Knowledge-Based Systems 110 (2016) 191–201

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Binary coordinate ascent: An efficient optimization technique for feature subset selection for machine learning

Amin Zarshenas*, Kenji Suzuki

Medical Imaging Research Center & Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA

ARTICLE INFO

Article history:
Received 27 February 2016
Revised 12 July 2016
Accepted 18 July 2016
Available online 19 July 2016

Keywords:
Machine learning
Classification
Feature selection
Wrapper
Optimization
Heuristic

ABSTRACT

Feature subset selection (FSS) has been an active area of research in machine learning. A number of techniques have been developed for selecting an optimal or sub-optimal subset of features, because it is a major factor to determine the performance of a machine-learning technique. In this paper, we propose and develop a novel optimization technique, namely, a binary coordinate ascent (BCA) algorithm that is an iterative deterministic local optimization that can be coupled with wrapper or filter FSS. The algorithm searches throughout the space of binary coded input variables by iteratively optimizing the objective function in each dimension at a time. We investigated our BCA approach in wrapper-based FSS under area under the receiver-operating-characteristic (ROC) curve (AUC) criterion for the best subset of features in classification. We evaluated our BCA-based FSS in optimization of features for support vector machine, multilayer perceptron, and Naïve Bayes classifiers with 12 datasets. Our experimental datasets are distinct in terms of the number of attributes (ranging from 18 to 11,340), and the number of classes (binary or multi-class classification). The efficiency in terms of the number of subset evaluations was improved substantially (by factors of 5–37) compared with two popular FSS meta-heuristics, i.e., sequential forward selection (SFS) and sequential floating forward selection (SFFS), while the classification performance for unseen data was maintained.

1. Introduction

Feature subset selection (FSS) in classification and regression has been an active area of research in machine learning. FSS plays an important role in machine learning and data mining, because it is a major factor to determine the performance of a machine-learning technique [1]. Because of its importance, a number of investigators have studied and developed various techniques. The goal of an FSS technique is to select an optimal or sub-optimal subset of features that makes a machine-learning technique the highest performance for a specific task (e.g., classification) [2]. A number of surveys have been published in literature to review the efficacy and efficiency of FSS techniques in different machine-learning tasks [1,3–6]. FSS for classification, which is the main focus of this paper, was surveyed in detail in [4,6]. A large number of investigators have applied FSS methods in classification tasks in their applications to improve the performance of their systems [7–11].

FSS techniques can be generally categorized into filter, wrapper, hybrid, and embedded methods [12,13]. Filter methods use a measurement to assign meaningful scores to different combinations of features, i.e., subsets of features, without using the knowledge on the employed classifier [14–17]. Correlation criteria [18] and mutual information (MI) [19] are two of the most popular measurements used in this type of technique, in order to determine the usefulness of feature subsets. On the other hand, wrapper methods utilize an algorithmic-dependent measurement to examine the usefulness of feature subsets in an iterative manner [20]. Hybrid methods exploit both filter and wrapper methods in a single context in a way to boost up the FSS algorithm [21]. For high dimensional data, a filter FSS would be often followed by a wrapper FSS in a hybrid FSS framework [22,23]. Incremental wrapper-based subset selection (IWSS) and its recently improved version, IWSS with replacement (IWSSr) belong to this FSS category [24]. Unlike the mentioned categories, embedded techniques such as decision trees [25] and L1-support-vector-mahcine (SVM) [26] accomplish their goal by directly including FSS as a part of the optimization objective of the learning algorithm.

The wrappers have been used widely for classification because they often obtained superior performance as they find feature subsets better suited with a pre-determined classifier [6]. In general, there are three main factors that describe a wrapper procedure: a type of the classifier, a feature subset evaluation criterion,

* Correspondence author.
E-mail address: mzarshen@hawk.iit.edu (A. Zarshenas).

http://dx.doi.org/10.1016/j.knosys.2016.07.026
0950-7051/

and an optimization technique to find the best combination of features. Several evaluation criteria such as accuracy [27,28], a false-positive (FP) elimination rate [29,30], and area under the receiver-operating-characteristic (ROC) curve (AUC) [31–35] have been used widely. Meta-heuristics including sequential and randomized techniques have been studied extensively and utilized in order to find the optimum or a sub-optimum subset of features. Genetic algorithms (GA) [29,36], particle swarm optimization (PSO) [37–39], and simulated annealing (SA) [27,40,41] are few examples of randomized search strategies used widely in FSS. Despite progress made through a randomized graph-based optimization approach, they are not fully satisfactory as they either yield a solution away from the optimum or they are computationally impractical [42]. There exist popular and commonly used deterministic search strategies, designed specifically for the task of FSS, as alternatives that try to overcome the above mentioned problems, such as sequential forward selection (SFS) [2,4,43,44] and sequential forward floating selection (SFFS) [28,42,45].

In this paper, we propose and develop a novel optimization technique, namely, a binary coordinate ascent (BCA) algorithm, inspired by the popular coordinate descent algorithm [46,47], for efficiently solving combinatorial optimization problems. Our BCA optimization algorithm, as a deterministic local optimization approach, starts its search from an initial point in the space with binary representation of input variables and continuous output values. It iteratively updates the solution by optimizing the given cost function at each individual coordinate, one at a time. In this study, we investigated our BCA approach in a wrapper-based FSS framework for the task of classification in order to reduce a huge number of subset evaluations needed in earlier existing FSS techniques. We use a binary representation of feature subsets to set up the requirements for our BCA algorithm in FSS for classification. In order to find the best (i.e., optimum or sub-optimum) subset of features, AUC of a classifier which quantifies the generalization performance of a classification system, obtained in a 10-fold cross validation (CV) manner, is set as the evaluation criterion to determine the usefulness of different subsets. To examine the consistency of our wrapper-based BCA approach, we investigated the efficiency and performance of our proposed method coupled with SVMs [48], multilayer perceptron (MLP) [49], and naïve Bayes (NB) [50] classifiers. In order to reduce the risk of overfitting while performing assessment of FSS methods [51], an independent test set was used for comparisons of our proposed algorithm with two of the most popular FSS techniques, i.e., the SFS and SFFS. Additionally, we investigated the characteristics of our BCA algorithm when comparing it with the IWSSr as a filter-wrapper approach, under different scenarios. To our knowledge, no investigator has proposed the BCA algorithm or developed an FSS algorithm based on the BCA algorithm before. This paper is organized as follows. The proposed BCA optimization algorithm, binary representation formulation of the FSS, and the proposed FSS algorithms based on BCA are described in Section 2. Experimental results and comparisons are explained in Section 3. Thorough discussions of the proposed algorithms are made in Section 4; and the paper is concluded in Section 5.

