Computers and Electrical Engineering 40 (2014) 16–28

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

A survey on feature selection methods ☆

Girish Chandrashekar *, Ferat Sahin

Electrical and Microelectronic Engineering, Rochester Institute of Technology, Rochester, NY 14623, USA

ARTICLE INFO

Article history:
Available online 7 December 2013

ABSTRACT

Plenty of feature selection methods are available in literature due to the availability of data with hundreds of variables leading to data with very high dimension. Feature selection methods provides us a way of reducing computation time, improving prediction performance, and a better understanding of the data in machine learning or pattern recognition applications. In this paper we provide an overview of some of the methods present in literature. The objective is to provide a generic introduction to variable elimination which can be applied to a wide array of machine learning problems. We focus on Filter, Wrapper and Embedded methods. We also apply some of the feature selection techniques on standard datasets to demonstrate the applicability of feature selection techniques.

1. Introduction

A feature is an individual measurable property of the process being observed. Using a set of features, any machine learning algorithm can perform classification. In the past years in the applications of machine learning or pattern recognition, the domain of features have expanded from tens to hundreds of variables or features used in those applications. Several techniques are developed to address the problem of reducing irrelevant and redundant variables which are a burden on challenging tasks. Feature Selection (variable elimination) helps in understanding data, reducing computation requirement, reducing the effect of curse of dimensionality and improving the predictor performance. In this paper we look at some of the methods found in literature which use particular measurements to find a subset of variables (features) which improves the overall prediction performance.

The focus of feature selection is to select a subset of variables from the input which can efficiently describe the input data while reducing effects from noise or irrelevant variables and still provide good prediction results [1]. One of the applications would be in gene microarray analysis [1–5]. The standardized gene expression data can contain hundreds of variables of which many of them could be highly correlated with other variables (e.g. when two features are perfectly correlated, only one feature is sufficient to describe the data). The dependant variables provide no extra information about the classes and thus serve as noise for the predictor. This means that the total information content can be obtained from fewer unique features which contain maximum discrimination information about the classes. Hence by eliminating the dependent variables, the amount of data can be reduced which can lead to improvement in the classification performance. In some applications, variables which have no correlation to the classes serve as pure noise might introduce bias in the predictor and reduce the classification performance. This can happen when there is a lack of information about the process being studied. By applying feature selection techniques we can gain some insight into the process and can improve the computation requirement and prediction accuracy.

To remove an irrelevant feature, a feature selection criterion is required which can measure the relevance of each feature with the output class/labels. From a machine learning point if a system uses irrelevant variables, it will use this information

☆ Reviews processed and approved for publication by Editor-in-Chief Dr. Manu Malek.
* Corresponding author. Tel.: +1 5857540555.
E-mail addresses: gxc6334@rit.edu (G. Chandrashekar), feseee@rit.edu (F. Sahin).

0045-7906/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compeleceng.2013.11.024

for new data leading to poor generalization. Removing irrelevant variables must not be compared with other dimension reduction methods such as Principal Component Analysis (PCA) [6] since good features can be independent of the rest of the data [7]. Feature elimination does not create new features since it uses the input features itself to reduce their number. Once a feature selection criterion is selected, a procedure must be developed which will find the subset of useful features. Directly evaluating all the subsets of features (2^N) for a given data becomes an NP-hard problem as the number of features grow. Hence a suboptimal procedure must be used which can remove redundant data with tractable computations.

In this paper we will look at some of these methods developed for this purpose. In [8], the variable elimination methods were broadly classified into filter and wrapper methods. Filter methods act as preprocessing to rank the features wherein the highly ranked features are selected and applied to a predictor. In wrapper methods the feature selection criterion is the performance of the predictor i.e. the predictor is wrapped on a search algorithm which will find a subset which gives the highest predictor performance. Embedded methods [1,9,10] include variable selection as part of the training process without splitting the data into training and testing sets. In this paper we will focus on feature selection methods using supervised learning algorithms and a very brief introduction to feature selection methods using unsupervised learning will be presented.

The rest of the paper is organized as follows. In Section 2 Filter methods are presented followed by Wrapper methods in Section 3. Section 4 provides a brief overview of the Embedded methods. In Section 5 we present briefly other feature selection techniques for unsupervised and semi-supervised learning and in Section 6 we present a brief discussion on the stability of the feature selection algorithms followed by Section 7 where we look at two classifiers which can be used for feature selection. In Section 8 we present some of the results obtained by applying the feature selection algorithms and finally in Section 9 we present the conclusion.

2. Filter methods

Filter methods use variable ranking techniques as the principle criteria for variable selection by ordering. Ranking methods are used due to their simplicity and good success is reported for practical applications. A suitable ranking criterion is used to score the variables and a threshold is used to remove variables below the threshold. Ranking methods are filter methods since they are applied before classification to filter out the less relevant variables. A basic property of a unique feature is to contain useful information about the different classes in the data. This property can be defined as feature relevance [8] which provides a measurement of the feature's usefulness in discriminating the different classes.

Here the issue of relevancy of a feature has to be raised i.e. how do we measure the relevancy of a feature to the data or the output. Several publications [1,8–11] have presented various definitions and measurements for the relevance of a variable. One definition that can be mentioned which will be useful for the following discussion is that "A feature can be regarded as irrelevant if it is conditionally independent of the class labels." [7]. It essentially states that if a feature is to be relevant it can be independent of the input data but cannot be independent of the class labels i.e. the feature that has no influence on the class labels can be discarded. As mentioned above inter feature correlation plays an important role in determining unique features. For practical applications the underlying distribution is unknown and is measured by the classifier accuracy. Due to this, an optimal feature subset may not be unique because it may be possible to achieve the same classifier accuracy using different sets of features.

Next we will look into two ranking methods which will help us understand the relevance of a feature. For the rest of the paper we use a standard notation to represent data and the variables. The input data $[x_{ij}, y_k]$ consists of N samples $i = 1$ to N with D variables $j = 1$ to D, x_i is the ith sample and y_k is the class label $k = 1$ to Y.

2.1. Correlation criteria

One of the simplest criteria is the Pearson correlation coefficient [1,12] defined as:

$$R(i) = \frac{cov(x_i, Y)}{\sqrt{var(x_i) * var(Y)}} \tag{1}$$

where x_i is the i_{th} variable, Y is the output (class labels), $cov()$ is the covariance and $var()$ the variance. Correlation ranking can only detect linear dependencies between variable and target.

