Computers in Biology and Medicine 43 (2013) 91–99

Contents lists available at SciVerse ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/cbm

RFS: Efficient feature selection method based on *R-value*

Jimin Lee, Nomin Batnyam, Sejong Oh*

[a] *Department of Nanobiomedical Science and WCU Research Center of Nanobiomedical Science, Dankook University, Anseodong, Cheonan 330-714, South Korea*

ARTICLE INFO

Article history:
Received 30 July 2011
Accepted 17 November 2012

Keywords:
Feature selection
Classification
Dataset
R-value

ABSTRACT

Feature selection is one of the most important issues in classification. Many filter and wrapper methods have been proposed. Here, we propose a new efficient feature selection method based on the *R-value*, which is a measure that is used to capture the overlapped areas among classes in a feature. Our strategy was to select features that have low overlapping areas among classes. Proposed idea is simple, but powerful for feature selection. The experiment results showed that the proposed method is better than previous typical methods in many cases. Accordingly, the proposed method can be used in combination with other feature selection methods.

1. Introduction

The primary goal of feature selection is to select relevant features and eliminate irrelevant ones in high-dimensional problems to improve the performance of learning models by alleviating the effects of dimensionality, enhancing generalization capability, speeding up learning process and improving model interpretability [1]. In unsupervised learning, feature selection is geared toward obtaining a good subset of features that forms a high quality of clusters for a given number of clusters, while in supervised learning its purpose is to identify a feature subset that produces higher classification accuracy [2].

Feature selection is one of the most important issues in classification, and is particularly relevant in the context of microarray datasets with thousands of features, most of which are likely to be uninformative. In machine learning literature there are two general approaches to feature selection: filters and wrappers [11,22]. Filter methods select the optimal feature subset based solely on training data by evaluating each feature based on specific statistics, but completely independent from the classification algorithm. In contrast, wrapper methods make use of the algorithm that will be used to build the final classifier to select a feature subset. When compared to filters, they tend to be more computationally expensive, but provide superior performance [3] since they are injected inside the learning algorithm and well suited to the interest of the classifier.

Many feature selection methods have been proposed. Here, a brief introduction to the algorithms that were used in our experiment will be reviewed. Leaner discriminant analysis (LDA) [21] is originally developed in 1936 by R.A. Fisher, and used as one of classical feature selection methods. LDA finds a linear transformation (“discriminant function”) of the two predictors, X and Y, that yields a new set of transformed values and provides a more accurate discrimination. This result is used for feature selection.

Feature Selection algorithm based on a Distance Discriminant (FSDD) [4] is a distance discriminant method that belongs to the filter category. The computational complexity of FSDD is low; thus, it fits well into the high-dimensional problems. The criterion used for selecting good features is d_b–ßd_w, where d_b is the distance between different classes, d_w is distance within classes, and ß is a user defined value that is usually set to 2 and used to control the impact of d_w. To compute the criterion, they used a normalized distance measure instead of the Euclidean distance. The algorithm ranks the features in descending order according to the evaluation function and selects the feature subset that gives the highest value.

Relief [5] algorithms are general and successful feature estimators that are used in feature subset selection as well as a variety of other settings, e.g., to select splits or to guide constructive induction in the building phase of decision or regression tree learning, as the feature weighting method and in inductive logic programming. The idea of Relief is to estimate the quality of features according to how well their values distinguish distances that are close to each other. To do so, the algorithm randomly selects an instance and identifies its nearest neighbors, one from its own class and others from the other classes. The quality estimator is then updated for all attributes to assess how well the feature distinguishes the instance from its closest neighbors. The basic algorithm is $W[A]=W[A]\text{-}diff(A,\ R_i,\ I)/m$,

* Corresponding author. Tel.: +82 41 550 3484; fax: +82 41 550 1149.
E-mail addresses: sejongoh@dankook.ac.kr, sejongoh@dku.edu (S. Oh).

0010-4825/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compbiomed.2012.11.010

where *W[A]* is the quality estimation of attribute *A*, R_i is the randomly selected instance, *I* is its nearest neighbor, and *m* is a user-defined value, which is the number of repetitions in the entire process. The function *diff*(*A*, R_i, *I*) calculates the difference between the values of attribute *A* for two instances, *R* and *I*.

Sometimes when good features are combined they are highly correlated with one another; thus, producing the redundancy of the feature set. Minimum Redundancy Maximum Relevance Feature Selection (MRMR) [6,10] has been proposed to solve the problem by (using mutual information) maximizing the mutual Euclidean distance and minimizing the pair-wise correlations of the features. The minimum redundancy condition is $W_I = 1/|s|_2 \sum_{i,j \in S} I(i,j)$, where *S* denotes the feature subset, and *I*(*i*, *j*) is the mutual information of two variables *i* and *j*. To maximize the total relevance $V_I = \ 1/|S| \sum_{i \in S} I(h,i)$, where *I*(*h*, *i*) represents the mutual information between targeted classes *h* and gene expressions *i*.