2. Methods

2.1. Coordinate descent algorithm

Coordinate descent (CD) algorithm [47] is one of the most popular non-derivative optimization techniques that have been used widely to solve a variety of optimization problems including quadratic programming. It has been employed in the machine-learning community for different tasks such as for training SVMs [46]. Convergence of the CD algorithm to the global optimum is guaranteed under specific assumptions such as convexity of the objective function and its smoothness, but such assumptions cannot always be made in real applications. Without following all of those assumptions, CD is capable of finding a local optimum. The CD algorithm can be formulated as follows:

$$X_i^{k+1} = \underset{a \in R}{\operatorname{argmin}} J\left(X_1^{k+1}, \ldots, X_{i-1}^{k+1}, a, X_{i+1}^{k}, \ldots, X_N^{k}\right) \quad (1)$$

where N is the number of dimensions of the input variables, $J(.): R^N \rightarrow R$ is the objective function, and X_i^k is the i^{th} variable of the solution at the k^{th} iteration. Without loss of generality, we can formulate the coordinate ascent (CA) algorithm by considering the maximization of the negative of the objective function J in Eq. (1). CD is motivated by the hypothesis that one can find the local optimum of an objective function by minimizing it along each coordinate, one at a time. If optimality assumptions are not followed, CD results are subject to the choice of initialization as well as other local search methods.

2.2. Binary coordinate ascent algorithm

In a variety of programming problems, a binary representation of input variables is possible, and it can be used to encode the input space [52]. FSS is one of the problems that can be represented in a binary variable framework, which was the main motivation for the proposed method. The general form of an unconstrained programming problem of maximizing a continuous objective function in a binary variable framework is given by:

$$X^* = \underset{X \in B^N}{\operatorname{argmax}} J(X), \; B = \{0, 1\}. \quad (2)$$

The problem in Eq. (2) is a special case of integer programming; therefore, an exhaustive search can possibly be performed if the number of variables is not too large, but the problem is considered as one of the NP-hard problems in the computational complexity theory [1]. As a result, a variety of suboptimal solution techniques such as evolutionary-algorithm-based optimizations [53] and swarm-based algorithms [54] have been used widely to deal with these problems. However, most of these either stochastic or deterministic heuristics are computationally expensive; thus, they require minor or major modifications for the task of FSS. Motivated by the given facts, we propose a new version of the CA algorithm, suitable for solving these types of problems in terms of efficiency and efficacy, namely, a binary coordinate ascent (BCA) algorithm. Unlike the CA algorithm that requires defining a search line optimization approach through single coordinates, the BCA employs a zero-one switch strategy to find the local optimum along each coordinate, one at a time. The detailed pseudocode of our BCA optimization algorithm is given in Table 1.

In the given algorithm in Table 1, the BCA starts its search from an initial point, i.e., the binary vector of all zeroes at the origin. We discuss the choice of the initial point that depends on the application requirements in a later section. The BCA then updates the solution to the optimization problem through maximizing the objective function at each coordinate individually. A zero-one switching strategy is used to find the maximum along a single coordinate. Once the BCA completes the search through all coordinates, here referred to as one BCA scan, the solution (i.e., maximum) $Y*$ is updated. The algorithm stops when there is no more significant change in the optimum objective function value. The total number of iterations then can be calculated as $S \times N$ where S is the total number of BCA scans, and N is the dimension of solution space. Other stopping criteria such as the number of iterations or the number of BCA scans S could be employed, as the BCA performs in an any-time programming algorithm manner. Eventually, the algorithm returns a pair of $Y*$ and $X*$ as the final maximum and argument of the maximum, respectively. Although the BCA is designed explicitly for the task of FSS, the algorithm can be performed in

Table 1
Pseudocode of our BCA optimization algorithm for combinatorial optimization in a binary representation framework.

BCA optimization algorithm

Initialization:
Set $X* = (0, 0, \ldots, 0)^T$, where $X* \in \{0, 1\}^N$ // *vector of all zeroes as the initial solution*
Set $Y^* = J(X^*)$ // *initial value of the objective function to be maximized*
Set Δ // Δ *and Stop variables are defined for convergence criterion*
$Stop = 0$
while ($Stop = 0$) // *convergence criterion is examined after each BCA scan*
  for ($i = 1 : N$) // *each complete loop (N iterations) is one BCA scan of all coordinates*
    $X = X^*$
    $X_i = not(X_i^*)$ // $not(0) = 1$ *and* $not(1) = 0$
    if ($J(X) > J(X*)$)
      $X^* = X$
    end
  end
  $Y = J(X*)$ // *objective function value for the current solution*
  if ($|Y^* - Y| < \Delta$) // *stopping criterion,* i.e., *convergence of the objective function value*
    $Stop = 1$
  end
  $Y^* = Y$
end
Output:
Final solution ($X*$, $Y*$)

any programming problem with a binary representation of input variables and continuous objective function values, given the fact that there is no initial assumption about the objective function $J(.)$ in the methodology.

2.3. BCA-based wrapper FSS

Based on our BCA optimization approach, we propose a fast wrapper FSS algorithm with two variants of initialization strategies: one starting with an empty set of features, namely, BCA based FSS with zero initialization (BCA-Zero); and the other with a feature-rating-based initialization scheme, namely, BCA-based FSS with initialization (BCA-Initialized). In order to set up the FSS problem to be solved with the BCA optimization algorithm, we first represent feature subsets in a binary fashion. In this framework, each feature subset is represented as a binary vector of length N, where N is the total number of initial input features (i.e., the number of attributes), and each element corresponds to one of the input features. An element of this vector is one only if the corresponding feature to that element is included in the feature subset corresponding to that binary vector. In this scenario, we can represent the FSS problem using Eq. (2) having X and $J(.)$ be the binary-encoded feature subsets and the performance of a pre-determined classifier, respectively. An example of the state diagram to visualize FSS in a binary fashion for the case of N=4 is given in Fig. 1. In the figure, nodes represent feature subsets (the total of 2^N solutions exist), while edges connect feature subsets with a Hamming distance of one, i.e., a shift between these subsets can be made by adding or removing only one feature.

