A Novel RFE-SVM-based Feature Selection Approach for Classification

Mouhamadou Lamine Samb[1], Fod éCamara[1], Samba Ndiaye[1], Yahya Slimani[2] and Mohamed Amir Esseghir[3]

[1] Department of Mathematics and Computer Science, Cheikh Anta Diop University, Dakar, Senegal.
[2] Department of Computer Science, Faculty of Sciences of Tunis, 1060 Tunis, Tunisia.
[3] Artois University, Faculty of Applied Sciences of Bethune, Technopark Futura, 62400, France.
{fode.camara, mouhamadou81.samb, samba.ndiaye}@ucad.edu.sn, yahya.slimani@fst.rnu.tn, mohamedemir@yahoo.fr

Abstract

The feature selection for classification is a very active research field in data mining and optimization. Its combinatorial nature requires the development of specific techniques (such as filters, wrappers, genetic algorithms, simulated annealing, and so on) or hybrid approaches combining several optimization methods. In this context, the support vector machine recursive feature elimination (SVM-RFE), is distinguished as one of the most effective methods. However, the RFE-SVM algorithm is a greedy method that only hopes to find the best possible combination for classification. To overcome this limitation, we propose an alternative approach with the aim to combine the RFE-SVM algorithm with local search operators based on operational research and artificial intelligence. To assess the contributions of our approach, we conducted a series of experiments on datasets from UCI Machine Learning Repository. The experimental results which we obtained are very promising and show the contribution of the local search on the classification process. The main conclusion is that the reuse of features previously removed during the RFE-SVM process improves the quality of the final classifier.

Keywords: *Classification, Supervised classification, Feature selection, Support Vector Machines, Recursive Feature Elimination, Local search operators*

1. Introduction

Feature selection has been an active research area in data mining communities because it allows significantly improving the comprehensibility of the resulting classifier models [1]. It consists to choose a subset of input variables from a dataset with very large of attributes by eliminating features with little or no predictive information. For example, cancer microarray data normally contains a small number of samples which have a large number of gene expression levels as features. In order to extract useful gene information from cancer microarray data and reduce dimensionality, several feature selection algorithms were systematically investigated [2, 3]. In the literature, three different trends of methods are identified: the filter methods, the wrapper methods and the embedded methods [4, 5].

In this paper, we specifically studied the RFE-SVM algorithm [2, 6, 3, 7] which is a wrapper method for feature selection method using Support Vector Machines. RFE-SVM method ranks all the features according to some score function and eliminates one or more features with the lowest scores. This process is repeated until the highest classification accuracy is obtained. Due to its successfully use in selecting informative genes for cancer classification, SVM-RFE gained a great popularity and is well known as one of the most effective feature selection method [2, 6, 3, 7]. However, the RFE-SVM is a greedy method that only hopes to find the best possible combination for classification. To overcome this limitation, we proposed a new feature selection approach which combines the RFE-SVM algorithm with local search. We experiment it using Ionosphere, Spam-Base, and Spect Heart Data datasets from UCI repository [8]. The experimental results show that the reuse of features previously removed during the RFE-SVM process can improve the quality of the final classifier.

The remainder of this paper is organized as follows. The Section 2 gives the state-of-the-art in feature selection. In section 3, we review local search, SVM and SVM-RFE-based feature selection methods. Our proposed approach is presented in section 4. Section 5 presents the experiment results. Finally, Section 6 concludes with a discussion of the contributions of our proposal and our current research plans.

2. Feature selection: basics and background

Feature selection techniques have become an apparent need in many applications for which datasets with tens or hundreds of thousands of variables are studied [4]. As formulated in [4, 9, 10, 11], it is motivated by the following reasons:

- better predictive performance;
- computational efficiency from working with fewer inputs;
- cost savings from having to measure fewer features;
- and simpler, more intelligible models.

Several feature selection approaches are proposed in the literature with the aim to find the best trade-off between these competitive goals.