In our previous study [7], we proposed *R-value* as an evaluation measure for datasets. The motivation for using *R-value* is that the quality of dataset has a profound effect on classification accuracy, and overlapping areas among classes in a dataset have a strong relationship that determines the quality of the dataset. For example, dataset D_1 produces higher classification accuracy than dataset D_2 in Fig. 1. Overlapping area is a region, where samples from different classes are gathered closely to one another. If an unknown sample belongs to the overlapping area, it is difficult to determine its class label. Therefore, the size of overlapping area may be criteria to measure the quality of features or whole dataset. The *R-value* captures overlapping areas among classes in a dataset. A high *R-value* for a dataset indicates that it contains wide overlapping areas among its classes, and classification accuracy on the dataset may become low. *R-value* have three features: $R(C_i,C_j)$, $R(C_k)$, and $R(D)$. $R(C_i,C_j)$ shows the overlapping areas between classes C_i and C_j. $R(C_k)$ shows how many samples in class C_k are located in the overlapping areas with other classes. $R(D)$ shows how many samples in dataset *D* are located in the overlapping areas among classes in the dataset. Fig. 2 shows the areas that $R(C_i,C_j)$, $R(C_k)$, and $R(D)$ are designed to capture.

The original *R-value* was designed to evaluate the entire dataset, but we also found that it could be applied to the feature selection task using the modified $R(D)$. The *R-value*-based feature selection (RFS) method scores the overlapping areas of each feature in candidate features, and then selects features that have low *R-value*. In Section 2, we describe the RFS method in detail. Proposed idea is simple, but powerful for feature selection.

The rest of the paper is organized as follows: an overview of *R-value* for feature selection is presented in Section 2. In Section 3, we describe the experiment's datasets and classifiers, classification accuracies, and computational complexity. Section 4 discusses and analyzes the experimental results and the effects and variations of user defined values. Finally, in Section 5 we conclude the paper and suggest future research.

2. Feature selection based on *R-value*

2.1. Scoring measure of RFS

The core concept of *R-value* is overlapping area. Let's suppose a dataset that has two classes and the distribution of feature

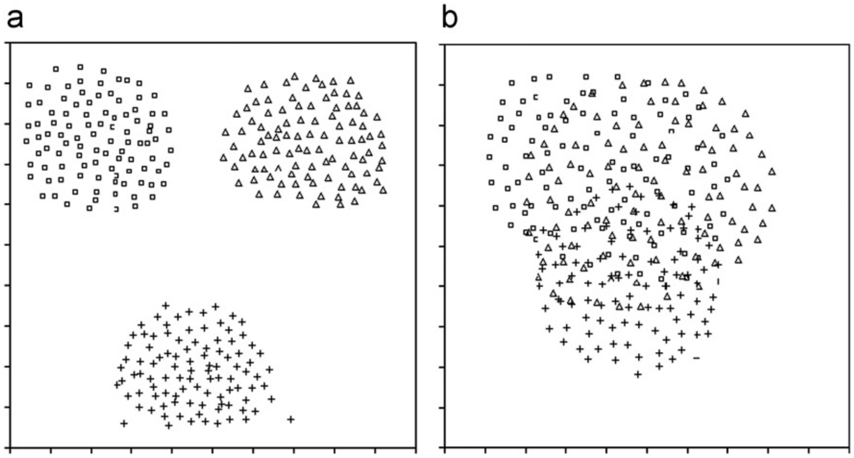


Fig. 1. Two datasets that have different overlapping areas: (a) D_1, (b) D_2.

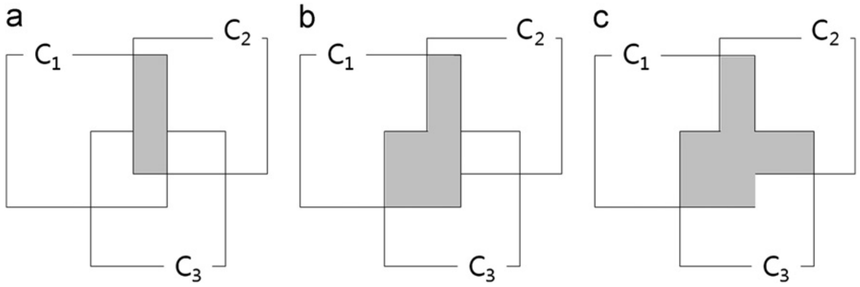


Fig. 2. Three features of *R-value* measure: (a) $R(C_i,C_j)$, (b) $R(C_k)$, (c) $R(D)$.

elements in feature F_k is like Fig. 3. As we can see, the element both p_1 and p_2 belong to class C_1. We can say that the element p_1 does not belong to overlapping area because all neighbor element of p_1 also belongs to C_1. In the case p_2, it belongs to overlapping area because four neighbors belong to different class C_2 whereas only two neighbors belong to same class C_1. If an element belonging to C_1 has two different class' neighbors and four same class' neighbors, is it in overlapping area? To determine this situation, we need some threshold value θ. If the ratio of different class' neighbors is greater than or equal to threshold, we can determine that it belongs to overlapping area. In Fig. 3 we choose 6 as the number of nearest neighbors of point p_1 and p_2. We denote K as a number of nearest neighbors of point p_i. Therefore, our RFS method contains two parameters: threshold θ and K. The range of θ is $0 \le \theta \le K$.

The decision function ola(P_m) is used to decide if point P_m belongs to overlapping area or not.

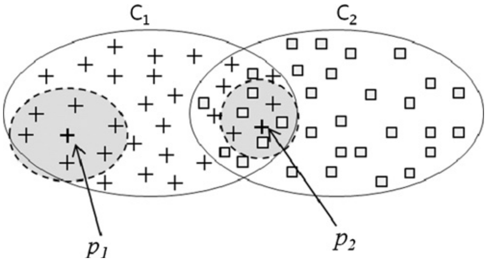


Fig. 3. *K*-nearest neighbor elements of p_1 and p_2.

