Improving Floating Search Feature Selection using Genetic Algorithm

Kanyanut Homsapaya* & Ohm Sornil

Graduate School of Applied Statistics, National Institute of Development Administration, Bangkok, Thailand
*E-mail: kanyanut.hom@stu.nida.ac.th

Abstract. Classification, a process for predicting the class of a given input data, is one of the most fundamental tasks in data mining. Classification performance is negatively affected by noisy data and therefore selecting features relevant to the problem is a critical step in classification, especially when applied to large datasets. In this article, a novel filter-based floating search technique for feature selection to select an optimal set of features for classification purposes is proposed. A genetic algorithm is employed to improve the quality of the features selected by the floating search method in each iteration. A criterion function is applied to select relevant and high-quality features that can improve classification accuracy. The proposed method was evaluated using 20 standard machine learning datasets of various size and complexity. The results show that the proposed method is effective in general across different classifiers and performs well in comparison with recently reported techniques. In addition, the application of the proposed method with support vector machine provides the best performance among the classifiers studied and outperformed previous researches with the majority of data sets.

Keywords: *classification; evaluation; feature selection; floating search; genetic algorithm.*

1 Introduction

Classification, a process for predicting the class of a given input data, is one of the most fundamental tasks in data mining. A number of available methods are commonly used for data classification, such as: decision trees; rule-based, probabilistic and instance-based methods; support vector machines (SVMs); and neural networks. Noisy and irrelevant data are major obstacles to data mining. They adversely affect system performance in terms of classification accuracy, building time, size, and interpretability of the model obtained [1,2]. These issues can introduce new properties in the problem domain. For example, noise can lead to the creation of small clusters of examples of a particular class in areas of the domain corresponding to another class, or it can cause missing data of examples located in key areas within a specific class [3].

Received November 27th, 2016, Revised May 4th, 2017, Accepted for publication October 19th, 2017.
 ISSN: 2337-5787, DOI: 10.5614/itbj.ict.res.appl.2017.11.3.6

Selecting features relevant to the problem is a critical first step in classification, especially when applied to large datasets. The aim is to select a representative subset of highly relevant dimensions while removing irrelevant and redundant ones [4]. Feature selection can significantly improve the running time of a machine-learning algorithm as well as improve the quality of the model.

Consequently, Bins and Draper [5] proposed a technique to reduce a large set of features (1,000) to a much smaller subset without removing any highly important features or decreasing classification accuracy. There are three steps in the algorithm: first, irrelevant features are removed using a modified form of the relief algorithm [6]; second, redundant features are eliminated using K-means clustering [7]; and, finally, a combinatorial feature selection algorithm is employed to the current feature subsets using the sequential floating backward selection (SFBS) algorithm. The basic concept is to filter feature subsets in each step until the smallest possible one is obtained.

Floating search methods dynamically increase and decrease the number of features until the desired target is reached. Instead of fixing the number of forward/backward steps, we can allow values to float so that they can be flexibly changed without pre-setting parameters, which is different from the *plus l take away r* method. Nonetheless, floating search has a tendency to become stuck at a local optimum solution since there is almost no chance to improve the solution's quality [8]. For this reason, we present an improvement to the floating search algorithm with the aim of removing some of its drawbacks and to aid finding a solution closer to the optimal one.

In this article, we propose a technique to improve the effectiveness of the floating search feature selection method that leads to a higher classification rate. Our method employs a genetic algorithm to enrich and improve the resultant features after each iteration of the sequential forward feature search (SFFS) process.

2 Background

2.1 Feature Selection Methods

Two important components of the feature selection process, subset generation and subset evaluation, are shown in Figure 1. The subset generation engine identifies feature subset candidates, while subset evaluation measures the quality of the subsets. Lastly, in order to terminate the process, a stopping criterion is tested in every iteration.

There are three main types of feature selection methods: filter, wrapper, and hybrid. Wrapper methods rely on a classification algorithm employed as the subset evaluation process for feature subsets [9]. Maroño *et al.* [10] proposed a wrapper method by applying ANOVA decomposition and functional networks to create the evaluation function. In general, the wrapper approach gives better performance than the filter approach since the feature selection process is optimized for the specific classification algorithm. Nevertheless, when wrapper methods are applied to huge dimensional datasets, they will incur high computational cost and may become unfeasible.

Filter methods use an independent criterion that relies on general characteristics of the data to evaluate and select feature subsets without involving a classification algorithm. Common evaluation functions usually are measures such as distance, mutual information (MI), dependency or entropy, calculated directly from the training data. Karegowda, *et al.* [11] developed a filter-based technique in a cascade fashion with a genetic algorithm (GA), using a correlation-based criterion.

Hybrid methods exploit the positive aspects of both wrapper and filter methods [4]. They utilize a filter-based technique to select highly representative features and apply a wrapper-based technique to add candidate features and evaluate the candidate subsets in order to select the best ones. This not only reduces the dimensionality of the data but also decreases the computational cost and improves classification performance. Somol, *et al.* [8] proposed a hybrid SFFS method by employing an evaluation function to filter some features and using a wrapper criterion to identify the optimal feature subset. Their experimental results showed that the method yielded a promising classification accuracy.

