Hyehyun Jeon 1 and Sejong Oh 2,*

1 Department of Data Science, Dankook University, Yongin 16890, Korea; wjsur1028@gmail.com
2 Department of Software Science, Dankook University, Yongin 16890, Korea; sejongoh@dankook.ac.kr
* Correspondence: sejongoh@dankook.ac.kr
Received: 20 April 2020; Accepted: 3 May 2020; Published: 4 May 2020

Abstract: As datasets continue to increase in size, it is important to select the optimal feature subset from the original dataset to obtain the best performance in machine learning tasks. Highly dimensional datasets that have an excessive number of features can cause low performance in such tasks. Overfitting is a typical problem. In addition, datasets that are of high dimensionality can create shortages in space and require high computing power, and models fitted to such datasets can produce low classification accuracies. Thus, it is necessary to select a representative subset of features by utilizing an efficient selection method. Many feature selection methods have been proposed, including recursive feature elimination. In this paper, a hybrid-recursive feature elimination method is presented which combines the feature-importance-based recursive feature elimination methods of the support vector machine, random forest, and generalized boosted regression algorithms. From the experiments, we confirm that the performance of the proposed method is superior to that of the three single recursive feature elimination methods.

Keywords: classification; feature selection; machine learning; recursive feature elimination

1. Introduction

As datasets continue to increase in size, it is important to select the optimal subset of features from a raw dataset in order to obtain the best possible performance in a given machine learning task. An efficient and small feature (variable) subset is especially important for building a classification model. High-dimensionality datasets can easily cause overfitting problems, in which case a reliable model cannot be obtained. Furthermore, such datasets require high computing power and large volumes of storage space [1], and often produce models with low classification accuracy. This is called the “curse of dimensionality” [2]. Thus, it is necessary to select a representative subset of features to solve these problems.

Feature selection [3–6] has become necessary to select the best subset of features, and is used in various fields including biology, medicine, finance, manufacturing and production, and image processing. Recursive feature elimination (RFE) is a feature selection method that attempts to select the optimal feature subset based on the learned model and classification accuracy. Traditional RFE sequentially removes the worst feature that causes a drop in “classification accuracy” after building a classification model. A new RFE approach was recently proposed which evaluates “feature (variable) importance” instead of “classification accuracy” based on a support vector machine (SVM) model, and chooses the least important features for elimination [7]. This approach can be applied to other classification models such as random forests (RFs) and gradient boosting machines (GBMs), both of which have in-built feature evaluation mechanisms.

In Figure 1, the overall process for selecting features using the feature-importance-based RFE method is shown. When a classifier is trained with a training dataset, feature weights that reflect the importance of each feature can be obtained. After all features are ranked according to their weights,
the feature that has the lowest weight value is removed. Then, the classifier is re-trained with the remaining features until it has no features left with which to train [7]. Finally, the entire ranking of the features using the feature-importance-based RFE method can be obtained. This approach is an embedded feature selection method [7] and has been shown to provide strong performance, compensating for the weaknesses of the filter [8] and wrapper methods [9].

![Diagram](image)

**Figure 1.** Process of feature-importance-based recursive feature elimination (RFE). N denotes the number of features.

Guyon [7] proposed an SVM-based RFE (SVM-RFE) algorithm to evaluate the importance of each feature. The SVM aims to find the hyperplane that divides classes the most. Kernel techniques in the SVM ensure that high-dimensional data are well separated. The values of the feature weight vector are obtained using the linear kernel, and the weight values are used to evaluate the importance of the features. Suppose $D(x)$ is the decision function for the hyperplane, and $c$ represents the number of classes. If the data has multiple classes (i.e., more than two), $q$, which denotes the entire number of hyper-planes, is calculated according to the equation $q = c(c - 1)/2$. Equation (1) refers to the decision function for the binary class case, and Equation (2) is for a multi-class dataset.

$$D(x) = \text{sign}(x \cdot w)$$  \hspace{1cm} (1) \\

$$D(x) = \text{sign}(x \cdot w_j), \ j = 1,2,3,...,q$$  \hspace{1cm} (2)
In the linear decision function, \( x \) denotes a vector with the components of a given spectrum, and \( w \) is a vector perpendicular to the hyperplane providing a linear decision function [10]. One of the main ideas of the SVM is that the decision boundary that separates the classes is defined by specific observations called support vectors. The weighted vector indicates the importance of the different variables to the decision function. Because the weighted vector is located where the decision boundary has the maximum margin, if the weighted vector has a large value for a particular feature, it indicates that this feature can separate the classes clearly. Equation (3) is used to obtain the weight value for the evaluation of variable importance according to SVM-RFE [10].

\[
W_S = \frac{1}{q} \sum_{i} w_i
\]  

(3)

Compared with other feature selection methods, SVM-RFE is a scalable, efficient wrapper method [11], and is widely applied in bioinformatics [12–15]. Recently, Duan [11] also developed multiple SVM-RFE.