2.2. Mutual Information (MI)

Information theoretic ranking criteria [1,5,8,12–14] use the measure of dependency between two variables. To describe MI we must start with Shannons definition for entropy given by:

$$H(Y) = -\sum_y p(y) log(p(y)) \tag{2}$$

Eq. (2) represents the uncertainty (information content) in output Y. Suppose we observe a variable X then the conditional entropy is given by:

$$H(Y|X) = -\sum_{x}\sum_{y} p(x,y)log(p(y|x)) \tag{3}$$

Eq. (3) implies that by observing a variable X, the uncertainty in the output Y is reduced. The decrease in uncertainty is given as:

$$I(Y,X) = H(Y) - H(Y|X) \tag{4}$$

This gives the MI between Y and X meaning that if X and Y are independent then MI will be zero and greater than zero if they are dependent. This implies that one variable can provide information about the other thus proving dependency. The definitions provided above are given for discrete variables and the same can be obtained for continuous variables by replacing the summations with integrations. The MI can also be defined as a distance measure given by:

$$K(f,g) = \int f(y)log\left(\frac{f(y)}{g(y)}\right) \tag{5}$$

The measure K in (5) is the Kullback–Leibler divergence [15,16] between two densities which can also be used as a measure of MI. From the above equations, we need to know the probability density function (PDF) of the variables to calculate MI. Since the data we obtain is of finite samples, the PDF cannot be calculated accurately. Several methods have been developed for estimating the MI in [12,15,16]. Once a particular method is chosen for calculating MI then one of the simplest methods for feature selection is to find the MI between each feature and the output class labels and rank them based on this value. A threshold is set to select $d < D$ features. This is a simple method and the results can be poor [13] since inter-feature MI is not taken into account. But MI is an important concept and is used in embedded methods which will be presented in a later section.

In [17] the author develops a feature ranking criteria based on Conditional Mutual Information for binary data (boolean data). A score table is updated as features are selected to the subset using the conditional mutual information criteria which is to be maximized. The score at each iteration is calculated using (6) given by:

$$s[n] = min_{l<k}\widehat{I}(Y;X_n|X_{v(l)}) \tag{6}$$

where $s[n]$ is the score which is updated at each iteration, X_n is the current evaluated feature, $X_{v(l)}$ is the set of already selected features. The above equation iteratively selects features which maximize MI with the class and not select features similar to ones already picked. This provides a good trade-off between independence and discrimination.

Since the data distribution is not known, various techniques can be used to evaluate different subsets with a chosen classifier. Other statistical tests found in literature can be used for feature ranking. In [13] twelve feature selection metrics are considered for the text classification problem [1,13,18]. All the features are ranked using each metric and a threshold is set which would select 100 words which are then applied to the predictor. Filter approaches applied to various applications can be found in [19–21,18,5]. Earlier comparisons for text classification using ranking methods can be found in [22]. In [23,24] the authors develop a ranking criteria based on class densities for binary data. A two stage algorithm utilizing a less expensive filter method to rank the features and an expensive wrapper method to further eliminate irrelevant variables is used. The RELIEF algorithm [25,26] is another filter based approach wherein a feature relevance criterion is used to rank the features. Using a threshold a subset of features is selected. The drawback of the RELIEF algorithm is in selecting a threshold. Authors in [26] compare the RELIEF and other wrapper methods for different datasets. In [19] discarded variables are used to perform multitask learning (MTL). In [27] a random variable called probe is used to rank the features using Gram-Schmidt orthogonalization.

The advantages of feature ranking are that it is computationally light and avoids overfitting and is proven to work well for certain datasets [1,28,5]. Filter methods do not rely on learning algorithms which are biased which is equivalent to changing data to fit the learning algorithm. One of the drawbacks of ranking methods is that the selected subset might not be optimal in that a redundant subset might be obtained. Some ranking methods such as Pearson correlation criteria (1) and MI (4) do not discriminate the variables in terms of the correlation to other variables. The variables in the subset can be highly correlated in that a smaller subset would suffice [11,28]. This issue of redundant vs. relevant variables is addressed in [1] with good examples. In feature ranking, important features that are less informative on their own but are informative when combined with others could be discarded [1,29]. Finding a suitable learning algorithm can also become hard since the underlying learning algorithm is ignored [11]. Also, there is no ideal method for choosing the dimension of the feature space.

In the next section, we will look at the second type of feature selection method called Wrapper methods. Unlike filter methods which use a feature relevance criteria, Wrapper methods rely on the classification for obtaining a feature subset.

3. Wrapper methods

Wrapper methods use the predictor as a black box and the predictor performance as the objective function to evaluate the variable subset. Since evaluating 2^N subsets becomes a NP-hard problem, suboptimal subsets are found by employing search algorithms which find a subset heuristically. A number of search algorithms can be used to find a subset of variables which maximizes the objective function which is the classification performance. The Branch and Bound method [8,30] used tree structure to evaluate different subsets for the given feature selection number. But the search would grow exponentially

[8] for higher number of features. Exhaustive search methods can become computationally intensive for larger datasets. Therefore simplified algorithms such as sequential search or evolutionary algorithms such as Genetic Algorithm (GA) [31] or Particle Swarm Optimization (PSO) [32] which yield local optimum results are employed which can produce good results and are computationally feasible.

We broadly classify the Wrapper methods into Sequential Selection Algorithms and Heuristic Search Algorithms. The sequential selection algorithms start with an empty set (full set) and add features (remove features) until the maximum objective function is obtained. To speed up the selection, a criteria is chosen which incrementally increases the objective function until the maximum is reached with the minimum number of features. The heuristic search algorithms evaluate different subsets to optimize the objective function. Different subsets are generated either by searching around in a search-space or by generating solutions to the optimization problem. First we will look at sequential selection algorithms followed by the heuristic search algorithms.

3.1. Sequential selection algorithms

These algorithms are called sequential due to the iterative nature of the algorithms. The Sequential Feature Selection (SFS) algorithm [33,34] starts with an empty set and adds one feature for the first step which gives the highest value for the objective function. From the second step onwards the remaining features are added individually to the current subset and the new subset is evaluated. The individual feature is permanently included in the subset if it gives the maximum classification accuracy. The process is repeated until the required number of features are added. This is a naive SFS algorithm since the dependency between the features is not accounted for. A Sequential Backward Selection (SBS) algorithm can also be constructed which is similar to SFS but the algorithm starts from the complete set of variables and removes one feature at a time whose removal gives the lowest decrease in predictor performance.

The Sequential Floating Forward Selection (SFFS) [33,34] algorithm is more flexible than the naive SFS because it introduces an additional backtracking step. The basic flowchart is given in Fig. 1 where k is the current subset size and d is the required dimension. The first step of the algorithm is the same as the SFS algorithm which adds one feature at a time based on the objective function. The SFFS algorithm adds another step which excludes one feature at a time from the subset obtained in the first step and evaluates the new subsets. If excluding a feature increases the value of the objective function then that feature is removed and goes back to the first step with the new reduced subset or else the algorithm is repeated from the top. This process is repeated until the required number of features are added or required performance is reached.