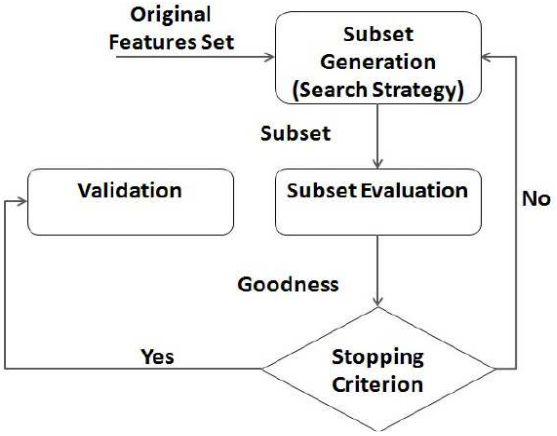

Figure 1 The feature selection process.

The sequential forward search (SFS) method operates in a forward search manner starting with an empty set and adds one feature subset during each round until a new feature subset that maximizes the criterion function value is found, whereas the sequential backward search (SBS) method starts with a full feature subset and eliminates a feature on each iteration until a predetermined criterion is satisfied. A drawback of both methods is that they have a nesting effect problem, which means that discarded features cannot be re-selected and selected features cannot be removed later. Since these algorithms do not examine all possible feature subsets, they are not guaranteed to produce an optimal result. Generalized forms GSFS and GSBS based on group collection feature testing are better solutions but at the cost of increased computational time. The *plus l take away r* method was proposed to take care of the nesting problem [12].

2.2 Floating Search Methods

Pudil, *et al.* [13] proposed floating search methods based on two main categories: the search process in a forward direction (SFFS) and in a backward direction (SBFS). These methods use a criterion function to select a feature and compare candidate subsets. SFFS and SBFS can be classified as a wrapper or a filter approach depending on the criterion function used. They perform well but the computational time is long, especially with large datasets. The floating search methods can be viewed as predictive text algorithms (PTAs) without the use of a fixed parameter. They have been shown to give very good performance (close to optimum results) and to overcome the nesting problem. SFFS, SBFS, and bidirectional selection as a combination of both are greedy search algorithms that add or discard features one at a time [13]. The floating search method consists of two phases: forward and backward. SFFS starts with an empty set and sequentially adds one feature at a time. The structure of the floating search algorithm is shown in Figure 2.

SBFS, the counterpart of the forward search, is initialized with a full set and sequentially eliminates one feature at a time after execution of SFFS. The SFFS search selects the best unselected feature according to a criterion function to form a new feature subset, while the SBFS search iteratively determines which members of the selected subset are to be removed if the remaining set improves performance according to the same criterion function as used in forward search. The algorithm loops back to forward search until the stopping condition is reached. There are disadvantages when using either algorithm. With SFFS it is not possible to succeed in eliminating redundant features generated in the search process, whereas SBFS cannot re-calculate evaluation feature usefulness together with other features at the same time.

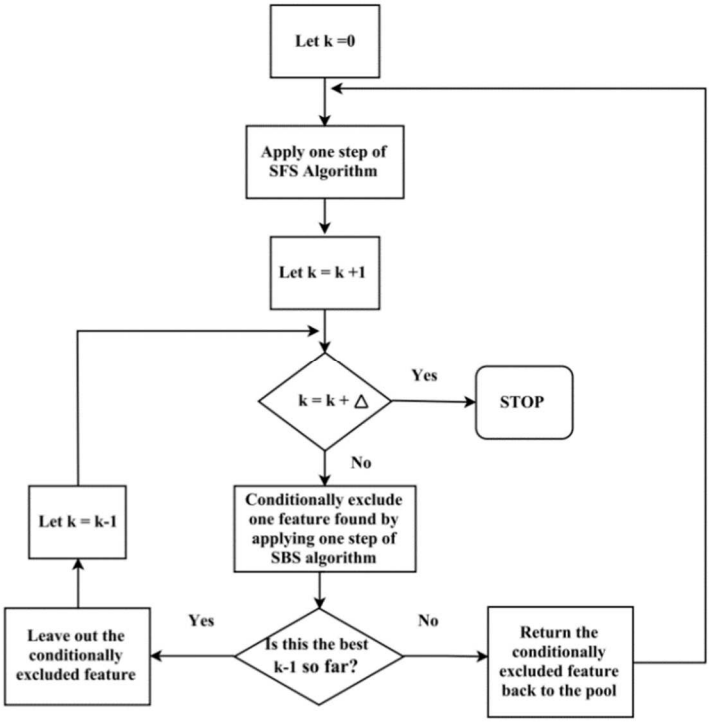

Figure 2 Structure of a floating search algorithm.

Improved versions of SFFS have been proposed in several researches to obtain better performance. Somol, *et al.* [11] present the adaptive sequential forward floating selection (ASFFS) algorithm with a parameter r, which specifies the number of features to be added in the inclusion phase, calculated dynamically. Parameter o is used in the exclusion phase to remove the maximum number of features if it improves performance. The benefit of ASFFS is that it provides a less redundant subset than the SFFS algorithm. Nakariyakul and Casasent [14] came up with an improved forward floating search algorithm, which has a new search step to check whether to replace a weak feature and remove it again until the replacement can no longer improve the criterion function. They found that this method obtained optimal solutions for many feature subsets and was less computationally intensive than exhaustive search optimal feature selection algorithms. Chaiyakarn and Sornil [15] proposed a filter-based method to return a small subset of features for classification by employing two different criterion functions in the forward and backward steps. The functions help remove redundant features, maximize inter-class distance and minimize intra-class distance.