The block diagram of our proposed wrapper FSS algorithm is shown in Fig. 2. In the BCA-Zero approach, our FSS algorithm starts its search from an empty set of features, i.e., a vector of all zeroes in a binary representation fashion. The BCA algorithm iteratively adds and removes features to and from the selected subset of features based on the objective function values. In fact, at each iteration, the BCA ensures whether existence of a feature, in a given subset of features, improves or drops the classification performance. If a feature was included in or removed from the feature subset accidently, BCA algorithm is freely capable of correcting the wrong decisions through the proceeding BCA scans, in order to approximate the optimal solution as much as possible. In our implementation, AUC, which approximates the generalization performance of a learning algorithm properly by estimating the probability that a classifier ranks a randomly chosen positive instance higher than a randomly chosen negative instance [55], is used as the objective function $J(.)$. Indeed, the objective function is the validation performance of a pre-determined classifier obtained by 10-fold CV on a validation set. Once the algorithm converges to the final solution, i.e., the best subset of features, a classifier is trained and tested on the entire validation set and an independent test

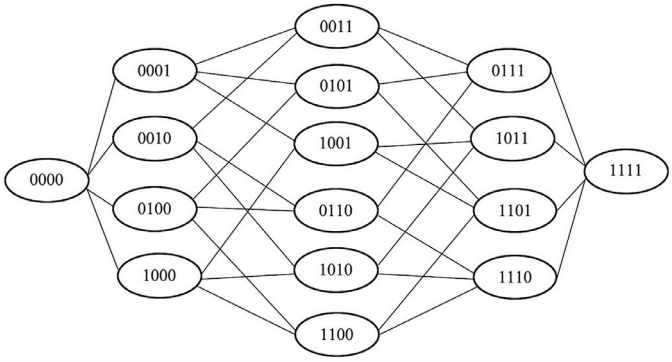

Fig. 1. State diagram for visualizing FSS in a binary representation framework. In this example, the diagram shows all combinations of feature subsets in a four-dimensional binary space, i.e., the total number of features is four.

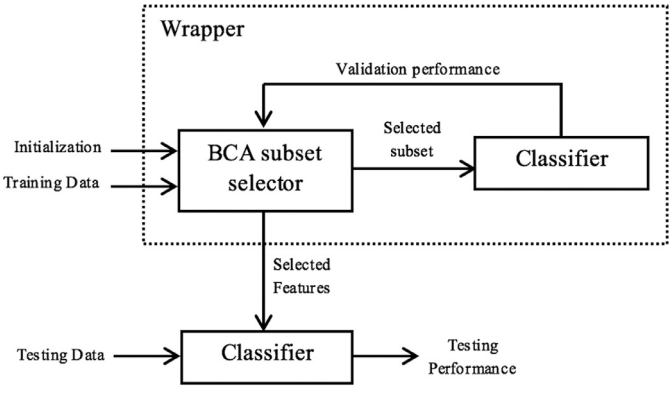

Fig. 2. Block diagram of our proposed wrapper FSS algorithm employing the BCA optimization algorithm.

Table 2
Description of the datasets used in our comparative evaluation experiments.

Data set	Number of instances	Number of Attributes	Number of Classes
Polyp	2670	79	2
Sonar	208	60	2
Landsat	6435	36	6
Parkinson	195	22	2
Breast Cancer	569	30	2
Segmentation	2310	19	7
Climate	540	18	2
Average	1846.7	37.7	

set, respectively, in order to evaluate the algorithm. To examine the consistency in the efficiency and performance of our wrapper approach with the choice of the classifier, in our implementations, we employed three well-known classifiers, i.e., SVM, MLP, and NB, which are distinct in their strategies to learn a model. Regarding the second scheme, i.e., the BCA-Initialized, the FSS procedure and all corresponding implementation details are exactly the same as that of the BCA-Zero, except that in this case, a feature-rating-based strategy is utilized to select the initial subset of features, instead of starting from the empty set of features. To do so, we use a ranking scheme of features based on a classifier-dependent score calculated for each feature individually. In other words, for each feature independently, we calculate the AUC (i.e., the objective function in our scheme) of the pre-determined classifier in a 10-fold CV fashion, as its individual score in the absence of other features. Once the features are ranked based on this score, the first T% of the best features (i.e., the highest AUCs) are selected in the initial subset of features. T is a variable set by the user and needs a prior knowledge of the dataset such as the total number of features. In our experiments, T is set to 20%, based on the number of attributes of the datasets that we experiment on.

3. Experimental results

In order to show the efficiency and performance of our proposed BCA FSS algorithm, we compared our algorithm with two of the most popular wrapper-based FSS techniques, i.e., SFS [43] and SFFS [42]. We also compared our proposed algorithm with a relatively new efficient filter-wrapper FSS technique, i.e., IWSSr [24]. We did the comparative evaluation in terms of the classification performance, the number of selected features, the number of subset evaluations, and the processing time. To have a rational comparison, for this experiment we used seven independent datasets with fairly large numbers of instances (average of 1846.7). Six out of seven datasets, which are either binary or multi-class classification tasks, are publicly available and obtained from the UCI machine-learning repository [56]. The seventh dataset, namely Polyp, is a medical database containing 2670 polyp candidates for one of each 79 morphology, intensity, and shape-based features, which were extracted from CT colonography (CTC) images. The number of dataset attributes ranges from 18 to 79 excluding the class attribute. The full description of the datasets is summarized in Table 2.

In order to examine the consistency of the performance of the proposed algorithm with the choice of the pre-determined classifier in a wrapper FSS framework, we coupled both the proposed algorithms and the reference techniques with three different classifiers, namely, the SVM, MLP, and NB. These classifiers were selected because they are distinct in terms of their nature of learning of a model. LIBSVM [57] and WEKA [58] packages were utilized for implementation of these learning algorithms in our experiments. All classifier parameters in our implementations were set to default values of these libraries. In order to reduce the risk of overfitting while performing FSS [51], we employed independent test sets for evaluation of each approach. We extracted approximately 70% of samples from each dataset as a training set, randomly and in a stratified manner, while we kept the rest for testing purpose. In other words, each FSS method was executed on the training data using a 10-fold CV strategy. Once the best subset of features (i.e., the best validation performance) was obtained, a model was then trained on the entire training set, using only selected features, and tested on the independent test set.