Definition 1. Decision function for a point P_m

$$\text{ola}(P_m) = \left[\sum_{i=1}^{K} \lambda(NP_i \in (U - Comp(P_m)))\right] - \theta, \tag{1}$$

where

- $\lambda(x)=1$ if $x>0$, else $\lambda(x)=0$
- NP_i: i-th nearest neighbor point of P_m
- $Comp(P_m)$: set of points that belong to class of P_m

In the proposed RFS method, scoring measure $R(F_k)$ for feature F_k is defined by Definition 2.

Definition 2. Scoring function for a feature F_k

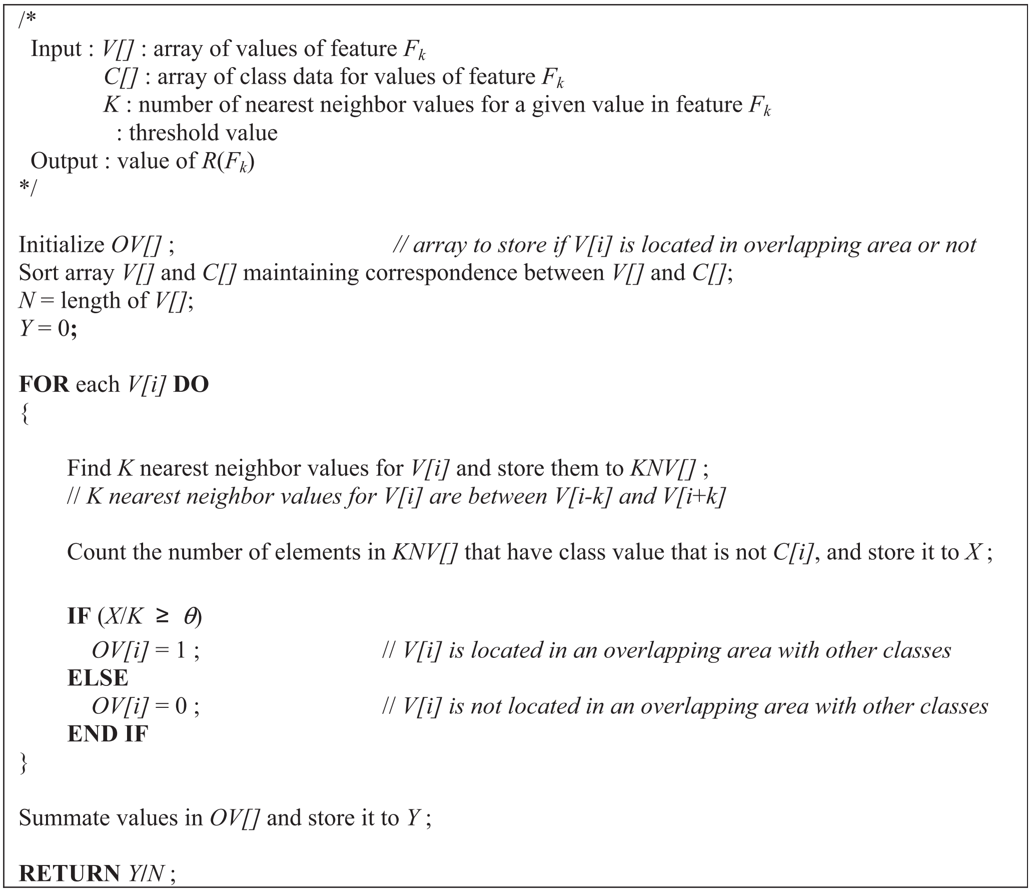
$R(F_k)$=(total number of feature values in F_k that belong to overlapping areas)÷ (total number of samples of given dataset)

$$\frac{1}{|U|}\sum_{m=1}^{|U|} \lambda(ola(P_m)), \tag{2}$$

where

- U: universal set of whole samples that belong to the given dataset
- $|U|$: cardinality of a set U (number of elements of the set U)

Fig. 3 shows the decision if point P_m belongs to overlapping area or not. Let's suppose K=6 and θ=2. In the case of p_1, all 6 NP_i of p_1 belong to same class of p_1. Therefore, ola(p_1)=0–2=−2, and it means that P_1 does not belong to overlapping area. In the case



```
/*
  Input : V[] : array of values of feature $F_k$
          C[] : array of class data for values of feature $F_k$
          K : number of nearest neighbor values for a given value in feature $F_k$
            : threshold value
  Output : value of $R(F_k)$
*/

Initialize OV[] ;                  // array to store if V[i] is located in overlapping area or not
Sort array V[] and C[] maintaining correspondence between V[] and C[];
N = length of V[];
Y = 0;

FOR each V[i] DO
{

    Find K nearest neighbor values for V[i] and store them to KNV[] ;
    // K nearest neighbor values for V[i] are between V[i-k] and V[i+k]

    Count the number of elements in KNV[] that have class value that is not C[i], and store it to X ;

    IF ($X/K \ge \theta$)
      OV[i] = 1 ;                  // V[i] is located in an overlapping area with other classes
    ELSE
      OV[i] = 0 ;                  // V[i] is not located in an overlapping area with other classes
    END IF
}

Summate values in OV[] and store it to Y ;

RETURN Y/N ;
```

Fig. 4. Algorithm for calculating $R(F_k)$.

of p_2, 4 NP_i of P_1 belong to different class C2. Therefore, ola(p_2)=4−2=2, and it means that p_2 does not belong to overlapping area. Additionally, if ola(p_i) > ola(p_j), p_i is located more deeply in an overlapping area than p_j.

The intuitive meaning of $R(F_k)$ is the ratio of overlapping areas of feature F_k. In the feature selection work, we choose features that have low value of $R(F_k)$ because low overlapping area brings high classification accuracy. If threshold θ is increased, $R(F_k)$ will be decreased. Based on the experiments, we found that the reasonable range of θ is $0 \le \theta \le k/2$.