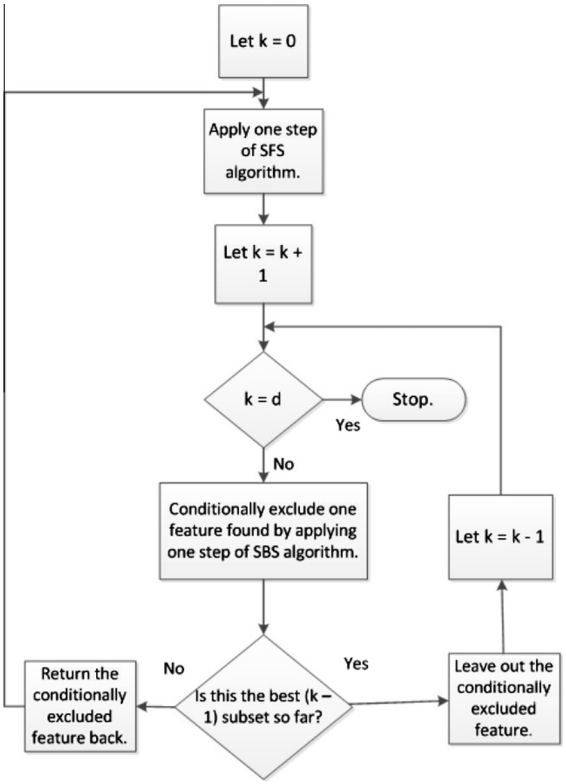

Fig. 1. SFFS flow chart.

The SFS and SFFS methods suffer from producing nested subsets since the forward inclusion was always unconditional which means that two highly correlated variables might be included if it gave the highest performance in the SFS evaluation. To avoid the nesting effect, adaptive version of the SFFS was developed in [35,36]. The Adaptive Sequential Forward Floating Selection (ASFFS) algorithm used a parameter r which would specify the number of features to be added in the inclusion phase which was calculated adaptively. The parameter o would be used in the exclusion phase to remove maximum number of features if it increased the performance. The ASFFS attempted to obtain a less redundant subset than the SFFS algorithm. It can be noted that a statistical distance measure can also be used as the objective function for the search algorithms as done in [9,10,33,35]. Theoretically, the ASFFS should produce a better subset than SFFS but this is dependent on the objective function and the distribution of the data. The Plus-L-Minus-r search method [35,37,38] also tries to avoid nesting. In the Plus-L-Minus-r search, in each cycle L variables were added and r variables were removed until the desired subset was achieved. The parameters L and r have to be chosen arbitrarily. In [37] the authors try to improve the SFFS algorithm by adding an extra step after the backtracking step in the normal SFFS in which a weak feature is replaced with a new better feature to form the current subset.

3.2. Heuristic search algorithms

Genetic Algorithm (GA) [31] can be used to find the subset of features [36,39–42] wherein the chromosome bits represent if the feature is included or not. The global maximum for the objective function can be found which gives the best suboptimal subset. Here again the objective function is the predictor performance.

The GA parameters and operators can be modified within the general idea of an evolutionary algorithm to suit the data or the application to obtain the best performance or the best search result. A modified version of the GA called the CHCGA [43,44] can be used for feature selection [36]. The CHCGA is a non-traditional GA which differs from GA in the following ways:

- The best N individuals are chosen from the pool of parents and offspring i.e. better offspring replaces lesser fit parents.
- A highly disruptive half uniform crossover (HUX) operator is used which crosses over exactly half of the non-matching alleles, wherein the bits to be crossed over are selected at random.
- During reproduction step, each member of the parent population is randomly selected without replacement and paired for mating. Not all the pairs are crossed over but before mating the Hamming distance between the parents are calculated and if half this distance does not exceed a threshold d, they are not mated. The threshold is usually initialized to $L/4$ where L is the chromosome length. If no offspring is obtained in the generation, the threshold is decremented by one. Due to these mating criteria of mating only diverse parents, the population converges as the threshold decreases.
- If there are no offspring generated and the threshold drops to zero, a cataclysmic mutation is introduced to create a new population. The best individual in the current parent population is taken as the template to create the new population. The rest $N-1$ individuals are obtained by randomly flipping a percentage (35–40%) of bits of the template. The regular mutation after crossover step is skipped each time and the above mentioned mutation is carried if required.

The CHCGA converges on the solution faster and provides a more effective search by maintaining the diversity and avoiding stagnation of the population. In [45] multi-objective GA is used for hand written digit recognition. In [46,47] several wrapper methods some of which mentioned above, are compared with different datasets. They derive various fitness functions with weighting/penalty imposing characteristics. A binary PSO [32,48,4] algorithm can also be used for wrapper implementation. In [49] comparison between GA and PSO using SVM for gene selection can be found.

The main drawback of Wrapper methods is the number of computations required to obtain the feature subset. For each subset evaluation, the predictor creates a new model i.e. the predictor is trained for each subset and tested to obtain the classifier accuracy. If the number of samples is large, most of the algorithm execution is spent in training the predictor. In some algorithms such as GA feature selection, the same feature subset might be evaluated multiple times since the classifier accuracies for evaluated subsets are not stored for future retrieval. Another drawback of using the classifier performance as the objective function is that the classifiers are prone to overfitting [8]. Overfitting occurs if the classifier model learns the data too well and provides poor generalization capability. The classifier can introduce bias and increases the classification error. Using classification accuracy in subset selection can result in a bad feature subset with high accuracy but poor generalization power. To avoid this, a separate holdout test set can be used to guide the prediction accuracy of the search [8].

In the next section, we will discuss Embedded methods which try to compensate for the drawbacks in the Filter and Wrapper methods.

4. Embedded methods

Embedded methods [1,9,10] want to reduce the computation time taken up for reclassifying different subsets which is done in wrapper methods. The main approach is to incorporate the feature selection as part of the training process. In Section 2 we mentioned that MI is an important concept but the ranking using MI yielded poor results since the MI between the feature and the class output only was considered. In [12] a greedy search algorithm is used to evaluate the subsets. The

objective function is designed such that choosing a feature will maximize the MI between the feature and the class output while the MI between the selected feature and the subset of the so far selected features is a minimum. This is formulated as:

$$I(Y,f) - \beta \sum_{s \in S} I(f;s) \tag{7}$$

where Y is the output, f is the current selected feature, s is the feature in the already selected subset S and β controls the importance of the MI between the current feature f and the features in the subset S. The output subset is applied to a Neural Network classifier. Eq. (7) will select better subset since the inter-feature MI is used in the calculation to select the non-redundant features. An improvement of this method is presented in [14] which estimates the MI using Parzen window method.

The mRMR (max-relevancy, min-redundancy) [24] is another method based on MI. It uses similar criteria as in (7) given as:

$$I(x_j;C) - \frac{1}{m-1} \sum_{x_l \in S_{m-1}} I(x_j;x_l) \tag{8}$$

where x_i is the mth feature in subset S and the set S_{m-1} is the so far selected subset with $m-1$ features. Instead of a greedy algorithm a two stage approach is implemented. First the criterion (8) is used to select a number k which is the optimal number of features which gives the lowest cross-validation classification error. In the second stage, wrapper methods are used to evaluate different subsets of size k or direct evaluations are done on different subsets to find the subset which consistently yields the smallest classification error. The application of mRMR can be found in [50,3] wherein the simplest incremental search method is used with four different classifiers for gene classification.