The performance results represented by AUC values as well as the number of selected features obtained for the proposed algorithms and the two reference methods, using three classifiers on seven datasets, are shown in Table 3. We performed statistical analysis on the results obtained for each classifier separately. In this study, we followed the statistical analysis methodology recommended in [59–61]. Namely, Friedman aligned ranks test was utilized to statistically compare the results obtained for all five approaches for each classifier, separately. If a significant difference with a confidence level of 0.05 was detected, we proceeded to the Holm post hoc analysis of experimental approaches using the BCA-Zero as the control algorithm. We use the notations NS and S under the results of each method, when there was no significant or a significant difference between the corresponding method and the BCA-Zero as the control algorithm, respectively. Regarding the AUC values, our experiments showed that none of the three statistical tests obtained significance; thus, we did not proceed to pairwise comparisons. Therefore, the performance, in terms of AUC values, of the proposed methods is considered comparable to that of the SFS, SFFS, and IWSSr techniques. The average rank of each approach obtained by the Friedman ranking method is also depicted in Table 3 for a better understanding of the comparisons. Note that the best AUC and the best number of selected features should result in ranks of 1 and 5, respectively. Moreover, one can see that there is no substantial difference between the two variants of the proposed algorithm in terms of the classification performance. The same comparison and statistical analysis methodology was also applied for the number of selected features obtained for each method. While most of the results were statistically comparable in terms of the number of selected features, IWSSr was capable of selecting a smaller subset of features. In fact, IWSSr was specifically designed to select compact subsets of features through a filter-wrapper approach and a novel feature replacement strategy [24]. Moreover, one might notice that the BCA-Initialized found a smaller subset of features on average, compared with that of the BCA-Zero, although the difference was not statistically significant. This result makes sense because the BCA-Initialized employs the more sophisticated initialization strategy.

Table 4 shows the number of feature subset evaluations (i.e., the number of trials before selecting the best feature subset) and the algorithm running time, for both our proposed algorithms and the reference FSS techniques. In order not to manipulate the reference methods, we used the exact implementation of the IWSSr technique [24], which uses a 5-fold CV (unlike the 10-fold CV in our implementations) to estimate the usefulness of feature subset candidates; therefore, we did not report the timing results for the IWSSr. All timing results were obtained on a workstation running the Ubuntu Linux 14.04 operating system with Intel Core i7-4790 K @ 4.0 GHz CPU and 16 GB RAM. Results show that the number of subset evaluations was reduced by the factors of approximately 2, 7 and 37 on average, when comparing our BCA-Initialized to the IWSSr, SFS and SFFS techniques, respectively. Comparing our BCA-Zero with the IWSSr, SFS and SFFS techniques, the number of subset evaluations was reduced approximately by factors of 1.5, 5 and 28 on average, respectively. Looking at Table 4, one can see that the processing speed was also improved by the proposed algorithms, which is consistent with our analysis of the number of subset

Table 3
Comparisons of the classification performance (in terms of AUC) as well as the number of selected features (#Feats) of our BCA algorithms with zero and feature-rating-based initialization schemes with that of the SFS, SFFS, and IWSSr FSS techniques for seven datasets using SVM, MLP, and NB classifiers. AUC results on independent (unseen) test data as well as the number of selected features (#Feats) are reported.

Dataset	BCA-Zero		BCA-Initialized		SFS		SFFS		IWSSr	
	AUC	#Feats	AUC	#Feats	AUC	#Feats	AUC	#Feats	AUC	#Feats
SVM										
Polyp	0.964	18	0.966	35	0.945	16	0.934	17	0.876	2
Sonar	0.779	17	0.743	17	0.756	9	0.744	10	0.798	6
Landsat	0.976	17	0.976	16	0.975	14	0.976	15	0.975	12
Parkinson	0.855	6	0.855	6	0.855	6	0.855	6	0.827	2
Breast Cancer	0.997	15	0.998	10	0.997	12	0.997	12	0.997	6
Segmentation	0.988	8	0.990	8	0.989	6	0.989	6	0.987	6
Climate	0.974	10	0.978	9	0.973	12	0.974	10	0.976	7
Mean ± SD	**0.933 ± 0.08**	**13 ± 4.9**	**0.929 ± 0.09**	**14.4 ± 9.9**	**0.927 ± 0.09**	**10.7 ± 3.8**	**0.924 ± 0.09**	**10.9 ± 4.2**	**0.919 ± 0.08**	**5.9 ± 3.4**
Average rank	**2.79**	**1.71**	**1.93**	**2.36**	**3.43**	**3.14**	**3.14**	**2.93**	**3.71**	**4.86**
			NS	**NS**	**NS**	**NS**	**NS**	**NS**	**NS**	**S**
MLP										
Polyp	0.802	21	0.896	21	0.899	46	0.783	23	0.974	7
Sonar	0.880	18	0.897	16	0.797	8	0.839	17	0.798	6
Landsat	0.973	17	0.977	23	0.976	36	0.969	28	0.964	7
Parkinson	0.767	4	0.705	8	0.796	12	0.776	12	0.776	3
Breast Cancer	0.999	12	0.998	9	0.997	5	0.998	18	0.994	7
Segmentation	0.992	15	0.982	11	0.989	14	0.994	18	0.989	5
Climate	0.973	11	0.962	5	0.959	14	0.953	10	0.962	5
Mean ± SD	**0.912 ± 0.10**	**14 ± 5.6**	**0.917 ± 0.10**	**13.3 ± 6.8**	**0.916 ± 0.09**	**19.3 ± 15.5**	**0.902 ± 0.10**	**18 ± 6.1**	**0.922 ± 0.09**	**5.7 ± 1.5**
Average rank	**2.43**	**2.64**	**2.86**	**3.43**	**3.07**	**2.36**	**3.29**	**1.79**	**3.36**	**4.79**
			NS	**NS**	**NS**	**NS**	**NS**	**NS**	**NS**	**NS**
NB										
Polyp	0.964	22	0.973	26	0.956	20	0.973	26	0.886	14
Sonar	0.802	16	0.734	11	0.664	14	0.783	8	0.726	7
Landsat	0.958	13	0.959	13	0.959	13	0.958	13	0.957	9
Parkinson	0.680	6	0.672	6	0.680	6	0.680	6	0.832	4
Breast Cancer	0.998	10	0.999	9	0.998	12	0.998	9	0.995	8
Segmentation	0.982	8	0.979	8	0.982	8	0.982	8	0.978	7
Climate	0.978	10	0.978	10	0.977	10	0.977	10	0.980	5
Mean ± SD	**0.909 ± 0.12**	**12.1 ± 5.4**	**0.899 ± 0.14**	**11.9 ± 6.6**	**0.888 ± 0.15**	**11.9 ± 4.6**	**0.907 ± 0.12**	**11.4 ± 6.8**	**0.908 ± 0.09**	**7.7 ± 3.3**
Average rank	**2.57**	**2.29**	**2.64**	**2.57**	**3.29**	**2.43**	**2.79**	**2.71**	**3.71**	**5**
			NS	**NS**	**NS**	**NS**	**NS**	**NS**	**NS**	**S**