2.3 Feature Subset Evaluation

In order to perform feature selection with a filter approach, a measure is needed to evaluate the relevance of the subset to the classification process. Several functions can be used in feature selection, such as the Mahalanobis Distance (MAHA) [16] or the Bhattacharyya Distance (BAVE) [17].

Mutual information (MI) is a widely used measure to evaluate candidate feature subsets [18]. MI can be calculated in Eq. (1) as follows:

$$I(X,Y) = H(X) + H(Y) - H(X,Y), \tag{1}$$

where H is an entropy function, Y is a class attribute, and X is the selected feature, given a random variable X, such that Eq. (2) can be defined.

$$X \begin{cases} 0 \text{ with probability of } p \\ 1 \text{ with probability of } 1 - p, \end{cases}$$

$$H(X) = -p \log p - (1 - p) \log(1 - p) = H(p) \tag{2}$$

Note that the entropy function does not depend on the values that the random variable takes (0 and 1 in this case) but only depends on the probability distribution, p(x).

2.4 Genetic Algorithm

The genetic algorithm (GA), introduced by John Holland in 1975 [19], is an adaptive optimization search algorithm for finding an optimal solution inspired by natural selection in biological systems. The genes of an organism are gathered into structures called chromosomes; a set of chromosomes is referred to as a population. In general, there are three operations employed in GAs. First, *selection* is an operator for selecting potentially useful solutions for recombination and is achieved by either tournament or roulette wheel selection. Second, *crossover* refers to the process of producing an offspring chromosome from two matching parent chromosomes.

There are various types of crossovers: single point crossover, two-point crossover, and uniform crossover. Crossover is an operation to produce child subsets recombined from parental chromosomes that consist of splitting chromosome pairs at random. Third, *mutation* causes genetic diversity of chromosomes by making random binary changes in a chromosome, thus adversely affecting their fitness value. These principles have led to new solutions in the pursuit of better search solutions.

GAs have been successfully applied to feature selection [20] with the objective to save computational time without processing in an exhaustive fashion, which is achieved by finding promising regions and selecting quality feature subsets. Furthermore, hybrid GAs [21] are involved in a new search method that includes local search operators to improve the fine-tuning quality of a simple GA search.

The fitness function, based on the principle of survival of the fittest, is the process whereby a GA evaluates each individual's fitness and obtains the optimal solution after applying the genetic operators. This process is repeated many times and over many generations until the stopping criterion is satisfied. For feature selection, the feature subsets are represented as a binary: a feature is either included or not included in the feature subset.

3 The Proposed Algorithm

We now discuss our algorithm to select the best subset of size d of a total of D features, as shown in Figure 3. The inclusion step using MI as the criterion function (J) is executed to create a set of candidates for inclusion. In the exclusion step, a candidate feature subset is used to generate smaller subsets from the result of the inclusion step by removing one feature and re-evaluating them. A selection subset of size k + 1 is generated and compared to the previously best subset of size k + 1 from the inclusion part. If evaluation of the new subset is better qualified than the formerly selected set, the exclusion step retains the better one and iterates to smaller subsets, or else the algorithm goes back to the inclusion step.

Our feature improvement step based on GA is included after the exclusion step in each iteration. The objective is to replace the weakest feature by checking whether removing any feature in the currently selected feature subset and adding a promising one at each sequential step potentially improves the current feature subset. The chromosome structure consists of binary genes corresponding to individual features. The value of 1 at the i^{th} gene means that the i^{th} feature is selected; otherwise it is 0.

The initial population is generated from the resulting feature exclusion subsets of size k + 1 from the exclusion step by first removing the weakest features from the best subset resulting in a subset of size k. Each remaining feature is thus added to that subset, generating the niched initial population for GA. The fitness function used in this study is MI. Then, a new population is created by selection, crossover and mutation. The process is terminated when the current feature set reaches the size of D-2 features.

Input :Y_m is a feature set, m is a predefined number of selected features, J is a criterion function, P_c is the probability of crossover, P_m is the probability of mutation, *Population* is a set of individuals, *max_generation* is the maximum number of generations, and *Fitness* is a function which determines the quality of the individuals.

Output :The best solution in all generation.

(1) Feature Inclusion

Initialize :$Y_0 = \{\emptyset\}$; $m = 0$

Find the best feature and update Y_m

$x^+ = \arg\max_{x \in Y_m} [J(Y_m - x)]$

$Y_m = Y_m + x^+$; $m = m + 1$

(2) Feature Exclusion

Find the worst feature

$x^- = \arg\max_{x \in Y_m} [J(Y_m - x)]$

If $J(Y_m - x^-) > J(Y_m)$ then

$Y_m+1 = Y_m - x$;

Go to Step 3 Else Go to Step1

(3) Feature Improvement

Repeat

population ← SBFS feature subsets Y_m

generation =0;

loop for i **from** 1 **to** size(*Population*)**do**

s1 ← **selection (***Population,* Fitness)

s*2* ← **selection (***Population,* Fitness)

child ← **crossover(** *s1,s2)*with *pc* and check feasibility of *n* element

child ← **mutate(***child)* with *pm* and check feasibility of *n* element

Fitness(child)

generation =generation +1

until *generation* < *max_generation*

$m = m + 1$

return the best individual solution Y_m

Figure 3 Pseudo-code of the proposed algorithm.