The overall process of RF-based RFE (RF-RFE) is similar to that of SVM-RFE. RF, an ensemble method, is a representative bagging algorithm that has been shown to perform well in terms of predictive accuracy. The RF algorithm calculates the feature importance from the training model and employs two methods to measure variable importance. The mean decrease in accuracy (MDA) shows how much the model accuracy decreases from permuting the values of each variable. The mean decrease in Gini (MDG) is the mean of a variable’s total decrease in node impurity, weighted by the proportion of observations reaching that node in each individual decision tree. We used MDA measurement to implement RF-RFE in this work. Suppose \( B \) denotes the out-of-bag (OOB) observations of a tree \( t \), and \( VI \) indicates the importance of variable \( X_i \) in tree \( t \). Equation (4) shows the measure of MDA [16,17].

\[
W_R(X_i) = \frac{\sum_{b \in B} VI^{(b)}(X_i)}{ntree}
\]  

(4)

GBM-based RFE (GBM-RFE) uses the gradient boosting algorithm to train the classifier in the RFE method. GBM uses boosting, which is another representative ensemble method. The idea of boosting is to train weak learners sequentially, each of which tries to correct its predecessor. A weak learner can be defined as a classifier that is only slightly correlated with the true classification. To transform the weak learners into strong learners, the residual errors of the previous model are used as the weight values. GBM uses gradients in the loss function, which is a measure indicating how good the model’s coefficients are at fitting the underlying data. The Gini index, which uses frequency to evaluate the accuracy of a tree-based algorithm, is used to evaluate feature importance \( (W_G) \) in this study. If the Gini index has a large value, it means that the feature is important. The class variable is denoted as \( c \), and \( p_j \) denotes the ratio of the number of observations in each class at a given node.

\[
W_G = 1 - \sum_{j=1}^{c} p_j^3
\]  

(5)

In this study, we propose a new feature selection method — Hybrid-RFE — that is an ensemble of the feature evaluation methods of SVM-RFE, RF-RFE, and GBM-RFE, combining their feature weighting functions. We suggest two combinations: the simple sum and weighted sum. From the experiments, we confirm that Hybrid-RFE with the weighted sum shows the best performance.

2. Materials and Methods
2.1. Idea of Hybrid Recursive Feature Elimination

In machine learning tasks, ensembles of different methods may produce higher performance than each individual method. Therefore, we can expect that an ensemble of RFEs will produce a better performance than a single RFE. Hybrid-RFE is an ensemble algorithm for feature selection that combines the SVM-RFE, RF-RFE, and GBM-RFE methods. Hybrid-RFE has two types of weighting functions. The first type is the simple sum of the feature weights from the three RFE methods. The second type reflects both feature weights and model accuracies from the three RFE methods.

Each feature weight obtained from the three RFE methods should be normalized before combining because their scales of weight values are different. The weighting functions of Hybrid-RFE are summarized in Figure 2.

![Figure 2. Two types of weighting functions of Hybrid-RFE.](image)

2.2. Hybrid Method: Simple Sum

The simple sum function—$W_{\text{clus}}$—is calculated by the sum of the normalized weights—$\bar{W}_S$, $\bar{W}_R$, and $\bar{W}_G$—from SVM-RFE, RF-RFE, and GBM-RFE, respectively. Equation (6) describes the $W_{\text{clus}}$ function. In the equation, $X$ is a feature set to be evaluated. The weight values $w$ in a weight vector $W$ are normalized by Equation (7):

$$W_{\text{clus}}(X) = \bar{W}_S(X) + \bar{W}_R(X) + \bar{W}_G(X)$$  \hspace{2cm} (6)

$$\text{Normalized } w = (w - \min(W)) / (\max(W) - \min(W))$$  \hspace{2cm} (7)