2.2. Algorithm for calculating R(F$_k$)

Eq. (1) for calculating $R(F_k)$ can be implemented by following algorithm: Fig. 4.

3. Experiments

3.1. Data sets and classifiers

To test the RFS algorithm, we collected several types of datasets that have different numbers of features, classes, and instances. Multiple features and Madelon were obtained from the UCI Machine Learning Repository [20]. GD1027 [17], GD3175 [23], GD2547 [26], GD2545[13], DLBCL [24], Lung [25], Multi tissues [14–16], and BrcaEr [14,18,19] are microarray data.

Table 1 shows a summary of the datasets. We compared the RFS algorithm with FSDD, ReliefF, and mrmrMID on the datasets. The compared feature selection algorithms are up-to-date and produce higher accuracy than previous classical algorithms such as LDA. Parameters related to each feature selection algorithm are described in Table 2. For evaluation of the results of feature selection, we tested three classifiers, the *K*-nearest neighbor (KNN) [12], Naive Bayes (NB) [12], and support vector machine (SVM). We implemented FSDD, ReliefF, KNN, and NB in java according to the given algorithms in their original papers. In the case of mrmrMID, we downloaded c/c++ execution file from ‘http://penglab.janelia.org/proj/mRMR/index.htm’ and merged it with java. To test the SVM, we used the LIBSVM tool [8]. To avoid an overfitting problem, we adopted a *k*-fold cross validation [20], where *k* is 10. Random samples were chosen for each fold. The experiment was repeated five times, and we calculated the average classification accuracy. mrmrMID classification could not be tested on some of the datasets for its computational constraints. We describe testing condition related with classifiers in Table 3. Experiment results are presented in Section 3.2.

Table 1
Summary of datasets.

No	Data Set Name	Features	Classes	Instances	Fold	Ref
1	GD1027	15897	4	154	10	[17]
2	GD3715	12626	3	109	10	[23]
3	Multi tissues	1573	4	103	10	[14–16]
4	DLBCL	661	3	141	10	[24]
5	Lung	492	16	201	10	[25]
6	GD2547	12579	4	164	10	[26]
7	GD2545	12558	4	171	10	[13]
8	BrcaEr	754	2	146	10	[14,18,19]
9	Multiple features	649	10	2000	10	[20]
10	Madelon	500	2	2000	10	[20]

Table 2
Parameters related to feature selection algorithms.

Algorithm	Parameters
RFS	K=7, Θ=3
FSDD	ß=1
ReliefF	K=7, *repeat time*=(the number of instances)/2
mrmrMID	*threshold*=1, *selection method*=MID

Table 3
Testing condition related to classifiers.

Classifier	Testing condition
KNN	K=7, *distance function*=simple Euclidean distance
NB	For applying NB, we transform continuous data into discrete value. Interval is 0.01
SVM	*Kernel*=linear kernel

3.2. Classification accuracy

In this section, we describe the experimental results for classification accuracies obtained using three classifiers on eight datasets. We showed four accuracy graphs for first four datasets and four accuracy tables for second four datasets. In the case of GD2545, GD2546, and GD3715, mrmrMID goes over the bearable running time, and we did not obtain an accurate classification result. Therefore, the case produces no graph line or data value in accuracy graph. Fig. 5 shows the classification accuracy of three classifiers against the number of features for GD1027 data with the top 40 selected features. The KNN_GD1027, NB_ GD1027, and SVM_GD1027 graphs shown in Fig. 5 show the remarkable classification accuracy of RFS over the other methods. The accuracy graphs and tables show that RFS brings better classification accuracy than other methods in the case of KNN classifications (Tables 4–8, Figs. 6–9). For NB and SVM classification, other methods bring the best accuracy in many datasets. Previous studies [4,5] have shown that KNN classification induces better accuracy than other classification methods on the same dataset. Therefore, if a feature selection algorithm brings good classification accuracy using KNN, it is meaningful for the feature selection task. It is obvious that RFS is a strong feature selection method because it brings best classification accuracy using KNN in most of datasets.

3.3. Computational complexity

In the RFS algorithm, the most time consuming step is the sort array *V[]* and *C[]*. In the resent study, we used the Quick Sort [9] algorithm and a time complexity known as $O(n \log n)$. Loops require $n \times 2K$ computations because each point in a dataset requires $2K$ comparisons. If a dataset has m features, then total time complexity is $m \cdot O(n \log n + 2Kn)$. Table 9 summarizes the computation time of each feature selection algorithm. The java versions of the four algorithms were implemented on a 2.93 GHz 32 bits Pentium CPU. FSDD is the fastest algorithm, and RFS is second fastest. mrmrMID is the slowest algorithm and sometimes goes over the acceptable time. The RFS algorithm takes a reasonable time for huge datasets such as Multiple features and Madelon.

4. Discussions

The main goal of feature selection algorithms is to find good features that have clearly separated distribution of classes,

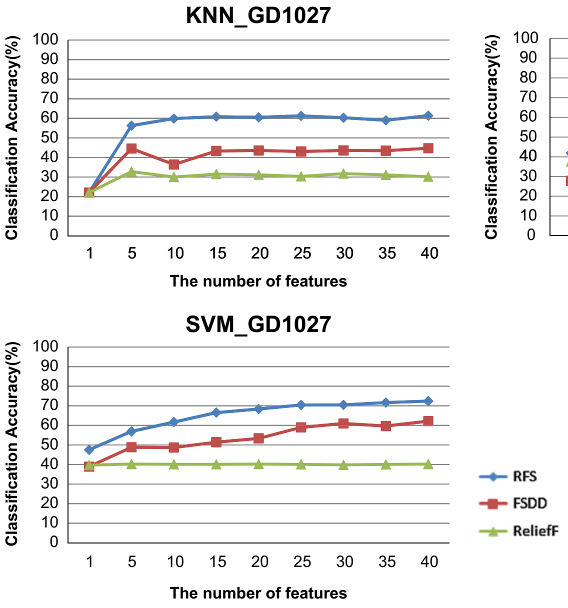


Fig. 5. Classification accuracy for the GD1027 dataset.