As defined in [4], feature selection, also known as variable selection, feature reduction, attribute selection or variable subset selection is a process that selects a subset of original features. The optimality of a feature subset is measured by an evaluation criterion. As the dimensionality of a domain expands, the number of features N increases. Finding an optimal feature subset is usually intractable [4] and many problems related to feature selection have been shown to be NP-hard [4, 12]. A typical feature selection process consists of four basic steps (shown in 1), namely, subset generation, subset evaluation, stopping criterion, and result validation [4, 11]. The main step is the subset generation which is a process with two basic issues:

1. According to the search starting point, we can distinguish three main approaches: (i) the forward selection which start with no variables and add them one by one, at each step adding the one that decreases the error the most, until any further addition does not significantly decrease the error; (ii) the backward selection which start with all the variables and remove them one by one, at each step removing the one that decreases the error the most, until any further

removal increases the error significantly; (iii) the stepwise selection which start with both ends and add or remove features simultaneously.

2. According to the search strategy, we can distinguish the complete search, the sequential search and the random search. The complete search guarantees to find the optimal result according to the evaluation criterion used, the sequential search gives up completeness and thus risks losing optimal subsets, and the random search starts with a randomly selected subset and proceeds in two different ways.

During the subset generation, each candidate subset is evaluated and compared with the previous best one according to a certain evaluation criterion. If the new subset turns out to be better, it replaces the previous best subset. The process of subset generation and evaluation is repeated until a given stopping criterion is satisfied.

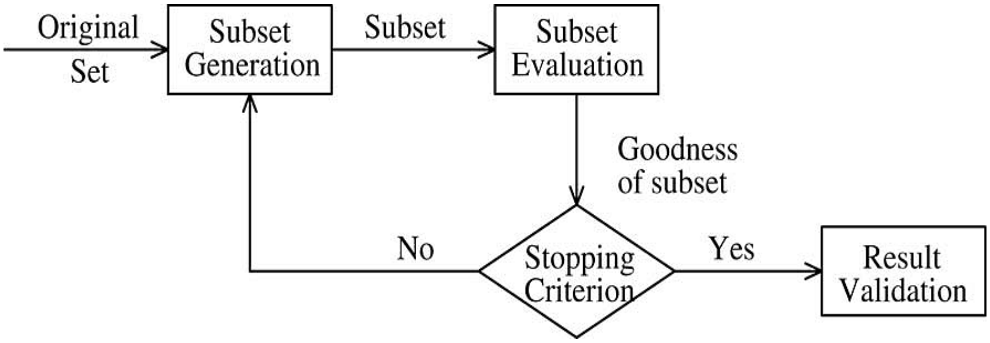


Figure 1. Feature Selection Process with Validation

Subset selection algorithms can be divided into wrapper [4, 13], filter [4, 13] and embedded methods [4, 5]. The filter model relies on general characteristics of the data to evaluate and select feature subsets without involving any mining algorithm. The wrapper model requires one predetermined mining algorithm and uses its performance as the evaluation criterion. It searches for features better suited to the mining algorithm aiming to improve mining performance, but it also tends to be more computationally expensive than the filter model [5, 10]. The embedded model attempts to take advantage of the two models by exploiting their different evaluation criteria in different search stages [5].

A broad overview of the different aspects of feature selection can be found in [1, 4, 13, 2, 9, 10].

3. Related Research

3.1.SVM

The Support Vector Machines are a set of supervised learning methods used for classification. They belong to a family of generalized linear classifiers. Their main aim is to simultaneously minimize the empirical classification error and maximize the geometric margin; hence they are also known as maximum margin classifiers [2, 14, 15, 16]. The SVM problem can be reduced to find the hyperplane that maximizes the distance from it to the nearest examples in each class.