Another method used in literature is to use the weights of a classifier [1,2,50] to rank the feature for their removal. Let w_j be defined as:

$$w_j = \frac{\mu_j(+) - \mu_j(-)}{\sigma_j(+) + \sigma_j(-)} \tag{9}$$

where $\mu_j(+)$ and $\mu_j(-)$ are the mean of the samples in class (+) and (−) and σ_j is the variance of the respective classes and j = *1 to D*. Eq. (9) [2] can be used as a ranking criteria to sort the features. The rank vector w can be used to classify since features rank proportionally contributes to the correlation. A voting scheme given as:

$$D(x) = w(x - \mu) \tag{10}$$

where w is the rank of the features or weight, $D(x)$ is the decision and μ is the mean of the data. Hence the weights (rank) of the features can be used as classifier weights. By conducting sensitivity analysis [1,2] of the weights, feature selection can be done i.e. the change in the weight w_j can be viewed as removing a feature j. In [2] it is suggested to use the change in the objective function, a linear discriminant function J which is a function of w_j. This concept of using the weights as the ranking and the search is done using the change in the objective function is applied to the SVM classifier [2,51,50] to perform Recursive Feature Elimination or known as the SVM-RFE method. The objective function of the SVM is presented later in Section 7. The SVM-RFE method in [2] is proposed for binary class classification, its multi-class classification technique can be found in [52]. In [53] the authors use the concept of RFE to derive a modified algorithm for selecting features in hyper spectral image data. In the SVM-RFE method, the l_2 norm is used in the SVM minimization problem. It is shown in the literature that other norms can be used which help in feature selection. In [54] the authors provide a comprehensive review of the different SVM based feature selection methods. The paper also presents a non-linear classification and feature selection approaches using SVM.

Similar to optimizing the SVM equation and assigning weight to features, the same can be done using Neural Networks. Multilayer perceptron networks are trained and feature weights are calculated using a saliency measure calculated from the trained network [55,56]. In [55] a penalty is applied for features with small magnitude at the node and the nodes connecting to these input features are excluded. This type of node removal is also called Network Pruning [55,56] is commonly used to obtain the optimum network architecture for Neural Networks. In [57] the authors derive a cost function for random variable elimination. In chemometric applications [58–60] regression models are developed to reduce the number of variables which helps in analyzing the chemical properties. In [19] the discarded variables are used to improve the prediction performance using multitask learning (MTL) approach. Lazy Feature Selection (LFS) approach is developed in [61] where the authors take advantage of the sparseness in the feature space as a feature selection method for text categorization problem. The LFS ranks the test samples against all training samples and k-NN [18,6] type algorithm is used to determine the class.

So far we have discussed the feature selection techniques using supervised learning i.e. the output class labels of the data were known or could be calculated. In practical applications, there can be processes whose operational details are unknown but their operational data is available. In the next section we will briefly look at some of the articles which apply feature selection using unsupervised learning.

5. Other feature selection techniques

Unsupervised learning deals with finding hidden structure in unlabelled data. Clustering techniques [7] are a primary example of unsupervised learning which tries to discover natural groupings in a set of objects without knowledge of class labels. Feature selection using unsupervised learning techniques are beyond the scope of this paper and will not be discussed in detail but in this section we mention a few articles which perform unsupervised feature selection. Feature selection using unsupervised learning can provide better description and reliability of the data than just unsupervised learning [7]. Several papers that attempt to solve the feature selection using unsupervised learning can be found in [7,62–65].

Semi-supervised learning is another class wherein both labelled and unlabelled data are used for learning [7,29,66,67]. It uses both labelled data (less number of samples) and unlabelled data (abundantly available) to modify a hypothesis obtained from labelled data alone. In [67] the authors use a clustering indicator construction to score a set of features. In [29] the authors use the maximum margin principle (SVM) using manifold regularization problem optimization similar to SVM-RFE [2].

Ensemble feature selection [68,69] is a relatively new technique used to obtain a stable feature subset. A single feature selection algorithm is run on different subsets of data samples obtained from bootstrapping method. The results are aggregated to obtain a final feature set. Authors in [68,69] use filter methods to rank the genes/features and use different aggregation methods such as ensemble-mean, linear aggregation, weighted aggregation methods to obtain the final feature subset.

6. Stability of feature selection algorithms

For a particular application, various feature selection algorithms can be applied and the best one can be selected which meets the required criteria. An overlooked problem is the stability of the feature selection algorithms. Stability of a feature selection algorithm can be viewed as the consistency of an algorithm to produce a consistent feature subset when new training samples are added or when some training samples are removed [68,70–73]. If the algorithm produces a different subset for any perturbations in the training data, then that algorithm becomes unreliable for feature selection. Examples of instabilities are demonstrated in [70,73] which can be verified by changing the training set and running the algorithm again. In [70], wrapper techniques are used to study their instability and stability measures are introduced along with possible solutions to alleviate the problem. Various measures are established in [68,70–72] to evaluate different subsets obtained for a certain number of runs. Using these measures, a more robust subset can be found for different datasets. In [73] multicriterion fusion algorithm is developed which uses multiple feature selection algorithms to rank/score the features which are combined to obtain a robust subset based on combining multiple classifiers to improve the accuracy. In [74] the author also suggests dividing the input features (based on their feature extraction procedures) to obtain different classifiers and combine the predictions to obtain a final decision.

7. Classifiers

In this section we want to provide a brief introduction to two classifiers in literature which can be used for feature selection tasks. We will present the SVM and RBF classifiers due to their wide range of applications found in literature.

7.1. Support Vector Machine (SVM)

SVM [51,2,24,18] is a marginal classifier which maximizes the margin between the data samples in the two classes. An optimal hyperplane boundary is drawn which will separate the data. In SVM, kernels are used to map the input data to a higher dimensional space where a decision boundary can be constructed. The decision function is given as:

$$D(x) = w\phi(x) + b \tag{11}$$

where w and b are the SVM parameters and $\phi(x)$ is a kernel function that maps the input data into the new M dimension. Eq. (11) defines the hyperplane and

$$\frac{D(x)}{\|w\|} \tag{12}$$

is the distance between hyperplane and pattern x. In Fig. 2 (taken from [51]) a graphical illustration of the boundary and the samples is provided. The objective of the training algorithm is to find w such that $M*$ is maximized. For the linear decision function in (11), the parameters are given as:

$$w = \sum_k a_k y_k x_k \tag{13}$$

$$b = (y_k - w * x_k) \tag{14}$$

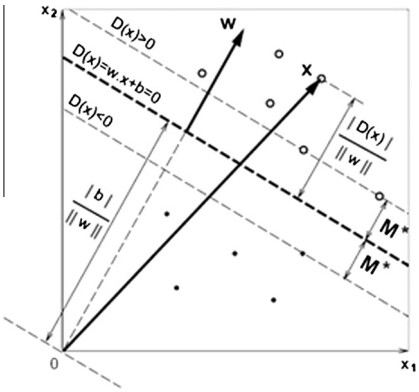

Fig. 2. Maximum margin for linear decision function $D(x)$ [10].