We now provide an illustrative example of how the proposed algorithm works and how it improves SFFS. Assume that the first five feature sets selected by the SFS method at each size are {f1}, {f1, f4}, {f1, f4, f5}, {f1, f4, f5, f7} with the corresponding J values of 4.1, 6.2, 9.1 and 10.2, respectively, and the next iteration is to determine subsets with five features.

3.1 Feature Inclusion

A feature is added to the feature subset. The SFS method adds up to a total of five features to the subset: J(f1,f4,f5,f7,f6) = 13. Assume that feature f6 is chosen using the SFS method and J for the 5th features is 14.

3.2 Feature Exclusion

A feature is removed from the feature subset. The SBS method is applied in this step by backtracking and conditionally removing one feature from the subset selected in Step 1, returning an improved subset, e.g. (f1,f5,f6,f7) with j value = 11, (f1,f4,f5,f7) with j value = 9, (f1,f4,f7,f6) with j value = 9.5, and (f4,f5,f7,f6) with j value = 10. In this case, the best feature subset of size 4 is (f1, f5, f6, f7).

3.3 Feature Improvement using Genetic Algorithm

The weakest feature is removed from the subset of size k from the previous step, which is (f1, f5, f6, f7), by iteratively evaluating the smaller subsets: (f1, f5, f7), (f1, f5, f6), (f5, f6, f7) and (f1, f7, f6). In this case, we assume that the best performance subset of size 3 is (f5, f7, f6). Then, each feature is added to each subset of (f5, f7, f6) in order to find the best four-feature subset, either (f5, f7, f6, f1), (f5, f7, f6, f2), (f5, f7, f6, f3), (f5, f7, f6, f4), (f5, f7, f6, f8), or (f5, f7, f6, f9.) The top n chromosomes are selected as the initial population for GA and passed through the crossover and mutation operations.

3.3.1 Crossover Operation

Crossover is a genetic operator mainly responsible for creating new solution regions in the search space to be explored; it is a random mechanism for exchanging information among strings in the mating pool [22]. Once a pair of chromosomes has been selected, crossover can take place to produce child chromosomes. A crossover point is randomly chosen from two randomly selected individuals (parents). This point occurs between two bits and divides each individual into left and right sections. Crossover then swaps the left (or the right) section of the two individuals, which we refer to as mating with a single crossover operation as follows:

Parent A) –f5, f7, f6, f2)

0	1	0	0	1	1	1	0	0	0

Parent B) – f5, f7, f6, f1)

1	0	0	0	1	1	1	0	0	0

Suppose the crossover point randomly occurs after the sixth bit, then each new child receives one half of each parent's bits:

Offspring1) – f2, f5, f7, f6)

0	1	0	0	1	1	1	0	0	0

Offspring2) – f1, f5, f7, f6)

1	0	0	0	0	1	1	0	0	0

This algorithm continues to select parental chromosomes to apply the crossover operation. Child chromosomes may have one bit more than the current size of the features subset, k. In this case, a random bit is automatically flipped to preserve the size of the chromosome (i.e. current feature set size).

3.3.2 Mutation Operation

The mutation operation is applied to all of the offspring chromosomes from the crossover step. Mutation operates at the bit level by randomly flipping bits in the new chromosome within the current population (turning a '0' into a '1', and vice versa).

Offspring1) –f5, f7, f6, f1)

1	0	0	0	1	1	1	0	0	0

After mutation) – f5, f7, f6, f2)

0	1	0	0	1	1	1	0	0	0

After all child chromosomes have passed through the mutation operator, the resultant chromosomes are evaluated by the fitness function. After this, we can discover the best performing features subset, which is (f5, f7, f6, f2). We assume that J({f5, f7, f6}) = 8.35, and that J({f5,f7, f6, f2}) = 12, which is larger than the prior largest value for four features, J = 11 Thus, the best four-feature subset becomes {f5, f7, f6, f2} with J = 12, whereas the best three-feature subset remains {f1, f4, f5} since J({f1, f4, f5}) = 9.1 > J({f5, f7, f6}) = 8.35

The improvement step helps discover subsets not discoverable by the greedy nature of SFFS. From the above example, the SFFS algorithm is not able to produce this best four-feature four subset because it cannot backtrack to the set {f5, f7, f6} as a result of J({f1, f4, f5}) = 9.1 > J({f5, f7, f6}) = 8.35 and thus cannot add feature f2 to subset {f5, f7, f6}. Note that f2 is never selected in the first best four-feature sets of the SFFS method: {f1}, {f1, f4}, {f1, f4, f5}, and {f1, f5, f6, f7}.

The example above demonstrates the advantage of our proposed algorithm. The algorithm replaces the weak feature (feature f1 in our example) in the feature set {f1,f5,f7,f6} with feature 2, which results in a new set of four features {f5, f7, f6, f2}, which has a larger J value. Therefore, the search strategy of our

proposed algorithm is more thorough than the SFFS algorithm and thus it is more effective.