Formally, let us consider the following training set of n examples and p attributes:

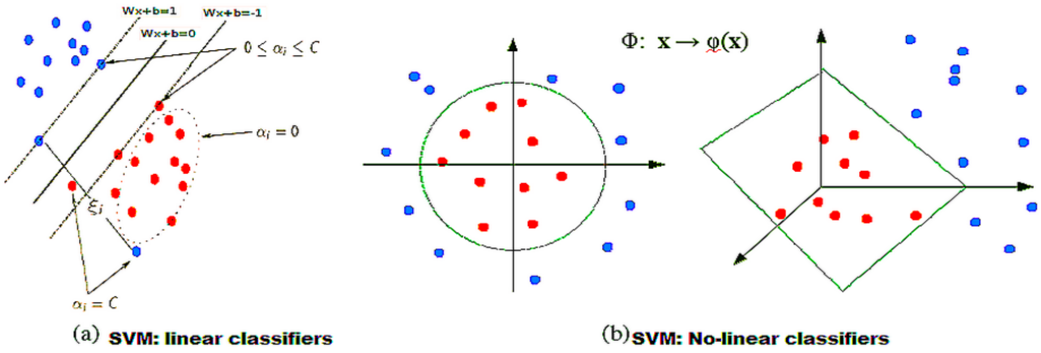
$$D = \{(x_i, c_i) \mid x_i \in R^p, c_i \in \{-1, +1\}, i = 1, \dots, n\},$$

where c_i is either +1 or -1 , indicating the class to which the p-dimensional real vector x_i belongs.

Any hyperplane which divides the points having $c_i = 1$, from those having $c_i = -1$ can be written as the set of points satisfying:

$$(3.1) \qquad w.x - b = 0$$

where w is normal to the hyperplane, $|b|/||w||$ is the perpendicular distance from the hyperplane to the origin, || w || is the Euclidean norm of w and b is a constant.



(a) **SVM: linear classifiers**

(b) **SVM: No-linear classifiers**

Figure 2. SVM-linear (Figure 2a) & SVM-nonlinear (Figure 2b)

Find the maximum-margin hyperplane can generally be reduced to determine the parameters w and b to minimize || w ||. This is a quadratic programming (QP) optimization problem which consists to minimize.

$$(3.2) \qquad \left\{ ||w||^2 + C \sum_i \varepsilon_i \; |y_i(w.\Phi(x_i) + b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0 \right\}$$

where C is a parameter to be chosen by the user (a larger C corresponding to assigning a higher penalty to errors), ϕ is a mapping function which transform the data to some other in order to find a separating hyperplane (Figure 2) and εi, i = 1, .., l are slack variables which verify the following constraints:

$$(3.3) \qquad xi.w + b \geq \; +1 - \; \varepsilon i \; for \; ci = \; +1$$

$$(3.4) \qquad x_i w + b \; \leq \; -1 - \; \varepsilon_i \; for \; c_i = \; -1$$

$$(3.5) \qquad \varepsilon_i \geq 0 \; \forall \; i$$

The equation 3.2 can be rewritten to a Lagrangian function and derive its dual form as:

$$3.6 \qquad \left\{\sum_{i} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\, \alpha_j y_i y_j K(x_i \cdot x_j) | 0 \le \alpha_i \le C, \sum \alpha_i . y_i = 0\right\} \square$$

where α_i are Lagrangian multipliers. The multipliers can be calculated by exploiting quadratic programming techniques or faster heuristic algorithms. After they are calculated, we can determine model parameters w and b by using the fact that $K(x_i, x_j)\ \phi(x_i).\phi(x_j)$ where K is a kernel function. With all the multipliers and the parameters w and b, a new example can be classified by investigating which side of the hyperplane it resides. More formally, this consists in using the following decision function:

$$3.7 \qquad f(x) = sign\left[\sum_{i} \alpha_i y_i K(x_i, x) + b]\right]$$

3.2. RFE-SVM

The well-studied RFE-SVM algorithm [2, 6, 3, 7] is a wrapper feature selection method which generates the ranking of features using backward feature elimination. It was originally proposed to perform gene selection for cancer classification [2]. Its basic idea is to eliminate redundant genes and yields better and more compact gene subsets. The features are eliminated according to a criterion related to their support to the discrimination function, and the SVM [15] is re-trained at each step. RFE-SVM is a weight-based method; at each step, the coefficients of the weight vector of a linear SVM are used as the feature ranking criterion. The RFE-SVM algorithm [2] can be broken into four steps:

1. Train an SVM on the training set;
2. Order features using the weights of the resulting classifier;
3. Eliminate features with the smallest weight;
4. Repeat the process with the training set restricted to the remaining features.