Table 4
Classification accuracy for the GD2547 dataset.

Classifier	mtds	*m*								
		1	5	10	15	20	25	30	35	40
KNN	RFS	15.25	61.02	61.66	60.45	63.25	64.58	66.20	65.89	65.33
	FSDD	15.25	55.22	57.89	59.13	57.10	57.68	57.60	56.95	57.69
	ReliefF	15.25	38.03	38.04	38.26	40.24	41.37	39.40	39.25	41.35
	mrmrMID									
NB	RFS	48.63	39.32	38.46	42.48	42.72	44.75	46.18	47.05	45.97
	FSDD	38.14	36.79	42.24	39.71	38.35	41.51	39.60	41.64	41.70
	ReliefF	35.48	35.84	35.99	37.13	34.13	35.43	33.00	32.63	29.63
	mrmrMID									
SVM	RFS	53.99	59.96	61.68	65.71	66.54	66.16	67.88	69.06	69.31
	FSDD	40.97	58.26	62.12	61.31	60.71	61.36	61.81	62.40	62.63
	ReliefF	37.99	37.99	39.52	38.54	39.53	41.36	40.70	42.04	42.39
	mrmrMID									

Table 5
Classification accuracy for the GD2545 dataset.

Classifier	mtds	*m*								
		1	5	10	15	20	25	30	35	40
KNN	RFS	14.59	58.42	59.86	59.77	61.99	61.29	63.62	62.33	63.01
	FSDD	14.59	45.06	46.33	49.72	47.97	48.67	48.68	49.37	49.60
	ReliefF	14.59	41.32	44.80	48.10	46.71	50.20	48.92	48.44	49.00
	mrmrMID									
NB	RFS	48.88	46.93	46.90	50.53	54.51	56.12	56.14	55.89	56.37
	FSDD	44.44	39.44	43.64	42.44	35.44	36.20	37.48	41.10	39.79
	ReliefF	32.86	39.09	41.45	41.65	44.25	44.62	43.07	43.78	44.59
	mrmrMID									
SVM	RFS	47.02	55.80	59.67	63.25	64.56	65.95	66.55	66.19	66.43
	FSDD	38.13	52.36	58.20	59.24	59.94	60.88	59.71	61.84	65.10
	ReliefF	31.94	31.12	31.35	31.94	32.06	32.89	34.76	34.18	35.12
	mrmrMID									

through different approaches. mrmrMID's approach is to consider relationship between features. Table 10 summarizes conditions of good features for three feature selection algorithms, except mrmrMID. The reason RFS shows the best performance is as follows:

(1). Classification accuracy has a strong relationship with the size of overlapping area in a dataset [7].

(2). Other algorithms try to capture overlapping area through indirect ways, whereas RFS captures it directly.

Table 6
Classification accuracy for the BrcaEr dataset.

Classifier	mtds	*m*								
		1	5	10	15	20	25	30	35	40
KNN	RFS	46.53	87.11	88.24	87.94	88.37	88.78	87.83	87.70	87.56
	FSDD	46.53	72.89	74.39	78.63	79.56	80.10	81.91	83.04	82.63
	ReliefF	46.53	86.05	87.14	86.71	87.01	86.86	86.45	87.54	86.72
	mrmrMID	46.53	86.85	86.18	84.80	85.51	85.23	85.36	85.52	86.31
NB	RFS	78.44	88.08	85.70	83.13	78.89	75.28	75.42	76.61	74.43
	FSDD	49.49	55.43	57.28	65.42	68.86	75.47	75.91	79.86	78.76
	ReliefF	75.76	85.20	85.12	83.84	83.32	82.79	82.17	79.33	78.39
	mrmrMID	78.57	55.02	59.76	61.81	61.71	65.13	68.81	68.56	69.50
SVM	RFS	89.24	88.79	88.37	88.10	88.11	87.84	87.02	87.27	86.45
	FSDD	59.64	78.74	81.07	81.36	83.01	83.56	83.59	83.17	82.60
	ReliefF	86.62	87.85	88.24	87.70	87.02	86.47	85.37	86.45	86.86
	mrmrMID	89.64	86.16	86.55	85.49	85.48	85.76	84.93	84.38	83.30

Table 7
Classification accuracy for the Multiple features dataset.

Classifier	mtds	*m*								
		1	5	10	15	20	25	30	35	40
KNN	RFS	10.00	87.43	92.60	94.01	94.72	95.21	95.18	95.20	95.22
	FSDD	10.00	63.52	86.53	87.80	88.17	89.39	90.43	91.45	92.00
	ReliefF	10.00	41.27	76.71	82.13	84.68	85.88	87.28	88.24	89.47
	mrmrMID	10.00	83.15	93.85	94.88	95.56	95.42	95.45	95.38	95.74
NB	RFS	48.29	63.74	70.81	71.95	69.36	68.40	68.09	67.59	68.15
	FSDD	25.09	35.63	32.87	34.58	33.32	33.66	33.13	33.64	37.12
	ReliefF	24.73	50.64	46.14	42.56	38.70	37.84	37.03	36.84	36.76
	mrmrMID	36.34	50.36	61.54	70.60	69.54	67.11	66.93	68.57	69.91
SVM	RFS	37.78	89.23	93.57	95.39	96.72	97.34	97.27	97.53	97.75
	FSDD	22.88	70.74	85.81	88.60	89.24	90.59	92.39	93.49	94.31
	ReliefF	24.56	64.18	81.09	85.00	87.54	89.53	90.91	91.76	92.43
	mrmrMID	31.57	86.12	94.96	96.33	97.11	97.17	97.32	97.42	97.88

Table 8
Classification accuracy for the Madelon dataset.