2.3. Hybrid Method: Weighted Sum

The weighted sum function considers both the weight values and model accuracy. Model accuracy refers to the classification accuracy obtained from the model. For this process, the given dataset is divided into a training set and test set. The training set is used to build a model and obtain the weight values for the features. The test set is used to obtain the classification accuracy of the model. Equation (8) describes the weighted sum function $W_{\text{clus}}$. In this equation, $\text{SVM.acc}$, $\text{RF.acc}$, and $\text{GBM.acc}$ represent the model accuracy of SVM-RFE, RF-RFE, and GBM-RFE, respectively.

$$W_{\text{clus}}(X) = \bar{W}_S(X) \times \text{SVM.acc} + \bar{W}_R(X) \times \text{RF.acc} + \bar{W}_G(X) \times \text{GBM.acc}$$  \hspace{2cm} (8)

3. Results
3.1. Datasets

To compare the proposed method with the previous RFE methods, we chose eight benchmark datasets from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/index.php) and the NCBI Gene Expression Omnibus repository (https://www.ncbi.nlm.nih.gov/geo/). The benchmark datasets are summarized in Table 1. The first four datasets are from UCI, and the rest are from NCBI. The UCI datasets contain a relatively small number of features. The NCBI datasets have numerous features and small numbers of observations (samples). Each feature contains the gene expression values of a specific gene. These are typical datasets that require feature selection as they contain more than 20,000 features. To save computing time, we randomly selected 1000 features from them.

| No. | Dataset | # of Observations | # of Features | # of Classes |
|-----|---------|-------------------|---------------|--------------|
| 1   | satellite | 6435              | 36            | 6            |
| 2   | messidor | 351               | 33            | 2            |
| 3   | sonar    | 208               | 60            | 2            |
| 4   | messidor | 1151              | 20            | 2            |
| 5   | GDS1027  | 154               | 1000          | 4            |
| 6   | GDS2546  | 167               | 1000          | 4            |
| 7   | GDS2547  | 164               | 1000          | 4            |
| 8   | GDS3715  | 109               | 1000          | 3            |

3.2. Experiments

We compared the previous RFE methods (SVM-RFE, RF-RFE, GBM-RFE) and the proposed Hybrid-RFE method. Hybrid-RFE was tested using simple sum and weighted sum functions. All experiments were performed using R language (http://www.r-project.org). The SVM-RFE code was referenced from multiple SVM-RFE [11], and we implemented the code for the others as modified versions of SVM-RFE (See Supplementary Materials).

The performance evaluation metric for each method was the classification accuracy obtained on the selected features from the RFE methods. Four classification algorithms were used: k-nearest neighbor (KNN), SVM, RF, and naïve Bayes (NB). The default tuning parameters were used for each classifier. Table 2 lists these classifiers and their corresponding R packages.

Five-fold cross-validation was used not only to evaluate the accuracy of each classifier after feature selection, but also to perform the RFE process. In other words, the final feature importance list of an RFE method was calculated by averaging the five feature importance lists from each cross-validation fold.

All of our RFE algorithms have a “halve.aboves” parameter. If the number of features is larger than the parameter value, RFE removes half of the least important features from the feature list. This approach was suggested by [11] as a means to speed up the feature selection process. In the regular RFE process, only the least important feature was removed from the feature list. We set the “halve.aboves” parameter to 200.

| No. | Classifier | R Package |
|-----|------------|-----------|
| 1   | GBM        | caret      |
| 2   | SVM        | e1071      |
| 3   | RF         | randomForest|
| 4   | KNN        | e1071      |

Table 2. Classifiers and their corresponding R packages.
| Classifier | R package |
|------------|-----------|
| KNN        | class     |
| SVM        | e1071     |
| RF         | randomForest |
| NB         | e1071     |

### 3.3. Experimental Results

Using the feature importance ranking list from the three previous RFE methods and two proposed RFE methods, four classifiers were tested to find the best feature set that produced the highest classification accuracy. The results of the basic experiment are summarized in Table 3. Each cell expresses the highest classification accuracy, and the number of selected features is shown in parentheses next to the accuracy value. For example, 0.904 (36) from SVM-RFE and KNN indicates that the SVM classifier produced the highest accuracy of 0.904 when it used the first 36 features from the SVM-RFE feature ranking list. The data in bold in the table represent the top values for a specific classifier. If two or more methods produced equal top accuracy, the method that used the least number of selected features was chosen.