3.3. Local Search Techniques

In these last decades, local search has grown from a simple heuristic idea into a mature field of research in combinatorial optimization that is attracting ever-increasing attention [13, 14]. It remains the method of choice for NP-hard problems as it provides an efficient approach for obtaining better solutions. Its procedure consists to iteratively examine a set of solutions in the neighborhood of the current solution in order to find the best to replace it. According to Hernandez et al. [14], two components characterize the behavior of a local search procedure:

1. An evaluation function. If s is a solution candidate, its quality is assessed using an evaluation function f (e.g. SVM classifier, etc.). Two criteria are used in this evaluation. The first one is the ability of s to obtain a good classification with A, a subset of attributes and the maximum margin (M) given by the SVM classifier. Given two candidate solutions s and s_0, $f(s)$ is better than $f(s_0)$, denoted by $f(s) > f(s_0)$, if it has the best classification accuracy. If the classification accuracy is the same one, the maximum margins are used to compare them [14].

2. A neighborhood function. As studied in [13, 14], in local search algorithm, applying a move operator *mv* to a candidate solution *s* leads to a new solution s_0, denoted by $s_0 = s \bigoplus mv$. Let $\Gamma(s)$ be the set of all possible moves which can be applied to *s*, then the neighborhood $NH(s)$ of *s* is defined by $NH(s) = \{s \bigoplus mv \mid s \in \Gamma(s)\}$. In [14], the move is based on the drop/add operation which removes a gene g_i from the solution *s* and add another gene g_j.

4. Our Approach

4.1. Problem Definition

We can formalize our problem as follows: Let *X* be the training set, *R* the set of attributes removed from *X* by RFE process and *s* the subset returned by RFE-SVM algorithm. Due to the fact that *s* is not necessarily an optimal solution because RFE-SVM is greedy, our approach consists to find the best improving neighboring solution $s' \in N(s)$ such that $SVM(s_0) > SVM(s)$ and$\forall s'' \in N(s)$, $SVM(s_0) \geqslant SVM(s'')$. Notice that $N(s)$ represents the neighborhood of *s*. To achieve that, we specially used the Bit-Flip and the Attribute-Flip local search operators that we presented in the section 4.2. The basic idea of our approach is to give a second chance to the removed attributes.

4.2. Local Search Tools

Local search procedure is a search method that iteratively examines a set of solutions in the neighborhood of the current solution and replaces it with better neighbor if one exists.

In this paper, we devise effective LS procedures inspired from successfully search techniques adapted to the FS. The following paragraphs detail the neighborhood structures that will be deployed within the local search procedures. They will be, also, discussed in the context of FS search space exploration.

4.2.1. Bit-Flip (BF) Local Search explores neighboring solutions by adding or removing one feature at a time. For solutions encoded with binary strings this operator inverts one bit value for each neighbor. In comparison to the sequential search procedures, the generated neighborhood covers both solutions explored in SFS (see eq. 4.9) and SBE (see eq. 4.10) The Bit-Flip operator (NH_{BF}) neighborhood is illustrated by the equation 4.8.

$$4.1 \qquad NH_{BF}(S) = \{X \mid X = NH^{+}(S) \cup NH^{-}(S)\}$$

Where

$$4.2 \qquad NH^{+}(S) = \{X \mid X = S \cup \{f_i\}, \forall f_i \in F, f_i \notin S\}$$

$$4.3 \qquad NH^{-}(S) = \{X \mid X = S - \{f_i\}, \forall\, f_i \in S\}$$

The problem of nesting effect encountered with both sequential forward and backward procedures is alleviated by the merge of the neighborhoods explored by both procedures.