The vector w is a linear combination of training patterns wherein α_k is non-zero for the marginal patterns which forms the support vectors. The value b is an average over the support vectors. The objective function of the training algorithm is:

$$J = \left(\frac{1}{2}\right)\|w\|^2 \tag{15}$$

For the SVM-RFE method described in Section 3, Eq. (13) gives the feature ranking criteria. To solve the objective in (15) quadratic programming methods have to be used. These methods are not easy to implement when compared to other classifier algorithms. Libraries like LIBSVM [75] can be used to implement wrapper techniques. Since SVM is a binary classifier, in this paper we use the one-vs-all classification technique for multiclass problems. In one-vs-all approach, different binary classifiers are trained wherein each one is trained to distinguish samples of a single class from all the remaining samples. It is a simple method which is proven to be robust and accurate when compared to other approaches [53,76].

7.2. Radial Basis Function Network

A Radial Basis Function Network [6,77] (RBFN or RBF) is a type of feed-forward Neural Network. The network structure consists of three layers: input layer, hidden layer and output layer as shown in Fig. 3. The input layer consists of a single D dimensional input. The hidden layer is composed of radially symmetric Gaussian kernels given by:

$$\phi_j = e^{-\frac{\|x-x_j\|^2}{\sigma^2}} \tag{16}$$

where $j = 1$ *to* M, M being the number of kernels, x_j the jth kernel centroid and ϕ_j is calculated for every input data vector. The output layer node is connected to each kernel in the hidden layer with a weight W_j. The output Y can be calculated using the equation:

$$Y = \sum_{j=1}^{M} W_j \phi_j \tag{17}$$

In the training phase the weight vector is calculated for all the samples in the training set. The weight vector can be calculated using:

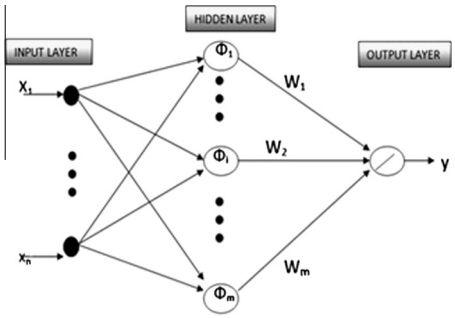

Fig. 3. Radial Basis Function Network structure.

$$W = \phi^{-1}.Y \tag{18}$$

where the ϕ matrix element ϕ_{ij} gives the value of ϕ_j for the ith sample. For the testing set the output values can be calculated for each data using (18) since the weight vector is known in the training phase. To find the centroids of the kernel function, clustering algorithms [77–81] are employed to enhance the generalizing capability of the classifier. In [78] the RBF network is used to EEG data classification wherein three clustering methods for the RBF are explored.

7.3. Validation methods

It is also important to choose validation method (classifier accuracy) for the chosen classifier. This method is known as cross-validation in which the input data is split into training and testing sets and the test set (unseen by the method or classifier) is validated against the training set to check if the classifier can reproduce the known output. Several types of cross-validation are used in literature. One of the simplest methods is the *2*-fold cross-validation wherein the data is randomly split into training and test sets. An extension of the 2-fold cross-validation is the *K*-fold cross-validation in which the data is randomly split into *K* subsets. For training $K-1$ subsets are chosen and the remaining subset is used for testing. This process is repeated until all the subsets are used for testing. Another version of the *K*-fold cross-validation is the Leave-one out cross-validation (LOOCV) wherein *K* is equal to the number of samples i.e. each sample is used for testing and the rest of the samples are used for training. This process is done until all the samples are tested. For model selection (feature selection or classifier) the training set may be further split into training and validation sets. The prediction of the validation set is used to reinforce the model selection. In [34] the authors address the problem of over fitting found when comparisons of the feature selections are made. The authors argue the cross-validation methods can contribute to over estimating the model performance. The author tests the SFS and SFFS method on various datasets with *k*-NN [18,6] classifier as the wrapper. They use the 2-fold and LOOCV cross-validation methods to compare the results of the two feature selection algorithms.

8. Results

In this section we present results of applying some of the feature selection algorithms on seven datasets using the two classifiers mentioned in Section 7. We implement two filter methods given by (1) and (4) wherein the features are ranked and the performance is obtained for the top ranked feature and subsequently adding the next ranked feature one by one to cover the whole feature set. For the MI criteria, the MI is found for each feature and the output class. In wrapper techniques we implement the SFFS algorithm and the CHCGA algorithm with the performance of the two classifiers as the objective function. Out of the seven datasets, five are from the UCI Machine Learning Repository [82] and the sixth is Fault Mode data from MKS Instruments. The Fault Mode data is obtained from a 5kW, 40 MHz RF Power generator from MKS Instruments. The generator consists of several subsystems including an AC/DC converter, RF Power Amplifier, RF Sensor, and Digital Control System given in Fig. 4.

Sensors are designed into the RF generator to provide accurate power delivery and facilitate health monitoring during critical semiconductor/thin film processing steps. The data used in the study was created by intentionally seeding various faults and collecting generator sensor data during an automated RF power sweep designed in LabView™. The seventh dataset is the discrete Fault Mode data obtained by applying histogram to the continuous data. The number of variables in each dataset is given in Table 1. The datasets are divided into 50% training and 50% testing sets and the performance measure was the prediction accuracy of the test set. For all the results, the test set was not shown to the feature selection algorithm. LIBSVM [75] library was used to implement the SVM classifier. For the RBF classifier, clustering must be performed to determine the optimum number of kernels, variance and their respective centres for (16). As suggested in [78] we perform an exhaustive search using *k*-means [6] technique to determine the RBF kernel centres. Since the input to the classifier varies according to the subset selected by the feature selection algorithm, using the parameters obtained from the whole dataset will bring down the efficiency of the classifier and results will not be accurate. Due to this, clustering must be done for each subset which would increase the number of computations significantly. To reduce the number of

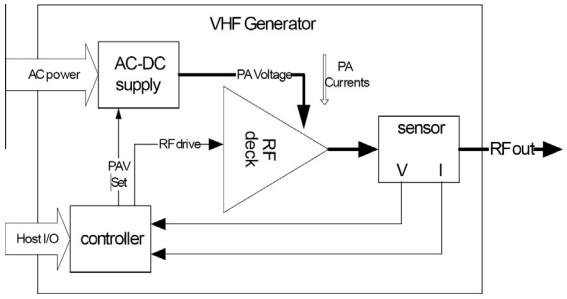

Fig. 4. RF Generator block diagram.

Table 1
Number of variables in the datasets used.