3.4 Terminating Condition

After each iteration, the selection / crossover / mutation cycle continues until all possible combinations of chromosomes in the population have been evaluated. The higher the fitness value, the higher the probability of that chromosome being selected for reproduction. This generational process is repeated until a pre-determined termination condition is reached. We terminate the algorithm when the current feature set reaches $d < D$ features, where D is the total number of features in the dataset). The pseudo-code is depicted in Figure 3.

A fitness function is commonly needed in GAs to evaluate a candidate chromosome of an individual to assess whether the latter should survive or not. At each iteration, calculation of the fitness function is processed repeatedly, which, because of its simplicity, is a fast process, although it still impacts performance. In our model, we use the MI criterion as a fitness function. Basically, it measures the amount of an information feature set in a group of variables for the sake of predicting the dependent data. In addition, the fitness function to be calculated includes the calculation of the classification rate, which requires a classifier.

4 Experimental Evaluations

To evaluate the proposed feature selection algorithm, 20 standard datasets of various sizes and complexities from the UCI machine-learning repository [20] were used in the experiments. These datasets have been frequently used as a benchmark to compare the performance of classification methods and consist of a mixture of numeric, real and categorical attributes. Numeric features are pre-discretized by the method demonstrated in [23], which begins by sorting a dataset and selecting only duplicate values for the cutting point bin. After this step, the number of discrete values to represent each bin is found. The range associated with an interval is divided into k intervals depending on the number of replicated values. This modification enables the discretization process to be faster and yields a higher performance than is otherwise possible.

Details of the datasets used in the experiments are shown in Table 1. From experiments, we found that a suitable set of parameters is as follows: a population size of 4-100 individuals, a bit-flip mutation rate of 0.01, and for single point crossover, a rate of 0.75-0.85 and the number of generations is 500.

Three classification modeling techniques were used in the experiments, i.e. Classification and Regression Tree (CART), Support Vector Machine (SVM) and Naïve Bayes. Training and testing data are used as provided in the datasets. For those not providing separate testing data, a 5-fold cross validation is applied. To evaluate a feature subset, MI is applied as the criterion function.

4.1 The Classifiers

Each instance in the training set contains one class label and several feature variables. The goal of a classifier is to produce a model (based on the training data) that predicts the target values of the test data given only the test data attributes. Three classification algorithms were used in the experiments, i.e. classification and regression tree (CART), Naïve Bayes and support vector machine (SVM).

CART [24] is a well-known decision tree algorithm, which represents a series of decisions for splitting each node in the tree and assigning a class outcome to each terminal node. In their study, CART employs the Gini impurity index as the measure to build the decision tree. Consider parent node l, which contains the data that belongs to the *j*th class; the impurity function for node l is given by Eq. (3) as follow:

$$i(l) = 1 - \sum p^2 (j|l), \tag{3}$$

and the declination of impurity of the split is denoted in Eq. (4) as follow:

$$\Delta\, i(l) = i(l) - p_L\, i(n_L) - p_R\, i(n_R), \tag{4}$$

where l is a parent node, which is split into nodes n_L and n_R. After that, the CART strategy is applied by choosing the feature that maximizes the decrease of impurity $\Delta i(l)$ at each subsequent node.

The Naïve Bayes algorithm is a statistical classifier for supervised learning [24] and is based on the principle of conditional probability. It can predict class membership probabilities, such as the probability that a given sample belongs to a particular class and its performance has been shown to be excellent in some domains but poor in specific domains, e.g. those with correlated features. The classification system is based on Bayes' rule under the assumption that the effect of an attribute on a given class is independent from the other attributes. This assumption is called class conditional independence, which makes computation simple. A conditional probability model for the classifier is given as P (C_i |x). Using Bayes' theorem, we can write in Eq. (5) as follow:

$$P(C_i \mid x) = \frac{(P(C_i) * P(x|C_j)}{P(x)} \quad (5)$$

where C_i is the ith class and x is the input vector. In this case, class variable C is conditional on several feature variables $x = x_1, \ldots, x_n$.

SVMs, originally proposed by Cortes and Vapnik [25] have become important in many classification problems for a variety of reasons, such as their flexibility, computational efficiency and capacity to handle high dimensional data. They are a recent method to extract information from a dataset. Classification is achieved by a linear or nonlinear separating surface in the input space of the dataset. SVMs have been applied to a number of applications, such as bioinformatics, face recognition, text categorization, handwritten digit recognition and so forth. SVM is a binary classifier that assigns a new data to a class by minimizing the probability of error.

Given a training set of instance-labelled pairs (x_i , y_i), $i = 1, \ldots, l$, where $xi \in Rn$ and $y \in \{1, -1\}$ l, the SVM requires the solution of the following optimization problem:

$$min_{w,b,\xi} \quad \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi i,$$

$$subject\ to\ y_i\ (w^T\ \emptyset\ (X_i) + b) \geq 1 - \xi i,\ \xi i > 0 \quad (6)$$

Its dual is:

$$min_{\alpha}\ \frac{1}{2}\ \alpha^T\ Q\alpha -\ e^T \alpha = 0 \quad (7)$$

$$subject\ to\ \ y^T\ \alpha = 0,$$

$$0 \ll \alpha i\ \ll C, i = 1, \ldots., n$$

where e is a vector of all ones, $C > 0$ is the upper bound, Q is an n by n positive semi-definite matrix, and $Q_{ij} = y_i y_j K(x_i, x_j)$d, where $K\left(x_i, x_j\right) = \emptyset\ (x_i)^T\ \emptyset\ (x_j)$ is the kernel. Here, training vectors are implicitly mapped into a higher (maybe infinite) dimensional space by the function $\emptyset$.