4.2.2 Attribute-Flip (AF) Local Search constructs neighborhood using permutation between selected and non-selected features (see eq. 4.11). All combinations are considered. Two properties characterize the resulting neighborhood: (i) the hamming distance is equal to 2; (ii) the operator preserves the feature subset size.

$$NB_{AF} = \{\mathrm{X} | \mathrm{X} = \mathrm{S} \cup \{\mathrm{f_i}\} - \{\mathrm{f_j}\}, \forall\, \mathrm{f_i} \in \mathrm{X}, \mathrm{f_j} \notin \mathrm{X}\} \quad (4.4)$$

The two operators explore different regions of the current solution neighborhood. There is no overlapping region ($NB_{BF}(S) \cap NB_{AF}(S) = \emptyset$) and the second neighborhood structure is much larger than the first which would require more computational time.

4.3. Our Algorithm

We propose a feature selection approach based on RFE-SVM algorithm. Our algorithm can be broken into two steps (see Figure 3): (1) The initialization step which gives an initial solution obtained by applying RFE-SVM algorithm [2]. (2) In the second step, we introduce local search procedure [13] in order to improve the quality of the initial solution. This is justified by the fact that the RFE-SVM algorithm does not necessarily return an optimal solution due to its greedy nature.

The Algorithm 1 represents the local search (LS) procedure that we use.

5. Empirical Study

The experiment was conducted with dual core 2.20 GHz with 4.00 Go of memory on windows platform, and we implemented the algorithm using Weka [17] which is a widely used data mining toolkit. To demonstrate real practicality of our approach, we ran experiments on the following datasets from UCI repository [8]: Ionosphere, the SpamBase and the SPECTF Heart datasets:

- Ionosphere which contains 351 instances and 34 attributes. We use 50% of records as the training data and the other 50% as the testing data;
- SpamBase which initially contains 4601 instances and 57 attributes. We use 512 records as training data and the remainder of records as the test data;
- SPECTF Heart which is divided into training data (80 instances) and testing data (187 instances).

For each dataset, we use the training data to build the classifier, and then use the testing data to measure how accurate this classifier can predict the class labels.

The percentage of the correct predictions is the accuracy value. The Table 1 highlights the differences between our algorithm and RFE-SVM under two metrics: the accuracy and the number of attributes of the selected subset. In Table 1 we observe the contribution of the local search on the RFE-SVM process.

Table 1. Comparison between RFE-SVM and our Approach

Datasets	*RFE –SVM*	*RFE −SVM +BF*	*RFE −SVM +AT*
Ionosphere	N = 5, P = 85.714%	N = 6, P = 85.714%	N = 5, P = 86.286%
SpamBase	N = 39, P=89.606%	N = 38, P=89.729%	N = 39, P=89.704%
Spect Heart Data	N = 5, P = 71.123%	N = 6, P = 72.727%	N = 5, P = 73.262%

The Table 1 shows clearly that the reuse of features previously removed during the RFE-SVM process improves the quality of the final classifier.

As you can see from the three used datasets, our algorithm returns a prediction rate which is better than that returned by the basic RFE-SVM algorithm. For example, on the *Ionosphere* dataset, we note that our algorithm returns a subset containing only 5 attributes with a prediction rate of 86.29%, while RFE-SVM algorithm returns a subset of 5 attributes with a prediction rate of 85.71%. On the dataset *SpamBase*, we note that our algorithm provides better results, because both the size of the returned subset and the classification accuracy are better than those obtained with the basic RFE-SVM algorithm.

Finally, the results of the third dataset used in our experiment highlight that our feature selection algorithm for supervised classification has a prediction rate higher than that obtained with the basic RFE-SVM algorithm.

```
Algorithm 1 Iterative Local Search procedure
Input:
    S: Solution returned by RFE-SVM
    S': Feature set removed by RFE-SVM
    Cla: the classifier
Output:
    $S_{local}$ : result of local search
Begin
    $S_1 \leftarrow S, S_{best} \leftarrow S_1$
    $Stop \leftarrow false$
Repeat
    $Sol_{list} \leftarrow NH(S_1, S')$
    $\forall Sol \in Sol_{list}, Evaluate(Sol,Cla)$
    $S1 \leftarrow getBest(Sol_{list})$
    If $S_1.fitness > S_{best}.fitness$ then
    $S_{best} \leftarrow S_1$
    Else
    $Stop \leftarrow true$
    End if
Until ($Stop \leftarrow true$)
    $S_{local} \leftarrow S_{best}$
Return ($S_{local}$)
End
```