Classifier	mtds	*m*								
		1	5	10	15	20	25	30	35	40
KNN	RFS	50.00	71.29	78.27	76.27	76.40	76.48	76.45	75.41	74.12
	FSDD	50.00	49.94	50.03	51.01	51.29	52.15	51.26	51.45	52.40
	ReliefF	50.00	67.26	75.29	81.26	81.00	81.29	79.79	78.70	77.98
	mrmrMID	50.00	54.06	53.67	55.37	54.27	55.37	55.10	55.08	55.20
NB	RFS	59.64	56.18	51.89	50.67	50.44	50.46	50.39	50.02	49.39
	FSDD	49.90	46.88	48.01	48.43	48.87	49.72	49.07	48.88	48.19
	ReliefF	58.91	54.40	51.31	50.99	50.79	50.52	51.12	50.99	50.79
	mrmrMID	55.88	52.35	52.17	52.03	52.48	51.74	51.52	51.45	50.61
SVM	RFS	60.09	61.39	61.77	61.74	61.61	61.58	61.53	61.11	61.10
	FSDD	47.92	48.77	54.44	56.68	56.59	56.11	56.76	59.67	59.51
	ReliefF	61.00	61.70	61.61	61.86	61.50	61.40	61.16	60.84	60.88
	mrmrMID	54.89	56.38	56.86	57.28	57.31	56.94	56.21	56.22	55.94

It is clear that if a feature has a perfectly separable distribution of classes, all algorithms produce same degree of evaluation value. However, in the case of overlapped distribution of classes, RFS gives an outstanding performance compared to the other algorithms. One of RFS advantages is it is less influenced by data distribution than FSDD and ReliefF, because they consider the whole dataset to calculate evaluation value of a feature, whereas RFS only considers data in the overlapped region.

The experimental results of the RFS show that it can be applied to feature selection task. Basically, RFS is a filter method that does not consider relationships among features. If we combine RFS with a non-filter method, we can expect better feature selection. For example, mrmrMID can be combined with RFS. Let's assume that we need to select *m* features from a dataset, then the combined method can be composed as follows:

(1). Select 2*m* features from whole features using the RFS method.

(2). Select *m* features from 2*m* features using the mrmrMID method.

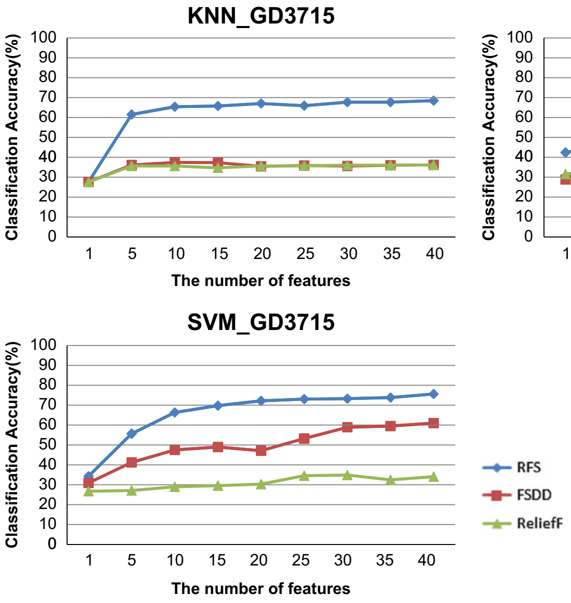


Fig. 6. Classification accuracy for the GD3715 dataset.

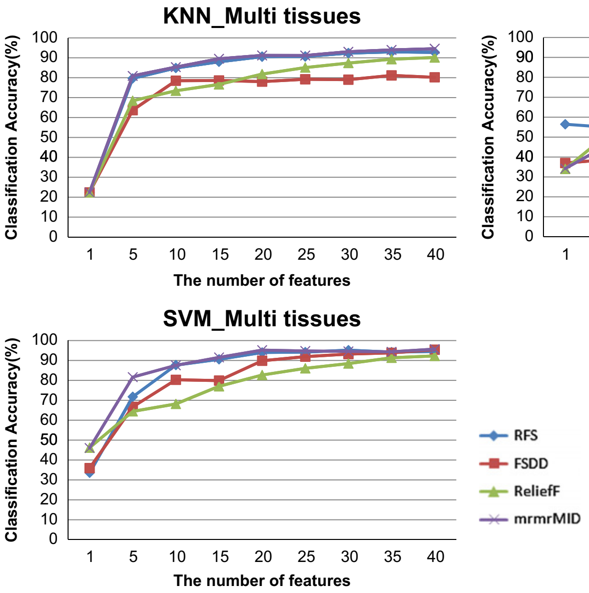


Fig. 7. Classification accuracy for the Multi tissues dataset.

The mrmrMID method considers relationships among features, relevance and redundancy. After producing candidate features using RFS, we can identify selected features that have maximum relevance and minimum redundancy using mrmrMID.