4.2 Performance of the Proposed Techniques using Classifiers

We studied the effectiveness of the proposed feature selection using three different classification methods: CART, SVM and Naïve Bayes on 20 standard UCI datasets [26]. Two performance measures were evaluated: classification

accuracy and number of selected features. Classification accuracy is the most common and simplest measure to evaluate a classifier. It is defined as the proportion of the total number of predictions that are correct. Furthermore, a good feature selection chooses a small subset of features from the original features that is sufficient to predict the target label. The 5-fold cross validation procedure is applied to report the result figures.

Table 1 Datasets used in the experiment.

Dataset	Attribute Characteristics	No. of instances	No. of attributes	No. of Classes
Wine	Integer	178	13	3
Breast Cancer (original)	Integer	699	10	2
Breast Cancer (WDBC)	Real	569	32	2
Breast Cancer (WPBC)	Real	198	34	2
Iris	Real	150	4	3
Pima-Indian diabetes	Integer, Real	768	8	2
Abalone	Categorical, Integer, Real	4,177	8	3
Dermatology	Categorical, Real	366	34	6
Heart	Categorical, Real	270	13	2
German (Credit Card)	Categorical, Integer	1,000	20	2
Lung Cancer	Integer	32	56	3
Soybean	Integer	307	35	4
Spambase	Integer, Real	4,601	57	2
Glass Identification	Real	214	10	7
Teaching Assistant	Categorical, Integer	151	5	3
Contact Lens	Categorical	24	4	3
Sonar	Real	208	60	2
Statlog (Australian)	Categorical, Integer, Real	690	14	2
Ionosphere	Integer, Real	351	34	2
Image Segmentation	Real	2,310	19	7

The results in Table 2 show that the classification accuracy was noticeably enhanced by the proposed algorithm with all classifiers compared to that without feature selection. The best performance was where the accuracy achieved 100% with 13, 22, 2, and 3 features selected for the Wine, Soybean, Contact Lenses and Iris datasets, respectively, using SVM. Additionally, high classification accuracy was achieved with small feature subsets Ionosphere, Soybean, Breast Cancer (WDBC), and Statlog (Australian).

Table 2 Classification Effectiveness: classification accuracy (%) and resulted number of selected features in parenthesis.

Dataset	Origi-nal datasets	No. of attributes	Proposed method with CART	Proposed method with SVM	Proposed method with Naïve Bayes
Wine	89.87%	13	100.00%(7)	100.00%(7)	97.14%(7)
Breast Cancer (Original)	93.13%	10	97.82%(5)	97.85%(5)	95.68%(5)
Breast Cancer (WDBC)	92.23%	32	95.49%(9)	96.13%(9)	91.00%(9)
Breast Cancer (WPBC)	72.00%	34	83.00%(6)	86.26%(6)	80.00%(6)
Iris	94.00%	4	98.44%(3)	100%(3)	95.68%(3)
Pima -Indian Diabetes	72.51%	8	73.18%(4)	76.04%(4)	71.89%(4)
Abalone	49.07%	8	52.00%(3)	58.00%(3)	49.26%(3)
Dermatology	95.08%	34	98.83%(26)	98.85%(26)	94.15%(26)
Heart	76.67%	13	80.00%(6)	81.11%(6)	79.00%(6)
German	68.50%	20	73.50%(6)	71.50%(6)	69.00%(6)
Lung cancer	59.67%	56	75.00%(21)	83.33%(21)	72.00%(21)
Soybean	85.00%	35	100.00%(22)	100.00%(22)	98.28%(22)
Spambase	93.26%	57	96.00%(26)	92.00%(26)	91.76%(26)
Glass Identification	62.00%	10	63.13%(5)	66.67%(5)	65.00%(5)
Teaching Assistant	54.92%	5	58.03%(2)	61.86%(2)	62.00%(2)
Contact Lens	76.00%	4	80.00%(2)	100.00%(2)	85.00%(2)
Sonar	69.50%	60	76.86%(7)	62.98%(7)	67.00%(7)
Statlog (Australian)	65.45%	14	74.30%(7)	79.04%(7)	75.24%(7)
Ionosphere	84.00%	34	88.00%(5)	90.62%(5)	90.10%(5)
Image Segmentation	85.00%	19	90.95%(14)	88.57%(14)	85.10%(14)

It can be seen that the classification accuracies using SVM, CART, and Naïve Bayes significantly improved from 7% to 15% after applying the proposed algorithm with feature subsets for the Wine, Breast Cancer, Statlog (Australian), Soybean, and Ionosphere datasets. We also note that Naïve Bayes yielded lower classification accuracy than SVM or CART.