From Table 3, it is evident that the top values were mainly produced by the two proposed methods. This means that the feature ranking lists generated by the proposed methods were better than those generated using the previous methods. The frequency of top accuracies for the compared RFE methods are summarized in Table 4. The frequencies of Hybrid-RFE with the simple sum function and weighted sum were 10 and 11, respectively (total of 21). The frequencies of SVM-RFE, RF-RFE, and GBM-RFE were 2, 4, and 4, respectively. In our experiment, Hybrid-RFE with the weighted sum function showed the best performance, whereas SVM-RFE showed the worst performance.

If two RFE methods A and B produce the same classification accuracies and the numbers of selected features are 15 and 10, respectively, method B is more efficient than method A. The ideal feature selection method produces a small feature subset size and high classification accuracy. Therefore, we compared the average size of the set of selected features from the five RFE methods. The results are summarized in Table 5. Only the GDS datasets were used in the comparison because the others have a small number of features in the original datasets. The proposed Hybrid-RFE methods were more efficient than previous RFE methods. In particular, Hybrid-RFE with a weighted sum function produced a remarkably small feature subset.

**Table 3. Experimental results.**

| Dataset | Algorithm | Accuracy on entire feature set | RFE Methods |
|---------|-----------|--------------------------------|-------------|
|         | SVM-RFE   | RF-RFE                         | GBM-RFE     | Hybrid-RFE (SS) 1 | Hybrid-RFE (WS) 2 |
| satellite | KNN       | 0.904 (36)                     | 0.904 (36) | 0.904 (36) | **0.907 (31)** | 0.904 (31) |
|         | SVM       | 0.898 (27)                     | 0.898 (36) | 0.898 (36) | 0.898 (35) | 0.899 (31) |
|         | RF        | 0.918 (36)                     | 0.919 (36) | 0.919 (36) | 0.919 (36) | **0.919 (33)** |
|          | KNN   | SVM   | RF    | NB    |
|----------|-------|-------|-------|-------|
| ionosphere | 0.855 | 0.949 | 0.923 | 0.724 |
|          | 0.904 | 0.951 | 0.938 | 0.781 |
|          | 0.915 | 0.939 | 0.930 | 0.856 |
|          | 0.901 | 0.949 | 0.929 | 0.808 |
|          | 0.906 | 0.952 | 0.932 | 0.849 |
|          | 0.918 | 0.906 | 0.933 | 0.835 |
| sonar    | 0.815 | 0.824 | 0.811 | 0.753 |
|          | 0.856 | 0.815 | 0.833 | 0.746 |
|          | 0.831 | 0.821 | 0.820 | 0.769 |
|          | 0.859 | 0.836 | 0.833 | 0.753 |
|          | 0.854 | 0.843 | 0.846 | 0.770 |
|          | 0.840 | 0.825 | 0.850 | 0.763 |
| messidor | 0.615 | 0.710 | 0.690 | 0.610 |
|          | 0.647 | 0.710 | 0.702 | 0.610 |
|          | 0.644 | 0.710 | 0.701 | 0.610 |
|          | 0.637 | 0.710 | 0.697 | 0.610 |
|          | 0.652 | 0.710 | 0.689 | 0.610 |
|          | 0.633 | 0.710 | 0.699 | 0.610 |
| GDS1027  | 0.812 | 0.565 | 0.623 | 0.475 |
|          | 0.802 | 0.552 | 0.633 | 0.511 |
|          | 0.807 | 0.547 | 0.633 | 0.528 |
|          | 0.803 | 0.540 | 0.528 | 0.530 |
|          | 0.833 | 0.532 | 0.530 | 0.537 |
|          | 0.825 | 0.545 | 0.537 | 0.572 |
| GDS2546  | 0.629 | 0.706 | 0.719 | 0.711 |
|          | 0.667 | 0.730 | 0.712 | 0.743 |
|          | 0.697 | 0.712 | 0.725 | 0.729 |
|          | 0.661 | 0.731 | 0.725 | 0.740 |
|          | 0.685 | 0.727 | 0.718 | 0.746 |
|          | 0.668 | 0.732 | 0.742 | 0.748 |
4. Discussion

There is no “super” feature selection method for every dataset. Each dataset has unique characteristics that influence the working mechanisms of the feature selection method. Therefore, several feature selection methods need to be tested. Furthermore, there is no relationship between classifiers and feature selection methods—it is not known which feature selection method is best for SVM or which classifier is best for GBM-RFE. Therefore, it is necessary to check combinations of classifiers and feature selection methods in order to build a high-performance classification model. A small number of good classifiers and feature selection methods are required in order to reduce the experimentation time.