Dataset	No. of Features
Breast cancer	10
Diabetes	8
Ionosphere	34
Liver disorder	6
Medical	19
Fault mode	16

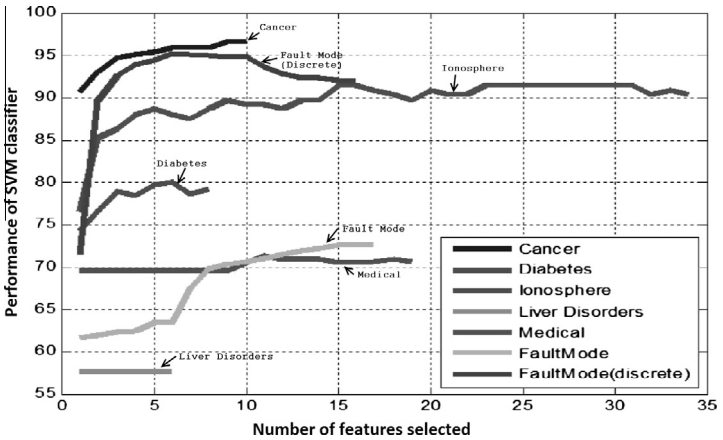

Fig. 5. Results for correlation criteria using SVM.

computations we first find the number of kernels using the exhaustive search in range *D to 2∗D*. The variance in (16) is found each time by searching in the range [0.1, 0.2, 0.3, ...10]. By doing this we obtain better classification. To avoid indirectly learning the test data, the RBF exhaustive search is done on the training data itself. For the CHCGA algorithm, the GA parameters are: population size = 40, mutation rate = 0.4, maximum number of generations = 800. The data from the RF generator was also used to perform fault prediction analysis using the above mentioned algorithms which can be found in [83].

In Fig. 5 the result for the first ranking method is given. The *x*-axis is the number of features and the *y*-axis is the performance of the subset containing the specified number of features. Fig. 6 gives the result for the MI criteria (4), Fig. 7 gives the result of applying SFFS algorithm with SVM as the wrapper and Fig. 8 gives the result of applying SFFS with RBF as the wrapper. In Fig. 5 we can see that the performance for the first ranked feature of Cancer and Ionosphere give low performance due to the calculation of MI. Since we are approximating the PDF of a single feature and the output class distribution, calculation of MI will not be accurate and is easily influenced by marginal densities [22]. The irregular graph for the filter methods also

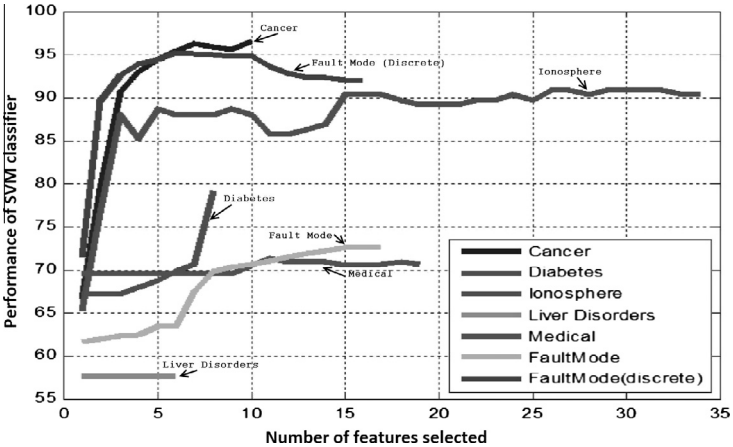

Fig. 6. Result for MI using SVM.

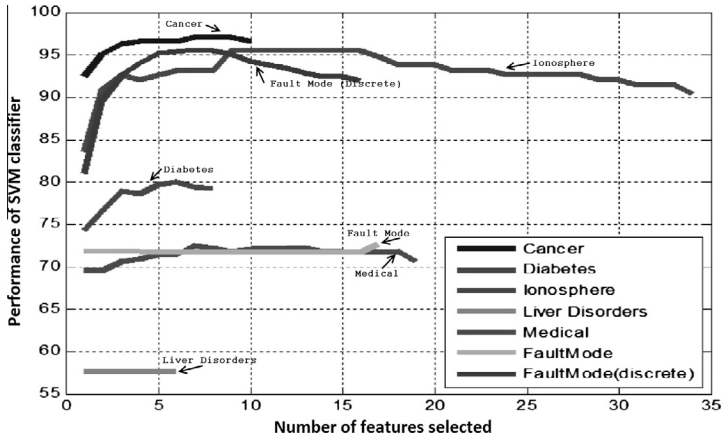

Fig. 7. Results for SFFS using SVM.

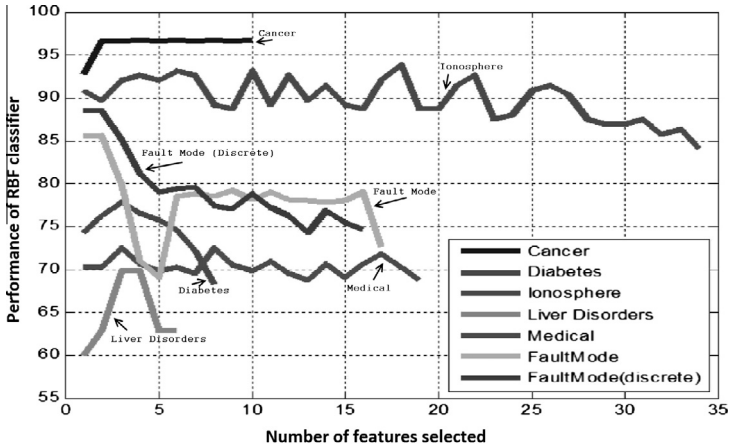

Fig. 8. Results for SFFS using RBF.

proves that the ranking methods are trivial. Since SFFS can produce only one subset, we generate subsets of lower size by backward elimination or a subset with larger number of features by forward addition to the subset selected by SFFS. By doing this we can see that the maximum performance is always obtained from the SFFS algorithm and the performance of all other subsets are either equal to or lower than the maximum performance. Fig. 8 gives the results of applying SFFS using RBF classifier with the optimization procedure stated above. It can be seen in Fig. 8 that the graph does not give a clear idea of the feature selection due to optimization of RBF parameters for each subset. If the sample values are close to the centre of the kernel, the weight contribution will be higher. It can be noticed in Fig. 7 that for Liver Disorder data the performance is the same for any subset which is interesting since the same result is not obtained from RBF due to the optimization of the RBF classifier.

In Table 2 we give the results of running the CHCGA algorithm with objective to find the subset that has the maximum classifier performance using SVM. We can see that the results in Fig. 7 and Table 2 are close. Table 3 gives the results of running the CHCGA algorithm with RBF as the wrapper.

Table 2
Experimental results for CHCGA with SVM.

Dataset	Maximum performance	No. of selected features
Breast cancer	97.361	5
Diabetes	80.469	7
Ionosphere	94.286	16
Liver disorder	57.558	4
Medical	73.2	8
FaultMode	76.392	6
FaultMode (discrete)	98.829	6

Table 3
Experimental results for CHCGA with RBF.

Dataset	Maximum performance	No. of selected features
Breast cancer	96.774	5
Diabetes	76.822	7
Ionosphere	94.285	16
Liver disorder	72.675	4
Medical	70	8
FaultMode	79.639	6
FaultMode (discrete)	82.799	6

9. Conclusion

In this paper we have tried to provide an introduction to feature selection techniques. The literature on feature selection techniques is very vast encompassing the applications of machine learning and pattern recognition. Comparison between feature selection algorithms can only be done using a single dataset since each underlying algorithm will behave differently for different data.