In 97.70% of the cases, the proposed technique improved classification effectiveness and greatly reduced the number of features selected, thus increasing classification efficiency, for all of the classification methods. We

actually achieved 100.00% selection accuracy in four datasets with the proposed method. Regarding the classification methods, SVM yielded the highest classification accuracy in 65% of the datasets, while CART gave the highest accuracy in 35% of the datasets.

As shown in table 3, the proposed algorithm based on SVM and CART outperformed the others for 8 out of 12 datasets and 7 out of 12 datasets, respectively. The SVM classifier achieved better results with the Wine, Soybean and Iris datasets by 1.73%, 2.15% and 18.75%, respectively, compared with recent research on feature selection by Yang, *et al.* [27], and a 2.6% improvement with the Iris dataset compared with Gupta's study [28].

Table 3 Comparison on classification accuracy with other recently reported methods on common datasets (%).

Dataset	Proposd method with CART	Proposed method with SVM	[29]	[30]	[31]	[27]	[28]	[32]	[33]
Breast Cancer (original)	**97.80**	**97.90**	-	97.40	94.40	96.50	-	-	94.80
Breast Cancer (WDBC)	**95.50**	**96.10**	95.40	-	-	-	-	-	93.00
Iris	**98.40**	**100.00**	97.30		-	97.30	96.70	96.60	-
Pima Indian Diabetes	73.20	76.00	73.80	79.90	76.00	73.20	-	-	-
German	73.50	71.50	72.60	76.20	-	74.50	-	69.90	-
Soybean	**100.00**	**100.00**	-	88.30	-	97.80	-	-	-
Wine	**100.00**	**100.00**	-	-	91.60	98.30	-	-	-
Heart	80.00	81.10	-	-	61.10	84.80	87.10	-	-
Sonar	76.80	62.90	-	-	83.70	-	-	-	-
Abalone	52.00	**58.00**	54.50	-	-	-	30.00	25.70	-
Dermatology	**98.80**	**98.9**	-	-	-	95.40	-	-	-
Contact Lenses	**76.00**	**100.00**	-	-	-	-	75.00	-	-

Not only did the proposed algorithm reduce features from 13 to 7, 35 to 22, and 34 to 5 for the Wine, Soybean, and Ionosphere datasets, respectively, but also the classification accuracies improved by 12.35%, 17.64%, and 7.14% when compared with the accuracy using full datasets. With the Soybean dataset, the proposed algorithm reduced the number of features from 35 to 22 and the classification accuracy using SVM was 100.00%, which is much higher compared to the others methods. Moreover, the proposed algorithm also reduced the number of features from 8 to 3 and 4 to 2 with the Abalone and

Contact Lens datasets, respectively, and accuracy was again higher compared to the other methods.

The proposed algorithm based on a feature selection algorithm produced effective and small feature sets with higher classification accuracy on several different datasets because of the feature improvement step using a genetic algorithm that replaced the weakest features. The algorithm performed a more thorough search with a better chance of finding the optimal solution. Our proposed algorithm was able to extract a more relevant and effective feature set from the original feature set by employing the genetic operations of selection, crossover, and mutation to discover efficient and effective feature subsets.

5 Conclusions

Feature selection is critical to the performance of classification. In this paper, we proposed a feature selection algorithm that improves the performance of SFFS by incorporating a feature improvement step based on a genetic algorithm. This step helps discover important subsets that are not possible using SFFS alone. The algorithm employs mutual information as the feature subset evaluation function. The proposed technique was evaluated using 20 standard datasets from the UCI repository using three different classification methods. The results show that the proposed feature selection technique significantly improved classification accuracy and gave a much smaller feature set, thus improving efficiency. In addition, it performed very well in comparison with previously reported methods.

References

[1] Wu, X. & Zhu, X., *Mining with Noise Knowledge: Error-aware Data Mining*, IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, **38**(4), pp. 917-932, 2008.

[2] Sáez, José A., Luengo, J. & Herrera, F., *Evaluating the Classifier Behavior with Noisy Data Considering Performance and Robustness: the Equalized Loss of Accuracy Measure*, Neurocomputing, **176,** pp. 26-35, 2016.

[3] Sáez, José A., Galar, M., Luengo, J. & Herrera, F. *Analyzing the Presence of Noise in Multi-class Problems: Alleviating Its Influence with the One-vs-One Decomposition*, Knowledge and Information systems, **38**(1), pp. 179-206, 2014.

[4] Dash, M. & Liu, H., *Feature Selection for Classification, Intelligent Data Analysis*, **1**(1-4), pp. 131–156, Mar. 1997.

[5] Bins, J. & Draper, B., *Feature Selection from Huge Feature Sets, Proceedings on Eighth IEEE International Conference on Computer Vision*, pp. 159-165, 2001.

[6] Kira, K. & Rendell, L., *The Feature Selection Problem: Traditional Methods and a New Algorithm*, AAAI-92 Proceedings, pp. 129-134, 1992.

[7] MacQueen, J., *Some Methods for Classification and Analysis of Multivariate Observations*, Proceedings of the Fifth Berkley Symposium on Mathematics, Statistics and Probability, pp. 281-297, 1967.

[8] Somol, P., Pudil, P., Novovicova, J. & Pacliik, P., *Flexible Hybrid Sequential Floating Search in Statistical Feature Selection, Structural*, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science, Vol. 4109, Fred, A., Caelli, T., Duin, R.P.W., Campilho, A., Ridder, D., eds., pp. 632-639, Springer, 2006.