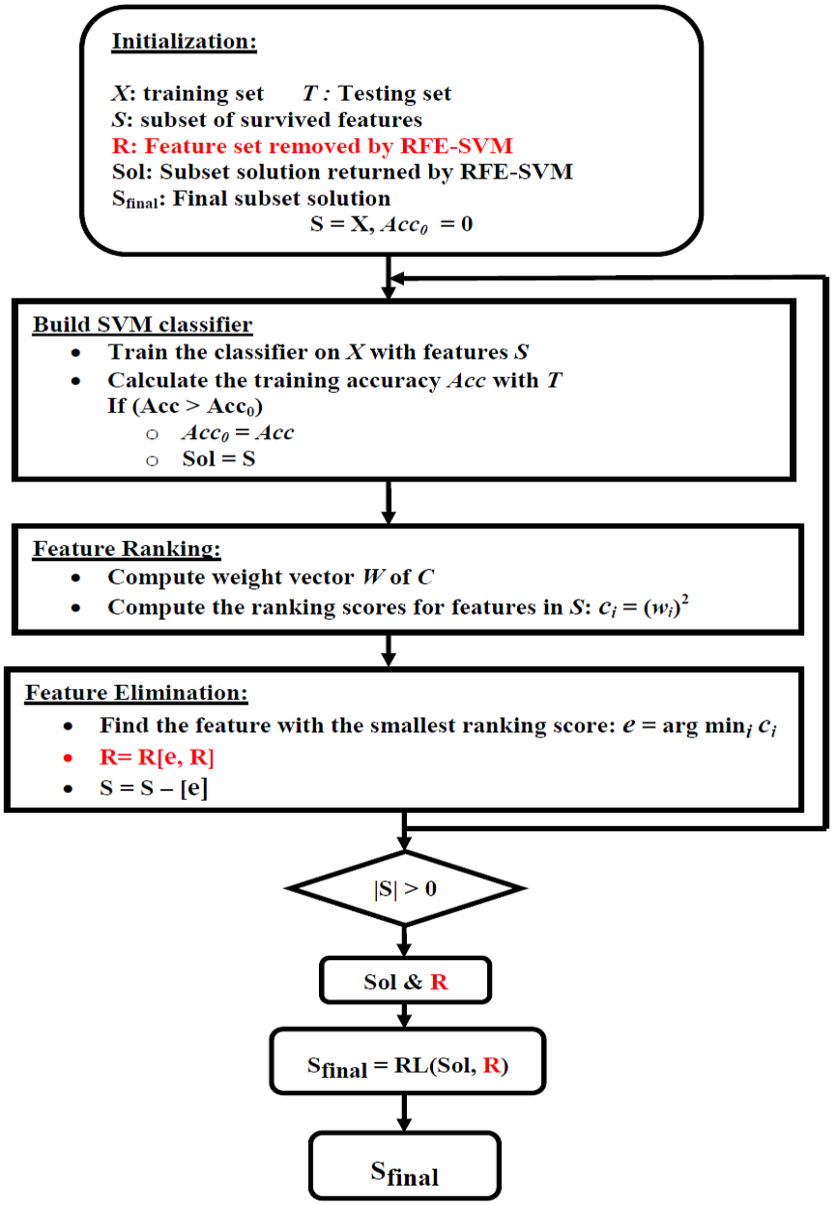

Initialization:
X: training set T : Testing set
S: subset of survived features
R: Feature set removed by RFE-SVM
Sol: Subset solution returned by RFE-SVM
Sfinal: Final subset solution
S = X, Acc0 = 0
Build SVM classifier
Train the classifier on X with features S
Calculate the training accuracy Acc with T
If (Acc > Acc0)
Acc0 = Acc
Sol = S
Feature Ranking:
Compute weight vector W of C
Compute the ranking scores for features in S: ci = (wi)2
Feature Elimination:
Find the feature with the smallest ranking score: e = arg mini ci
R= R[e, R]
S = S – [e]
|S| > 0
Sol & R
Sfinal = RL(Sol, R)
Sfinal