Note: when considering the results for Holm post hoc statistical analysis, NS and S under the results of a method indicate that there is no significant or a significant difference between the corresponding method and the BCA-Zero as the control algorithm, respectively.

evaluations. Similar statistical analysis methodology discussed earlier was used to examine the significance for the timing and the number of subset evaluation results. In this case, all p-values obtained by the Friedman aligned ranks tests were less than 0.05, yielding to reject the null hypothesis in favor of the alternative hypothesis, i.e., there exist some significant differences in the obtained results. Therefore, we proceeded to the Holm post hoc analysis. The statistical analysis showed that the differences between the number of subset evaluation results obtained by BCA-Zero were significant, with a significance level of 0.1, with respect to those of obtained by either of the SFS or SFFS techniques. The difference between the BCA-Zero and IWSSr was not statistically significant. This will be further discussed in detail with considering the impact of feature ordering and having higher dimensional datasets (more than 1000 features) in Section 4. In fact, we will see a clear difference between the computational complexity of the BCA and IWSSr. Moreover, we compared the two variants of our proposed approach, i.e., the BCA-Zero and BCA-Initialized. They were comparable in terms of timing results, whereas there was a slight improvement (although not statistically significant) in terms of the number of subset evaluations for the case of the BCA-Initialized. This was achieved by sacrificing some processing time for the algorithm initialization through the feature-rating-based initialization process. Note that, although we performed statistical analysis to confirm our comparisons, one might easily see from Table 4 that the SFS and SFFS techniques were always assigned the second last and the last ranks, respectively, by the Friedman ranking method in all experiments, where the higher the rank, the better the method is.

4. Discussion

In general, there is not a strong reason to believe that better approaches might overlap in selecting the best subset of features, without having the knowledge of the ground truth features. However, we believe it useful to analyze the selected features for the methods to gain insight into the characteristics of the methods. Table 5 shows the results of the pair-wise analysis of overlapping features for the four wrapper approaches in our experiments, i.e., the BCA-Zero, BCA-Initialized, SFS, and SFFS. The number of overlapping features alone may not necessarily provide an accurate similarity measure for this particular purpose, as it does not consider the length of each subset; therefore, we used the Jaccard similarity (JS) metric for feature analysis, i.e., the ratio of the size of the intersection of two sets divided by the size of their union. Looking at the average JS metrics depicted in Table 5, one might see that there is a better match between the SFS and SFFS techniques than between these two and the proposed algorithms, i.e., the SFS obtained similar features to those obtained by the SFFS. However, all JS values in pair-wise comparisons were comparable, indicating the pair-wise overlap of the selected features was comparable. This does not necessary show the effectiveness of any of these approaches. However, an interesting result can be depicted by considering the JS similarities obtained for the case of the MLP classifier and comparing them to those obtained for the SVM and NB classifiers. Most of the JS results obtained for the MLP were smaller than those of the SVM and NB classifiers. This difference can be understood by considering MLP classifiers as one of the learning algorithms with a built-in

Table 4
Comparisons of the computational complexity (in terms of the number of subset evaluations) as well as the timing results (in minutes) of our BCA algorithms with zero and feature-rating-based initialization schemes with that of the SFS, SFFS, and IWSSr FSS techniques for seven datasets using SVM, MLP, and NB classifiers.

Dataset	BCA-Zero		BCA-Initialized		SFS		SFFS		IWSSr
	Time	#subset	Time	#subset	Time	#subset	Time	#subset	#subset
SVM									
Polyp	11.9	237	26.9	237	282.3	3160	3336.8	22,939	234
Sonar	5.1	540	2.03	180	17.29	1830	110.53	9567	326
Landsat	63.6	108	89.4	108	338.7	666	1043.4	1810	322
Parkinson	0.7	88	0.64	66	1.93	253	6.43	792	57
Breast Cancer	1.6	90	1.17	60	7.59	465	32.15	1676	156
Segmentation	7.2	57	8.90	57	21.90	190	113.30	886	93
Climate	1.1	90	0.62	36	1.94	171	7.70	597	97
Mean ± SD	**13.0 ± 22.6**	**172.8 ± 171**	**18.5 ± 32.7**	**106.3 ± 74**	**95.9 ± 147.7**	**962.1 ± 1128**	**664.3 ± 1235.6**	**5466.7 ± 8330**	**183.6 ± 111**
Average rank	**4.43**	**3.79**	**3.57**	**4.5**	**2**	**2**	**1**	**1**	**3.71**
			NS	**NS**	**S**	**S**	**S**	**S**	**NS**
MLP									
Polyp	94.5	237	117.7	237	2808.7	3160	58,335.0	27,973	430
Sonar	9.3	300	4.01	120	87.70	1830	1531.6	16,094	326
Landsat	191.1	144	175.5	108	698.4	666	7606.6	4208	237
Parkinson	0.8	66	0.83	44	4.18	253	30.24	408	75
Breast Cancer	7.4	150	4.20	90	24.48	465	427.83	5364	169
Segmentation	28.8	57	28.90	76	59.32	190	416.13	1005	99
Climate	2.1	54	1.82	54	5.62	171	46.30	1042	85
Mean ± SD	**47.7 ± 71.3**	**144.0 ± 95**	**47.6 ± 70.3**	**104.1 ± 64**	**526.9 ± 1036.4**	**962.1 ± 1128**	**9770.5 ± 21,584**	**8013.4 ± 10,334**	**203 ± 135**
Average rank	**3.43**	**4.29**	**3.57**	**4.71**	**2**	**2**	**1**	**1**	**3**
			NS	**NS**	**S**	**S**	**S**	**S**	**NS**
NB									
Polyp	11.8	395	13.7	395	107.2	3160	728.4	15,409	687
Sonar	1.4	240	1.56	240	10.90	1830	64.00	9343	362
Landsat	10.0	180	8.1	144	28.6	666	94.3	1736	237
Parkinson	0.4	88	0.31	44	1.22	253	6.48	1158	62
Breast Cancer	0.9	120	0.84	90	3.58	465	16.34	1796	149
Segmentation	0.9	57	1.12	57	2.86	190	7.71	453	94
Climate	0.3	36	0.37	36	1.18	171	3.29	430	86
Mean ± SD	**3.7 ± 4.9**	**159.4 ± 125**	**3.7 ± 5.2**	**143.7 ± 131**	**22.2 ± 38.7**	**962.1 ± 1128**	**131.5 ± 265.5**	**4332.1 ± 5792**	**239.5 ± 234**
Average rank	**3.57**	**4.14**	**3.43**	**4.71**	**2**	**2**	**1**	**1**	**3.14**
			NS	**NS**	**S**	**S**	**S**	**S**	**NS**

Note: when considering the results for Holm post hoc statistical analysis, NS and S under the results of a method indicate that there is no significant or a significant difference between the corresponding method and the BCA-Zero as the control algorithm, respectively.

