2551	**0.2543**
NMIFS	0.2731	0.2721	0.2693	0.2627	0.2615
F-score	0.2981	0.3063	0.2747	0.2772	0.2584
SPSA-FSR	0.3039	0.3000	0.2741	0.2640	0.2638
DisCoMax	0.3804	0.3728	0.3723	0.3564	0.3092
naive Bayes: the baseline is 0.3127					
Ours	**0.3052**	**0.3029**	**0.3123**	**0.3148**	0.3201
DCSIS	0.3405	0.3170	0.3198	0.3209	0.3210
MVSIS	0.3389	0.3275	0.3205	0.3194	0.3264
mRMR	0.3368	0.3089	0.3224	0.3219	**0.3042**
NMIFS	0.3474	0.3662	0.3799	0.3939	0.3788
F-score	0.3195	0.3411	0.3455	0.3610	0.3441
SPSA-FSR	0.3502	0.3699	0.3777	0.3775	0.3771
DisCoMax	0.4039	0.3872	0.3593	0.3315	0.3246
AdaBoost: the baseline is 0.2749					
Ours	**0.2657**	**0.2736**	**0.2621**	**0.2547**	**0.2720**
DCSIS	0.3874	0.3409	0.3100	0.2987	0.2997
MVSIS	0.3467	0.3120	0.2983	0.2971	0.2773
mRMR	0.3068	0.2838	0.2831	0.2694	0.2751
NMIFS	0.2689	0.2741	0.2925	0.2662	0.2953
F-score	0.3067	0.3057	0.3072	0.3133	0.2751
SPSA-FSR	0.3577	0.3163	0.3148	0.3099	0.2895
DisCoMax	0.3918	0.3921	0.3630	0.3491	0.2936

Table A5. Comparison of the RMSE values with different numbers of the selected features (SF) on QSAR Biodegradation based on SVM, naive Bayes and AdaBoost.

SVM: the baseline is 0.3524					
Methods\ SF numbers	10	15	20	25	30
Ours	**0.4000**	**0.3807**	**0.3841**	**0.3731**	**0.3574**
DCSIS	0.4624	0.3967	0.3962	0.3806	0.3615
MVSIS	0.4485	0.3976	0.3918	0.3909	0.3682
mRMR	0.4079	0.4025	0.3936	0.3770	0.3679
NMIFS	0.4030	0.3879	0.3854	0.3770	0.3611
F-score	0.4514	0.4074	0.3857	0.3787	0.3636
SPSA-FSR	0.4640	0.4100	0.3879	0.3744	0.3767
DisCoMax	0.4099	0.3983	0.3887	0.3900	0.3674
naive Bayes: the baseline is 0.5325					
Ours	**0.5169**	**0.5075**	**0.4867**	0.5018	0.5164
DCSIS	0.5206	0.5296	0.5274	0.5136	0.4853
MVSIS	0.5603	0.5301	0.5250	0.5090	0.4988
mRMR	0.5460	0.6005	0.5583	0.5705	0.5558
NMIFS	0.5501	0.5168	0.5148	0.5208	**0.4925**
F-score	0.5172	0.5197	0.5074	**0.4962**	0.5162
SPSA-FSR	0.6133	0.6126	0.5876	0.5752	0.5453
DisCoMax	0.6734	0.6556	0.5849	0.5822	0.5431
AdaBoost: the baseline is 0.3816					
Ours	0.4055	**0.3873**	**0.3810**	**0.3725**	**0.3684**
DCSIS	0.4375	0.4084	0.3864	0.3845	0.3691
MVSIS	0.4338	0.3956	0.3890	0.3819	0.3886
mRMR	0.4007	0.3955	0.3857	0.3889	0.3743
NMIFS	**0.3827**	0.3898	0.3731	0.3835	0.3780
F-score	0.4285	0.3879	0.3986	0.3830	0.3818
SPSA-FSR	0.4193	0.4009	0.4168	0.4043	0.4011
DisCoMax	0.4637	0.4500	0.4372	0.4319	0.3982