Feature selection techniques show that more information is not always good in machine learning applications. We can apply different algorithms for the data at hand and with baseline classification performance values we can select a final feature selection algorithm. For the application at hand, a feature selection algorithm can be selected based on the following considerations: simplicity, stability, number of reduced features, classification accuracy, storage and computational requirements. Overall applying feature selection will always provide benefits such as providing insight into the data, better classifier model, enhance generalization and identification of irrelevant variables. For the results in this paper we use the classifier accuracy and the number of reduced features to compare the feature selection techniques. We have also successfully used feature selection for improving predictor performance and for fault prediction analysis of Fault Mode data.

References

[1] Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res 2003;3:1157–82.
[2] Guyon I, Weston J, Barhill S, Vapnik V. Gene selection for cancer classification using support vector machines. Mach Learn 2002;46:389–422.
[3] Ding C, Peng H. Minimum redundancy feature selection from microarray gene expression data. J Bioinform Comput Biol 2005;3:185–205.
[4] Chuang L-Y, Chang H-W, Tu C-J, Yang C-H. Improved binary PSO for feature selection using gene expression data. Comput Biol Chem 2008;32:29–38.
[5] Lazar C, Taminau J, Meganck S, Steenhoff D, Coletta A, Molter C, et al. A survey on filter techniques for feature selection in gene expression microarray analysis. IEEE/ACM Trans Comput Biol Bioinform 2012;9.
[6] Alpaydin E. Introduction to machine learning. The MIT Press; 2004.
[7] Law MH, Figueiredo M rio AT, Jain AK. Simultaneous feature selection and clustering using mixture models. IEEE Trans Pattern Anal Mach Intell 2004;26.
[8] Kohavi R, John GH. Wrappers for feature subset selection. Artif Intell 1997;97:273–324.
[9] Langley P. Selection of relevant features in machine learning. In: AAAI fall symp relevance; 1994.
[10] Blum AL, Langley P. Selection of relevant features and examples in machine learning. Artif Intell 1997;97:245–70.
[11] John GH, Kohavi R, Pfleger K. Irrelevant features and the subset selection problem. In: Proc 11th int conf mach learn; 1994. p. 121–9.
[12] Battiti R. Using mutual information for selecting features in supervised neural net learning. IEEE Trans Neural Networks 1994;5.
[13] Forman G. An extensive empirical study of feature selection metrics for text classification. J Mach Learn Res 2003;3:1289–306.
[14] Kwak N, Choi C-H. Input feature selection for classification problems. IEEE Trans Neural Networks 2002;13:143–59.
[15] Comon P. Independent component analysis a new concept? Signal Process 1994;36:287–314.
[16] Torkkola K. On feature extraction by non-parametric mutual information maximization. J Mach Learn Res 2003;3:1415–38.
[17] Fleuret F. Fast binary feature selection with conditional mutual information. Mach Learn Res 2004;5:1531–55.
[18] Bekkerman R, El-Yaniv R, Tishby N, Winter Y. Distributional word clusters vs. words for text categorization. J Mach Learn Res 2003;3:1245–64.
[19] Caruana R, de S V. Benefitting from the variables that variable selection discards. J Mach Learn Res 2003;3:1245–64.
[20] Koller D, Sahami M. Towards optimal feature selection. In: ICML, vol. 96; 1996. p. 284–92.
[21] Davidson JL, Jalan J. Feature selection for steganalysis using the mahalonobis distance. In: Proc SPIE 7541, Media Forensics and Security II 7541; 2010.
[22] Yang Y, Perdersen JO. A comparative study on feature selection in text categorization. International conference on machine learning; 1997.
[23] Javed K, Babri HA, Saeed M. Feature selection based on class-dependent densities for high-dimensional binary data. IEEE Trans Knowl Data Eng 2010;24.
[24] Peng H, Long F, Ding C. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 2005;27.
[25] Kira K, Rendell LA. The feature selection problem: traditional methods and a new algorithm. In: Proceedings of tenth national conference on artificial intelligence; 1992. p. 129–34.
[26] Acuna E, Coaquira F, Gonzalez M. A comparison of feature selection procedures for classifier based on kernel density estimation. Proc Comput Commun Control Technol 2003;1:468–72.
[27] Stoppiglia H, Dreyfus G, Dubios R, Oussar Y. Ranking a random feature for variable and feature selection. J Mach Res 2003;3:1399–414.
[28] Liu H, Setiono R. A probabilistic approach to feature selection a filter solution. In: International conference on machine learning - ICML; 1996. p. 319–27.
[29] Xu Z, King I, Lyu MR-T, Jin R. Discriminative semi-supervised feature selection via manifold regularization. IEEE Trans Neural Networks 2010;21.
[30] Narendra P, Fukunaga K. A branch and bound algorithm for feature subset selection. IEEE Trans Comput 1977;6:917–22.
[31] Goldberg D. Genetic algorithms in search, optimization and machine learning. Addison-Wesley; 1989.
[32] Kennedy J, Eberhart RC. Particle swarm optimization. In: Proc IEEE int'l conf on neural networks, IV; 1995. p. 1942–1948.
[33] Pudil P, Novovicova J, Kittler J. Floating search methods in feature selection. Pattern Recog Lett 1994;15:1119–25.
[34] Reunanen J. Overfitting in making comparisons between variable selection methods. J Mach Learn Res 2003;3:1371–82.
[35] Pudil P, Novovicova J, Kittler J, Paclik P. Adaptive floating search methods in feature selection. Pattern Recog Lett 1999;20:1157–63.