Table 5
Analysis of overlapping features between each pair of wrapper FSS algorithms (BCA-Zero, BCA-Initialized, SFS, and SFFS). Similarities of the selected features with respect to the feature subset length are compared in terms of the Jaccard similarity (JS).

Dataset	BCA-Zero/BCA-Initialized	BCA-Zero/SFS	BCA-Zero/SFFS	BCA-Initialized/SFS	BCA-Initialized/SFFS	SFS/SFFS
SVM						
Polyp	0.293	0.214	0.207	0.186	0.182	0.222
Sonar	0.478	0.444	0.588	0.368	0.500	0.357
Landsat	0.737	0.550	0.455	0.579	0.476	0.813
Parkinson	1.000	1.000	1.000	1.000	1.000	1.000
Breast Cancer	0.563	0.500	0.500	0.467	0.467	1.000
Segmentation	1.000	0.556	0.556	0.556	0.556	1.000
Climate	0.583	0.833	1.000	0.750	0.583	0.833
Mean ± SD	**0.665 ± 0.26**	**0.585 ± 0.26**	**0.615 ± 0.29**	**0.558 ± 0.26**	**0.621 ± 0.22**	**0.746 ± 0.32**
MLP						
Polyp	0.200	0.264	0.189	0.264	0.294	0.353
Sonar	0.214	0.130	0.207	0.263	0.320	0.250
Landsat	0.429	0.472	0.364	0.639	0.546	0.778
Parkinson	0.333	0.231	0.333	0.427	0.333	0.714
Breast Cancer	0.167	0.133	0.364	0.167	0.421	0.150
Segmentation	0.368	0.526	0.737	0.667	0.611	0.789
Climate	0.455	0.563	0.500	0.357	0.500	0.600
Mean ± SD	**0.309 ± 0.12**	**0.331 ± 0.19**	**0.385 ± 0.19**	**0.376 ± 0.19**	**0.398 ± 0.19**	**0.519 ± 0.26**
NB						
Polyp	0.778	0.615	0.778	0.582	1.000	0.582
Sonar	0.421	0.429	0.500	0.563	0.583	0.467
Landsat	0.733	0.625	0.857	0.857	0.857	0.733
Parkinson	0.714	0.714	0.714	0.714	0.714	1.000
Breast Cancer	0.727	0.692	0.583	0.500	0.636	0.500
Segmentation	0.778	1.000	1.000	0.778	0.778	1.000
Climate	1.000	0.818	0.818	0.818	0.818	1.000
Mean ± SD	**0.736 ± 0.17**	**0.699 ± 0.18**	**0.750 ± 0.17**	**0.687 ± 0.14**	**0.769 ± 0.14**	**0.755 ± 0.24**

Table 6
Detailed structure of the experimental FSS techniques studied in this paper.

	BCA-Zero	BCA-Initialized	SFS	SFFS	IWSSr
Initialization	Empty set	Feature-rating-based	Empty set	Empty set	Empty set
Feature ordering	None	None	None	None	Ranking-based
Search method	BCA optimization algorithm	BCA optimization algorithm	Stepwise forward search	Stepwise forward backward search	Incremental with replacement

Table 7
Description of the new datasets used in our detailed comparative evaluation.

Data set	Number of instances	Number of Attributes	Number of Classes
Colon	62	2000	2
Leukemia	72	7070	2
ALLAML	72	7129	2
Prostate	102	5966	2
CLL_SUB	111	11,340	3
Average	83.8	6701.0	

FSS capability through optimization of the weights of the neural network.

Earlier we compared BCA-based FSS algorithms with two wrapper techniques, i.e., the SFS and the SFFS, and a filter-wrapper technique, i.e., IWSSr. However, the comparison of a wrapper approach with a hybrid FSS technique, without considering the effect of the interior components of the FSS technique separately, might not be completely fair. In fact, there are three main modules that differentiate the experimental FSS methods in this paper, including a type of the search method, an initialization strategy, and a feature ordering technique. Table 6 shows the detailed structure of each of the experimental FSS techniques studied in this paper. In order to obtain a detailed comparison between BCA-algorithm-based FSS and the IWSSr, and to further explore the effect of interior components, we investigated the effect of presence and absence of feature ordering. Additionally, we were interested to study the effectiveness of the corresponding approaches when the dataset dimensionality is high (more than 1000 features). To this end, we considered comparing four methods including the BCA-Zero, BCA-Zero with feature ordering (here referred to as BCA-withOrder), IWSSr, and IWSSr without feature ordering (here referred to as IWSSr-withoutOrder) on 10 datasets. Five out of the ten datasets were explained earlier and can be found in Table 2. The new five datasets, with number of attributes ranging from 2000 to 11,340, are publicly available and obtained form the ASU feature selection repository [62]. The full description of the new datasets is summarized in Table 7. For feature ordering, to obtain an unbiased comparison, we used the same ordering that the IWSSr employs, i.e., sorting the features based on the symmetrical uncertainty (SU) as the feature usefulness [24].