RFS has two parameters K and θ. If we choose 5–8 for K and under $k/2$ for θ, then they have little effect on the result of RFS. Therefore, choosing K and θ is not difficult task. In our experiment we choose 7 and 4 for K and θ respectively. Choosing an optimal number of K is same as choosing K for KNN algorithm. The large number of K may produce wide area of overlapped area. Accordingly, the optimal number of K makes relative overlapped areas of features to be maximized. This is a further topic of research.

Identifying the optimal number of features that produces the best classification accuracy is a difficult task, and RFS provides no information that facilitates this task. In general, classification accuracy is proportional to number of selected features and has to trade off against the running time of classification. We can increase number of selected features until the running time is reasonable.

5. Conclusions

We propose an efficient feature selection method based on overlapping areas among classes in a dataset. The proposed method had a reasonable time complexity and induced a high classification accuracy. In addition, it is easy to understand the mathematical background of the proposed method and it is also

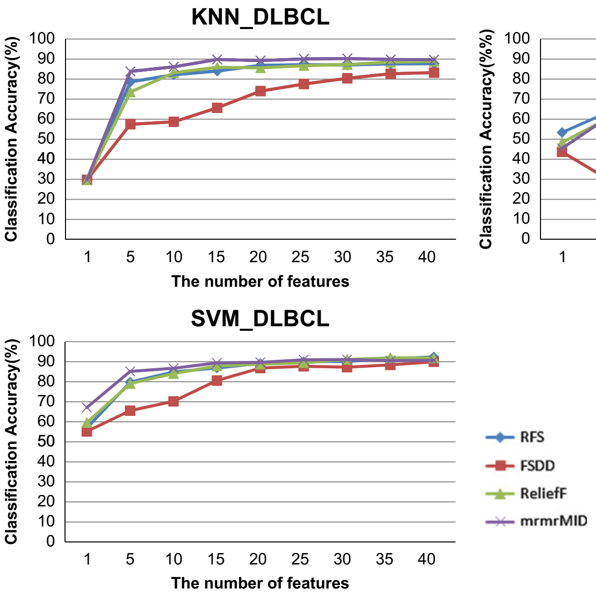


Fig. 8. Classification accuracy for the DLBCL dataset.

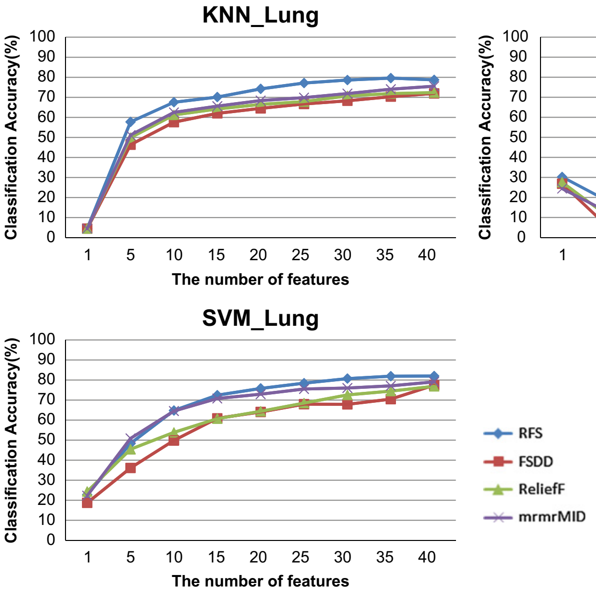


Fig. 9. Classification accuracy for the Lung dataset.

Table 9
Runtime time (s) of each feature selection algorithm.

Data	Method			
	RFS	FSDD	ReliefF	mrmrMID
GD1027	5.311	0.048	376.854	–
GD3715	2.950	0.027	126.134	–
Multi tissues	0.336	0.003	0.757	3.460
DLBCL	0.202	0.003	0.271	4.439
Lung	0.220	0.002	0.361	5.969
GD2547	4.499	0.041	253.616	–
GD2545	4.698	0.042	264.118	–
BrcaEr	0.219	0.003	0.337	4.662
Multiple features	4.033	0.030	23.339	30.196
Madelon	3.252	0.021	18.137	31.127

Table 10
Comparison of criteria of a good feature.

RFS	has small size of overlapping area
FSDD	far distance among different classes, close distance inside of a class
ReliefF	nearest neighbor of a sample should be found in the same class of the sample

easy to implement. We demonstrated the efficiency of the RFS method through an experiment involving eight datasets that had different numbers of features, classes, and instances. The RFS method can be combined with other feature selection methods. Finding and testing efficient combinations of these methods will be conducted in future studies.

Acknowledgment

This study was supported by Grant no. R31-2008-000-10069-0 from the World Class University (WCU) Project of the Ministry of

Education, Science & Technology (MEST) and the Korea Science and Engineering Foundation (KOSEF).

References

[1] Definition of feature selection. 〈http://en.wikipedia.org/wiki/Feature_selection〉.

[2] Y. Kim, W.N. Street, F. Menczer, Feature selection in data mining, in: W. John (Ed.), Data Mining, IGI Publishing, Hershey, 2003, pp. 80–105.

[3] D.P. Berrar, W. Dubitzky, M. Granzow, A Practical Approach to Microarray Data Analysis, Kluwer Academic Publishers, Norwell, MA, 2009, P. 1.

[4] J. Liang, S. Yang, A. Winstanley, Invariant optimal feature selection: a distance discriminant and feature ranking based solution, Pattern Recognition 41 (2008) 1429–1439.