[36] Sun Y, Babbs C, Delp E. A comparison of feature selection methods for the detection of breast cancers in mammograms: adaptive sequential floating search vs. genetic algorithm. Conf proc IEEE eng med biol soc, vol. 6.
[37] Nakariyakul S, Casasent DP. An improvement on floating search algorithms for feature subset selection. Pattern Recog 2009;42:1932–40.
[38] Stearns S. On selecting features for pattern classifiers. In: Proceedings of the 3rd international conference on pattern recognition; 1976. p. 71–5.
[39] Alexandridis A, Patrinos P, Sarimveis H, Tsekouras G. A two-stage evolutionary algorithm for variable selection in the development of rbf neural network models. Chemomet Intell Lab Syst 2005;75:149–62.
[40] Jouan-Rimbaud D, Massart DL, Leardi R, Noord OED. Genetic algorithms as a tool for wavenumber selection in multivariate calibration. Anal Chem 67.
[41] Yang J, Honavar V. Feature subset selection using a genetic algorithm. IEEE Intell Syst Appl 1998;13:44–9.
[42] Puch W, Goodman E, Pei M, Chia-Shun L, Hovland P, Enbody R. Further research on feature selection and classification using genetic algorithm. In International conference on genetic algorithm; 1993. p. 557–64.
[43] Eshelman L. The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination. In: Rawlins GJE, editor. In foundations of genetic algorithms. Morgan Kauffman; 1991.
[44] Cordon O, Damas S, Santamaria J. Feature-based image registration by means of the chc evolutionary algorithm. Image Vis Comput 2006;24:525–33.
[45] Oliveira L, Sabourin R, Bortolozzi F, Suen C. A methodology for feature selection using multiobjective genetic algorithms for handwritten digit sting recognition. Int J Pattern Recog Artif Intell 2003;17:903–29.
[46] Ferri F, Pudil P, Hatef M, Kittler J. Comparative study of techniques for large-scale feature selection. Pattern Recog Pract 1994:403–13.
[47] Kudo M, Sklansky J. Comparison of algorithms that select features for pattern classifiers. Pattern Recog 2000;33:327–36.
[48] Tu C-J, Chuang L-Y, Chang J-Y, Yang C-H. Feature selection using pso-svm. Int J Comput Sci 2006;33.
[49] Alba E, Garcia-Nieto J, Jourdan L, Talbi E-G. Gene selection in cancer classification using pso/svm and ga/svm hybrid algorithms. Evol Comput 2007:284–90.
[50] Mundra PA, Rajapakse JC. Svm-rfe with mrmr filter for gene selection. IEEE Trans Nanobiosci 2010;9.
[51] Boser B, Guyon I, Vapnik V. A training algorithm for optimal margin classifiers. In: In fifth annual workshop on computational learning theory; 1992. p. 144–52.
[52] Chapelle O, Keerthi SS. Multi-class feature selection with support vector machines; 2008.
[53] Archibald R, Fann G. Feature selection and classification of hyperspectral images with support vector machines. IEEE Geosci Remote Sens Lett 2007;4.
[54] Neumann J, Schnorr C, Steidl G. Combined svm-based feature selection and classification. Mach Learn 2005;61:129–50.
[55] Setiono R, Liu H. Neural-network feature selector. IEEE Trans Neural Networks 1997;8:654–62.
[56] Romero E, Sopena JM. Performing feature selection with multilayer perceptrons. IEEE Trans Neural Networks 2008;19.
[57] Stracuzzi DJ, Utgoff PE. Randomized variable elimination. J Mach Learn 2004;5:1331–64.
[58] Wu D, Zhou Z, Feng S, He Y. Uninformation variable elimination and successive projections algorithm in mid-infrared spectra wavenumber selection. Image Signal Process 2009.
[59] Centner V, Massart D-L, de Noord OE, de Jong S, Vandeginste BM, Sterna C. Elimination of uninformative variables for multivariate calibration. Anal Chem 1996;68:3851–8.
[60] Alsberg BK, Woodward AM, Winson MK, Rowl JJ, Kell DB. Variable selection in wavelet regression models. Anal Chim Acta 1998;368:29–44.
[61] Peng Y, Xuefeng Z, Jianyong Z, Yunhong X. Lazy learner text categorization algorithm based on embedded feature selection. J Syst Eng Electron 2009;20:651–9.
[62] Xing E, Karp R. Cliff: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. In: 9th International conference on intelligence systems for molecular biology; 2001.
[63] Pudil P, Novovicova J, Kittler J. Feature selection based on the approximation of class densities by finite mixtures of the special type. Pattern Recog 1995;28:1389–98.
[64] Mitra P, Murthy C, Pal SK. Unsupervised feature selection using feature similarity. IEEE Trans Pattern Anal Mach Intell 2002;24.
[65] Pal SK, De RK, Basak J. Unsupervised feature evaluation: a neuro-fuzzy approach. IEEE Trans Neural Networks 2000;11.
[66] Zhu X. Semi-supervised learning literature survey. Tech rep 1530, computer sciences. University of Wisconsin-Madison; 2005.
[67] Zhao Z, Liu H. Semi-supervised feature selection via spectral analysis. In: Proc 7th SIAM data mining conf (SDM); 2007. p. 641–6.
[68] Haury A-C, Gestraud P, Vert J-P. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. PLoS ONE 2011;6:e28210.
[69] T A, T H, de Peer Y V, P D, Y S. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. Bioinformatics 2010;26:392–8.
[70] Dunne K, Cunningham P, Azuaje F. Solutions to instability problems with sequential wrapper-based approaches to feature selection. Tech rep, Trinity College; 2002.
[71] Kalousis A, Prados J, Hilario M. Stability of feature selection algorithms: a study on high dimensional spaces. Knowl Inform Syst 2007;2:95–116.
[72] Somol P, Novovicova J. Evaluating stability and comparing output of feature selectors that optimize feature subset cardinality. IEEE Trans Pattern Anal Mach Intell 2010;32:1921–39.
[73] Yang F, Mao K. Robust feature selection for microarray data based on multicriterion fusion. IEEE/ACM Trans Comput Biol Bioinform 2011;8.
[74] Dietterich T. Machine learning research: four current directions. Artif Intell Mag 1997;18:97–136.
[75] Chang C-C, Lin C-J. Libsvm: a library for support vector machines. ACM Trans Intell Syst Technol 2011;2:1–27.
[76] Rifkin R, Klautau A. In defense of one-vs-all classification. J Mach Learn Res 2004;5:101–41.
[77] Haykin S. Neural networks: a comprehensive foundation. Prentice Hall; 2008.
[78] Cinar E, Sahin F. A study of recent classification algorithms and a novel approach for EEG data classification. In: IEEE 2010 international conference on systems, man and cybernetics; 2010.
[79] Loong AS, Choon OH, Chin LH. Criterion in selecting the clustering algorithm in radial basis functional link nets. WSEAS Trans Syst 2008;7:1290–9.
[80] Marcos JV, Hornero R, Alvarez D. Radial basis function classifiers to help in the diagnosis of the obstructive sleep apnoea syndrome from nocturnal oximetry. Med Biol Eng Comput 2008;46:323–32.
[81] Hongyang L, He J. The application of dynamic k-means clustering algorithm in the center selection of rbf neural networks. In: Proc 3rd international conference on genetic and evolutionary computing, vol. 177; 2009. p. 488–91.
[82] http://archive.ics.uci.edu/ml/.
[83] Chandrashekar G, Sahin F. In-vivo fault prediction for rf generators using variable elimination and state-of-theart classifiers. 2012 IEEE international conference on systems, man, and cybernetics October 14–17, COEX, Seoul, Korea; 2012.

Girish Chandrashekar received his B.E. degree in Electronics and Communication from M.S. Ramaiah Institute of Technology in 2009 and M.S. degree in Electrical Engineering from Rochester Institute of Technology in 2013. His research interests include Robotics, Image Processing and Machine Learning.

Ferat Sahin received his M.Sc. and Ph.D. in Electrical Engineering from Virginia Polytechnic Institute and State University in 1997 and 2000, respectively. He joined Rochester Institute of Technology in 2000, where he works as an associate professor. He is also the director of Multi Agent Bio-Robotics Laboratory. His current fields of interests are System of Systems, Swarm Intelligence, Robotics, Distributed Computing, and Structural Bayesian Network Learning.