Unlike the datasets used in Section 3, the new datasets have relatively small numbers of samples (average of 83.8); thus, the outer stratified hold-out (70% training - 30% test) strategy that was used earlier might not be an appropriate choice for performance estimation. Therefore, we performed a 5-fold stratified CV strategy and reported the AUC, the number of subset evaluations, and the number of selected features, averaged over 5 runs, in Table 8. We were interested in making comparisons in order to investigate the effect of individual and combined modules including the search method and the feature ordering on the resulting number of subset evaluations and the number of selected features. Wilcoxon signed ranks test [59] is the statistical test we utilized for each of the pair-wise comparisons over ten datasets for each of the classifiers separately. Considering the comparison of BCA search with the incremental search with replacement strategy, the performance of BCA-Zero and BCA-withOrder was higher than that of the IWSSr-withoutOrder and IWSSr with average factors of 3.3 and 2.3, respectively (with statistically significant differences), in terms of the number of subset evaluations. The difference was much higher for higher dimensionalities. This suggests a difference in the complexity, which will be discussed further later. More interestingly, we can see that exploiting the feature ordering had a reduction effect for the incremental with replacement search approach (factor of 1.7), while the same effect was not seen for BCA. In fact, BCA optimization showed more robustness to the ordering. In terms of the number of selected features, the incremental with replacement was capable of finding smaller subsets than BCA. Indeed, the former was specifically designed to reduce the number of selected features, while the BCA optimization algorithm focused on efficiency. However, the difference became much smaller when utilizing the feature ordering for both search strategies. In fact, exploiting the feature ordering reduced the number of selected features for BCA by 44%. This suggests a further study on the number of selected features and the effect of feature ordering on BCA-based FSS. Moreover, if we compare the BCA-Zero (without feature ordering) and the IWSSr (with feature ordering), the number of subset evaluations was reduced by a factor of 2.3 (with statistically significant difference). Lastly, note that the AUC obtained by the four methods were all comparable.

We saw earlier that the BCA-Zero and BCA-Initialized algorithms outperformed the SFS, SFFS, and IWSSr techniques in terms of the number of subset evaluations. We investigated the complexity of these approaches with theoretical and experimental formula of the number of subset evaluations $f(N)$ as a function of the number of initial attributes N. The formula for the case of the SFS technique is deterministic and can be derived analytically. For the case of the proposed algorithms and SFFS technique, we utilized regression to find experimental formulas. We fit two linear models and a quadratic polynomial to the data obtained by the proposed algorithms and the SFFS technique, correspondingly. For the case of IWSSr, approximation was not straight-forward; thus, we used the experimental values obtained earlier. The worst-case complexity of IWSSr is $\mathcal{O}(N^2)$, while in practice it is usually less and depends on different factors such as the number of selected features. The three experimental expressions for BCA-Zero, BCA-Initialized, and SFFS as well as the analytical function for the SFS are shown in Fig. 3. The error bars in Fig. 3 show the exact data obtained through our experiments. For example, the error bar on the SFFS curve at $N = 60$ is obtained on the Sonar dataset with 60 attributes using the SVM, MLP, and NB classifiers (total of 3 points). Taking the hypothesis that the actual points indeed follow the approximated curves, which might be in fact a reasonable hypothesis by a subjective evaluation of Fig. 3, we might conclude that the number of subset evaluations for the BCA-Zero and BCA-Initialized algorithms follow a linear complexity, with some scaling factor (3–5), with respect to the number of initial attributes N. In fact, the complexity of the BCA optimization algorithm is $\mathcal{O}(SN)$, where S is the number of BCA scans over N input attributes. Having a similar judgment and considering the exact formula for the SFS technique, we might conclude that the SFS and SFFS techniques both follow

Table 8
Comparisons of the classification performance in terms of AUC, the computational complexity in terms of the number of subset evaluations (#subsets), and the number of selected features (#Feats) of the BCA-Zero and IWSSr FSS with and without using the feature ordering, for ten datasets using SVM, MLP, and NB classifiers.

Dataset	BCA-Zero			IWSSr-withoutOrder			BCA-withOrder			IWSSr		
	AUC	#subsets	#Feats	AUC	#subsets	#Feats	AUC	#subsets	#Feats	AUC	#subsets	#Feats
SVM												
Polyp	0.837	316.0	21.8	0.742	418.6	7.0	0.851	237.0	9.2	0.715	303.0	3.4
Sonar	0.829	276.0	19.8	0.771	265.8	5.6	0.821	240.0	15.0	0.802	417.0	8.4
Parkinson	0.902	79.2	10.0	0.901	94.0	6.0	0.891	48.4	5.2	0.896	73.4	3.4
Breast	0.993	96.0	13.8	0.992	179.0	7.0	0.992	96.0	9.4	0.991	164.2	6.6
Climate	0.949	61.2	8.0	0.951	66.4	5.2	0.947	43.2	8.4	0.951	96.8	7.0
Colon	0.674	4000.0	14.6	0.739	13,695.6	6.2	0.882	4000.0	5.0	0.807	9960.8	4.0
Leukemia	0.850	14,140.0	18.8	0.724	57,633.8	7.2	0.988	14,140.0	2.0	0.988	19,793.4	1.8
ALLAML	0.871	14,258.0	18.6	0.785	69,050.4	8.8	0.968	14,258.0	1.4	0.968	17,107.4	1.4
Prostate	0.901	11,932.0	27.0	0.901	62,746.6	9.8	0.962	11,932.0	4.6	0.967	26,215.8	3.4
CLL_SUB	0.810	22,680.0	63.0	0.728	195,470.6	16.8	0.670	29,484.0	74.2	0.644	235,939.0	21.6
Mean	**0.862**	**6783.8**	**21.54**	**0.823**	**39,962.1**	**7.96**	**0.897**	**7447.9**	**13.44**	**0.873**	**31,007.1**	**6.1**
Average rank	**2.25**	**3.05**	**1.2**	**2.95**	**1.4**	**2.8**	**2.25**	**3.65**	**2.35**	**2.55**	**1.9**	**3.65**
				NS	**S**	**S**				**NS**	**S**	**S**
MLP												
Polyp	0.869	363.4	28.4	0.850	470.8	7.8	0.859	395.0	24.0	0.863	627.4	11.2
Sonar	0.826	300.0	25.0	0.784	409.2	9.2	0.809	264.0	17.0	0.832	390.2	8.6
Parkinson	0.899	74.8	14.0	0.891	80.0	3.6	0.886	57.2	5.8	0.892	73.6	3.4
Breast	0.990	114.0	13.4	0.992	195.8	8.4	0.990	78.0	10.8	0.991	144.6	5.8
Climate	0.952	54.0	10.2	0.953	76.0	6.0	0.951	54.0	8.0	0.949	108.8	7.6
Colon	0.815	4000.0	13.8	0.789	15,778.6	7.0	0.870	4400.0	5.4	0.878	9969.6	4.0
Leukemia	0.859	14,140.0	18.4	0.893	61,038.4	7.6	0.984	14,140.0	2.2	0.984	22,619.2	2.2
ALLAML	0.912	14,258.0	18.4	0.893	61,038.4	7.6	0.968	14,258.0	1.4	0.968	17,107.4	1.4
Prostate	0.901	11,932.0	27.0	0.901	62,746.6	9.8	0.974	11,932.0	5.4	0.968	27,417.4	3.6
CLL_SUB	0.810	22,680.0	63.0	0.728	195,470.6	16.8	0.793	22,680.0	57.2	0.747	159,248.8	14.0
Mean	**0.883**	**6791.6**	**23.16**	**0.867**	**39,730.4**	**8.38**	**0.908**	**6825.8**	**13.72**	**0.907**	**23,771.7**	**6.18**
Average rank	**2.4**	**3.35**	**1**	**3.15**	**1.2**	**2.8**	**2.45**	**3.55**	**2.5**	**2**	**1.9**	**3.7**
				NS	**S**	**S**				**NS**	**S**	**NS**
NB												
Polyp	0.824	379.2	23.0	0.852	566.2	11.4	0.838	347.6	23.4	0.829	717.0	14.2
Sonar	0.805	312.0	21.4	0.774	403.4	8.4	0.830	276.0	22.0	0.759	433.2	9.2
Parkinson	0.906	92.4	7.4	0.906	90.2	5.0	0.886	57.2	5.8	0.892	73.6	3.4
Breast	0.992	144.0	9.6	0.990	144.4	6.4	0.990	78.0	10.8	0.991	146.8	6.2
Climate	0.947	57.6	7.4	0.952	82.8	6.2	0.951	50.4	7.0	0.949	92.8	6.2
Colon	0.818	4400.0	14.6	0.801	12,788.8	5.8	0.849	4000.0	5.0	0.869	9836.8	4.0
Leukemia	0.868	14,140.0	15.4	0.850	63,270.2	8.0	0.984	14,140.0	2.0	0.972	19,793.4	1.8
ALLAML	0.853	14,258.0	15.0	0.877	65,310.2	8.2	0.968	14,258.0	1.8	0.968	18,533.0	1.6
Prostate	0.823	11,932.0	29.4	0.817	75,044.0	12.4	0.961	11,932.0	4.8	0.958	28,514.4	3.8
CLL_SUB	0.746	22,680.0	51.2	0.622	219,794.0	18.6	0.687	22,680.0	36.8	0.726	170,355.4	14.8
Mean	**0.858**	**6839.5**	**19.44**	**0.844**	**43,749.2**	**0.904**	**0.894**	**6781.9**	**11.94**	**0.891**	**24,849.6**	**6.52**
Average rank	**2.65**	**2.9**	**1.3**	**2.9**	**1.6**	**2.85**	**2.1**	**3.8**	**2.1**	**2.35**	**1.7**	**3.75**
				NS	**S**	**S**				**NS**	**S**	**S**