[5] M. Robnik-Sikonja, I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, Mach. Learn. 53 (2003) 23–69.

[6] C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, in: Proceedings of the IEEE Computer Society Conference on Bioinformatics, IEEE Computer Society, 2003, p. 523.

[7] S. Oh, A new dataset evaluation method based on category overlap, Comput. Biol. Med. 41 (2011) 115–122.

[8] C. Chang, C. Lin, LIBSVM—A Library for Support Vector Machines. 〈http://www.csie.ntu.edu.tw/~cjlin/libsvm/〉.

[9] C.A.R. Hoare, Algorithm 64: Quicksort, Communications of the ACM 4 (7) (1961) 321.

[10] H. Peng, F. Long, 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) 1226–1238.

[11] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Trans. Knowl. Data Eng. 17 (2005) 491–502.

[12] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Second ed., Morgan Kaufmann Publishers Inc., Burlington, MA, 2005, p. 7.

[13] Q.-H. Ye, L.-X. Qin, M. Forgues, P. He, J.W. Kim, A.C. Peng, R. Simon, Y. Li, A.I. Robles, Y. Chen, Z.-C. Ma, Z.-Q. Wu, S.-L. Ye, Y.-K. Liu, Z.-Y. Tang, X.W. Wang, Predicting hepatitis B virus-positive metastatic hepatocellular carcinomas using gene expression profiling and supervised machine learning, Nat. Med. 9 (4) (2003) 416–423.

[14] Y. Hoshida, Nearest template prediction: a single-sample-based flexible class prediction with confidence assessment, PLoS ONE 5 (2010) e15543.

[15] A.I. Su, M.P. Cooke, K.A. Ching, Y. Hakak, J.R. Walker, T. Wiltshire, A.P. Orth, R.G. Vega, L.M. Sapinoso, A. Moqrich, A. Patapoutian, G.M. Hampton, P.G. Schultz, J.B. Hogenesch, Large-scale analysis of the human and mouse transcriptomes, Proc. Nat. Acad. Sci. USA 99 (2002) 4465–4470.

[16] Y. Hoshida, J.-P. Brunet, P. Tamayo, T.R. Golub, J.P. Mesirov, Subclass mapping: identifying common subtypes in independent disease data sets, PLoS ONE 2 (2007) e1195.

[17] S. Di Giovanni, S.M. Knoblach, C. Brandoli, S.A. Aden, E.P. Hoffman, A.I. Faden, Gene profiling in spinal cord injury shows role of cell cycle in neuronal death, Ann. Neurol. 53 (4) (2003) 454–468.

[18] L. van 't Veer, H. Dai, M. van de Vijver, Y. He, A. Hart, M. Mao, H. Peterse, K. van der Kooy, M. Marton, A. Witteveen, G. Schreiber, R. Kerkhoven, C. Roberts, P. Linsley, R. Bernards, S. Friend, Gene expression profiling predicts clinical outcome of breast cancer, Nature 415 (2002) 530–536.

[19] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J.A. Olson, J.R. Marks, J.R. Nevins, Predicting the clinical status of human breast cancer by using gene expression profiles, Proc. Nat. Acad. Sci. USA 98 (2001) 11462–11467.

[20] UCI Machine Learning Repository. 〈http://archive.ics.uci.edu/ml/〉.

[21] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. 7 (1936) 179–188.

[22] Y. Saeys, I. Inza, P. Larranaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.

[23] A. Whitehead, D.L. Crawford, Variation in tissue-specific gene expression among natural populations, Genome Biol. 6 (2) (2005) R13.

[24] U.R. Chandran, C. Ma, R. Dhir, M. Bisceglia, M. Lyons-Weiler, W. Liang, G. Michalopoulos, M. Becich, F.A. Monzon, Gene expression profiles of prostate cancer reveal involvement of multiple molecular pathways in the metastatic process, BMC Cancer 7 (2007) 64.

[25] C.F. Ockenhouse, W.B. Bernstein, Z. Wang, M.T. Vahey, Functional genomic relationships in HIV-1 disease revealed by gene-expression profiling of primary human peripheral blood mononuclear cells, J. Infect. Dis. 191 (12) (2005) 2064–2074.

[26] J.F. Dillman III, C.S. Phillips, L.M. Dorsch, M.D. Croxton, A.I. Hege, A.J. Sylvester, T.S. Moran, A.M. Sciuto, Genomic analysis of rodent pulmonary tissue following bis-(2-chloroethyl) sulfide exposure, Chem. Res. Toxicol. 18 (1) (2005) 28–34.

Jimin Lee received his Bachelor degree in Computer Science from Dankook University, Korea, in 2011. She is currently a M.S. student in the Department of NanoBioMedical Science at Dankook University. She is also a researcher at WCU Research Center of NanoBioMedical Science. Her main research interests are machine learning algorithms and bioinformatics.

Nomin Batnyam received her Bachelor degree in Information Technology from Mongolian International University, Mongolia, in 2010. She is currently a M.S. student in the Department of NanoBioMedical Science at Dankook University. She is also a researcher at WCU Research Center of NanoBioMedical Science. Her main research interests are machine learning algorithms and bioinformatics.

Sejong Oh received a Doctor, Master, and Bachelor degree in Computer Science from Sogang University, Korea, in 2001, 1991, and 1989, respectively. From 2001 to 2003, he was a Postdoctoral Fellow in the Laboratory for Information Security Technology at George Mason University, USA. Since 2003 he joined the Department of Computer Science at Dankook University, Korea, and is currently Associate Professor in WCU Research Center of NanoBioMedical Science. His main research interests are bioinformatics, information system, and information system security.