Figure 3. RFE-SVM Algorithm based Local Search Procedure

6. Conclusion and Future Works

In this work, we proposed a modified RFE-SVM-based feature selection method for classification problem. It aims to combine the RFE-SVM algorithm with local search operators in order to improve the quality of the final classifier. From the empirical results, we found that the reuse of features previously removed during the RFE-SVM process can improve the RFE-SVM classifier. In the future, we plan to run experiments on datasets with very large number of attributes.

References

[1] R. S. Huan Liu, H. Motoda and Z. Zhao, "Feature selection: An ever evolving frontier in data mining," JMLR: Workshop and Conference Proceedings 10: The Fourth Workshop on Feature Selection in Data Mining, **(2010)** pp. 4-13.

[2] Guyon, Weston, Barnhill, and Vapnik, "Gene selection for cancer classification using support vector machines," MACHLEARN: Machine Learning, vol. 46, **(2002)**.

[3] Pember A. Mundra and J. C. Rajapakse, "SVM-RFE with relevancy and redundancy criteria for gene selection," in PRIB, J. C. Rajapakse, B. Schmidt, and L. G. Volkert, Eds., vol. 4774, Springer, **(2007)**, pp. 242–252.

[4] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering", IEEE Trans. Knowl. Data Eng, vol. 17, no. 4, **(2005)**, pp. 491–502.

[5] T. N. Lal, O. Chapelle, J. Weston and A. Elisseeff, "Embedded Methods", Springer, **(2004)** November 20.

[6] Y. Tang, Y.-Q. Zhang and Z. Huang, "Development of two-stage SVM-RFE gene selection strategy for microarray expression data analysis", IEEE/ACM Trans. Comput. Biology Bioinform, vol. 4, no. 3, **(2007)**, pp. 365–381.

[7] K. Duan and J. C. Rajapakse, "Svm-rfe peak selection for cancer classification with mass spectrometry data," in APBC, **(2005)**, pp. 191–200.

[8] "Uci machine learning repository", http://archive.ics.uci.edu/ml/datasets.

[9] A. K. Jain and D. E. Zongker, "Feature selection: Evaluation, application, and small sample performance," IEEE Trans. Pattern Anal. Mach. Intell, vol. 19, no. 2, **(1997)**, pp. 153–158.

[10] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," J. Mach. Learn. Res., vol. 3, **(2003)** March, pp. 1157–1182.

[11]M. Dash and H. Liu, "Feature selection for classification," Intell. Data Anal, vol. 1, no. 1-4, **(1997)**, pp. 131–156.

[12]S. Davies and S. Russell, "NP-completeness of searches for smallest possible feature sets," **(1994)** October 18.

[13] M. A. Esseghir, G. Goncalves and Y. Slimani, "Memetic feature selection: Benchmarking hybridization schemata", in HAIS(1), **(2010)**, pp. 351–358.

[14] J. C. H. Hernandez, B. Duval and J.-K. Hao, "Svmbased local search for gene selection and classification of microarray data," in BIRD, **(2008)**, pp. 499–508.

[15]V. N. Vapnik, "The nature of statistical learning theory", New York, NY, USA: Springer-Verlag New York, Inc., **(1995)**.

[16]C. J. C. Burges, "A tutorial on support vector machines for pattern recognition", Data Min. Knowl. Discov., vol. 2, **(1998)** June, pp. 121–167, (http://portal.acm.org/citation.cfm?id=593419.593463).

[17]I. H. Witten and E. Frank, "Data mining: practical machine learning tools and techniques with Java implementations", San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., **(2000)**.