[9] Guyon, I. & Elisseeff, A., *An Introduction to Variable and Feature Selection, Journal of Machine Learning Research*, **3**, pp. 1157-1182, 2003.

[10] Maroño, N.S., Betanzos, A. & Castillo, E., *A New Wrapper Method for Feature Subset Selection, Proceedings European Symposium on Artificial Neural Networks*, pp. 515-520, 2005.

[11] Karegowda, A.G., Jayaram, M.A. & Manjunath, A.S., *Feature Subset Selection using Cascaded GA & CFS :A Filter Approach in Supervised Learning*, International Journal of Computer Applications, **23**(2), pp. 1-10, Jun. 2011.

[12] Zhang, H. & Sun, G., *Feature Selection using Tabu Search Method*, Pattern Recognition, **35**, pp. 701-711, 2002.

[13] Pudil, P. Novovicova, J. & Kittler, J., *Floating Search Methods in Feature Selection*, Pattern Recognition Letters, **15**, pp. 1119-1125, 1994.

[14] Nakariyakul, S. & Casasent, D.P. *An Improvement on Floating Search Algorithms for Feature Subset Selection*, Pattern Recognition, **41**(9), pp. 1932-1940, Sep 2009.

[15] Chaiyakarn, J. & Sornil, O., *A Two-Stage Automatic Feature Selection for Classification*, International Journal of Advancements in Computing Technology, **5**(14), pp. 168-179, Oct .2013 .

[16] De Maesschalck, Roy, Jouan-Rimbaud, Delphine. & Massart, D.L., *The Mahalanobis Distance*., Chemometrics and Intelligent Laboratory Systems, **50**, (1), pp. 1-18, 2000.

[17] Bruzzone, L. & Serpico, S.B., *A Technique for Feature Selection in Multiclass Problems*, International Journal of Remote Sensing, **21**(3), pp. 549-563, 2000.

[18] Battiti, R., *Using Mutual Information for Selecting Features in Supervised Neural Net Learning*, IEEE Transactions on Neural Networks, **5**(4), pp. 537-550, 1994.

[19] Goldberg, D., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, 1989.

[20] Brill, F., Brown, D. & Martin, W., *Fast Genetic Selection of Features for Neural Networks Classifiers*, IEEE Transactions: Neural Networks, **3**(2), pp. 324–328, March 1992.

[21] Cortes, C. & Vapnik, V., *Support-Vector Networks*, Machine Learning, **20**(3), pp. 273-297, 1995.

[22] Huang, C.-L. & Wang, C.-J., *A GA-based Feature Selection and Parameters Optimization for Support Vector Machines.*, Expert Systems with Applications, **31**(2), pp. 231-240, 2000.

[23] Tsai, C-J., Lee, C-I. & Yang, W-P., *A Discretization Algorithm based on Class-attribute Contingency Coefficient.*, Information Science, **178**, pp. 714-731, 2008.

[24] Brieman, L., Friedman, J., Olshen, R. & Stone, C., *Classification of Regression Trees*, Wadsworth Inc., 1984.

[25] Oh, I.S., Lee, J.S. & Moon, B.R., *Hybrid Genetic Algorithms for Feature Selection*, IEEE Transactions: Pattern Analysis and Machine Intelligence, **26**(11), pp. 1424–1437, Nov .2004.

[26] Asuncion, A. & Newman, D.J., *UCI Machine Learning Repository*, University of California, Department of Information and Computer Science, http//:www.ics.uci.edu/~mlearn/MLRepository.html (2007).

[27] Yang, T., Cao, L. & Zhang, C., *A Novel Prototype Reduction Method for the K-nearest Neighbor Algorithm with K ≤1*, Pacific-Asia Conference on Knowledge Discovery and Data Mining – Volume part II, pp. 89-100, 2010.

[28] Gupta, A., *Classification of Complex UCI Datasets using Machine Learning and Evolutionary Algorithms*, International Journal of Scientific & Technology Research, **4**(5), pp. 85-94, May 2015.

[29] Liu, H., Motoda, H. & Yu, L., *A Selective Sampling Approach to Active Feature Selection*, Artificial Intelligence, 159(1), pp. 49-74, Nov. 2004.

[30] Ratanamahatana, C. & Gunopulos, D., *Scaling up the Naïve Bayesian Classifier using Decision Trees for Feature Selection*, Applied Artificial Intelligence, **17**(5-6), pp. 475-487, 2003.

[31] Anwar, H., Qamar, U., & Qureshi, A.W.M., *Global Optimization Ensemble Model for Classification Methods*, The Scientific World Journal, **2014**, pp. 1-9, Apr. 2014.

[32] Tsai, C-F., Lin, W-Y., Hong, Z-F. & Hsieh, C-Y., *Distance-based Features in Pattern Classification*, EURASIP Journal on Advances in Signal Processing, **2011**(62), pp. 2-11, Dec. 2011.

[33] Lavanya, D., & Usha Rani, K., *Analysis of Feature Selection with Classification: Breast Cancer Datasets*, Indian Journal of Computer Science and Engineering, **2**(5), pp. 756-763, 2011.