The proposed Hybrid-RFE method has been shown to provide better performance than previous RFE methods. It can thus be considered first if feature selection is required. The key point of Hybrid-RFE is the weighting function that combines different RFE weights. If we can find a more efficient weighting function, Hybrid-RFE can be improved. Hybrid-RFE combines the normalized weights of different RFEs and adopts “local” normalization. If we could develop “global” normalization, feature weights from several RFEs could be more effectively combined. Furthermore, feature interaction and feature dependency are important factors for feature selection. We have not yet found a way to
measure these factors and combine them with feature importance. These are potential topics for further research.

Supplementary Materials: The source code of Hybrid-RFE and a sample application are available online at https://bitldku.github.io/home/sw/Hybrid_RFE.html.

Author Contributions: methodology, S. Oh; software, H. Jeon; validation, S. Oh; formal analysis, S. Oh; investigation, H. Jeon; resources, H. Jeon; data curation, S. Oh, H. Jeon; writing — original draft preparation, H. Jeon; writing — review and editing, S. Oh; visualization, S. Oh; supervision, S. Oh; project administration, S. Oh; funding acquisition, S. Oh. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the ICT & RND program of MIST/IITP, [2018-0-00242, Development of AI ophthalmologic diagnosis and smart treatment platform based on big data].

Conflicts of Interest: The authors declare no conflicts of interest.

References
1. Tang, J.; Alelyani, S.; Liu H. Feature selection for classification: A review, Chapman and Hall/CRC, New York, USA, 2014. pp. 37-64 doi: https://doi.org/10.1201/b17320
2. Bellman, R. Dynamic Programming, 6th ed.; Princeton Univ. Press; Princeton, N.J., USA, 1957; p.4.
3. Kumar, V.; & Minz, S. Feature selection: a literature review. SmartCR 2014, Vol.4 No.3, pp.211-229.
4. Tang, J., Alelyani, S., & Liu, H. Feature selection for classification: A review. Data class: Algor appl 2014, vol. 37, pp1-29.
5. Bolón-Canedo, V., Sánchez-Marono, N., Alonso-Betanzos, A., Benítez, J. M., & Herrera, F. A review of microarray datasets and applied feature selection methods. Inform Sci 2014, vol.282, pp.111-135.
6. Ang, J. C., Mirzal, A., Haron, H., & Hamed, H. N. A. Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection. IEEE/ACM trans comput biol bioinfo 2015, Vol.13, No. 5, pp. 971-989.
7. Saeyes, Y.; Inza, I.; Larrañaga P. A review of feature selection techniques in bioinformatics. Bioinformatics 2007, Vol.23, No.19, pp.2507-2517.
8. Liu, H.; Motoda, H. Computational Methods of Feature Selection, Chapman and Hall/CRC, New York, USA, 2007.
9. Liu, H.; Yu, L. Toward integrating feature selection algorithms for classification and clustering, IEEE T Knowl Data En 2005; Vol.17, No.4, pp.491.
10. Guyon, I.; Weston, J.; Barnhill, S.; Vapnik, V. Gene selection for cancer classification using support vector machines, Mach Learn 2002; Vol.46, No.1-3, pp.389-422.
11. Duan, K.B.; Rajapakse, J.C.; Wang, H.; Azuaje, F. Multiple SVM-RFE for gene selection in cancer classification with expression data. IEEE T Nanobiosci 2005; 4, 228-234.
12. Mundra, P.A.; Rajapakse, J.C. SVM-RFE with MRMR filter for gene selection. IEEE T Nanobiosci 2009; Vol. 9, No, 4, pp.31-37.
13. Zhou, X.; Tuck, D.P. MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data. Bioinformatics 2007; Vol. 23, No. 9, pp.1106-1114.
14. Tang, Y.; Zhang, Y.Q.; Huang, Z. Development of two-stage SVM-RFE gene selection strategy for microarray expression data analysis. IEEE/ACM T Comp Biol Bioinformatics, 2007; Vol. 4, No. 3, pp.365-381.
15. Ding, Y.; Wilkins, D. Improving the performance of SVM-RFE to select genes in microarray data. BMC bioinformatics 2006; Vol. 7, No. S2, p. S12.
16. Granitto, P.M.; Furlanello, C.L.; Biasioli, F.; Gasperi, F. Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products, Chemometr Intell Lab Syst 2006; Vol.83, No.2, pp.83-90.
17. Hierpe, A. Computing Random Forests Variable Importance Measures (VIM) on Mixed Continuous and Categorical Data, Master’s Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, (2016)