Note: when considering the results for Wilcoxon signed ranks statistical test, NS (not significant) and S (significant) under the results of the IWSSr-withoutOrder and IWSSr are obtained through a pairwise comparison with the BCA-Zero and BCA-withOrder, respectively. The average rankings are obtained through the Friedman ranking scheme when considering all methods, where the highest rank corresponds to the smallest value.

a quadratic complexity formula with respect to N, and there is a scaling factor (approximately 5) for the case of the SFFS technique. The approximated curve for the BCA-Zero over 10 datasets (from Table 8) as well as the exact points with error bars for the IWSSr technique are shown in Fig. 4. The same linear complexity for BCA-based FSS still exists when considering high dimensional datasets. As for IWSSr, the points do not completely follow a smooth trend. This is due to the deviation of the dataset-based factors such as the number of important features, which affects the complexity of the IWSSr.

To gain more insights into the reason why BCA-based FSS algorithms are linear, whereas the SFS and SFFS techniques are quadratic, we considered the graph-based representation of feature subsets (an example for N = 4 is shown in Fig. 1). Using the binary representation of feature subsets, we can represent feature subsets for any dataset with N number of attributes, as a graph with N+1 layers, where the L^{th} layer contains $\binom{N}{L}$ subsets. The SFS method starts searching from the subset of all 0 s (L = 0) and scans the graph in a greedy-algorithm-based manner through the last layer (L = N). In other words, once the SFS technique finds the best subset of the L^{th} layer, it searches throughout N-L candidates in the L+1th layer; thus, the total number of subset evaluations is quadratic with respect to N. The SFFS technique follows the same greedy approach, except that it allows the backward search through the previous layers while performing the graph search. Note that regardless of what layer the best subset of features is located in, the SFS and SFFS techniques scan all layers at least once, where for each layer L, an order of N evaluations is required for a forward or backward search. BCA-based FSS algorithms overcome the problem of computational complexity by reducing the number of evaluations needed to pass over a layer in the graph search, i.e., when in layer L, only one subset of either the L+1th layer (adding a feature) or the L-1th layer (removing a feature) will be examined. In fact, BCA-based FSS follows a line search strategy through the graph. Moreover, one might notice that the BCA algorithm does not necessarily scan all the layers. To make it clear, consider the example of Sonar dataset with 60 features. One can see from Table 3 that the number of selected features obtained by five methods and three classifiers were all less than 30% of N;

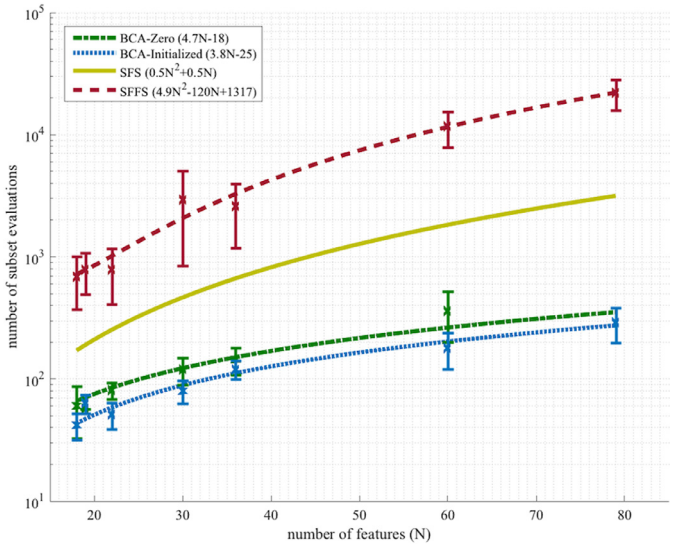

Fig. 3. Number of wrapper subset evaluations as a function of the number of input attributes (N). Linear and non-linear fitted function curves for the BCA algorithms with zero and feature-rating-based initializations, SFS, and SFFS techniques are illustrated. Note the log scale in the vertical axis.

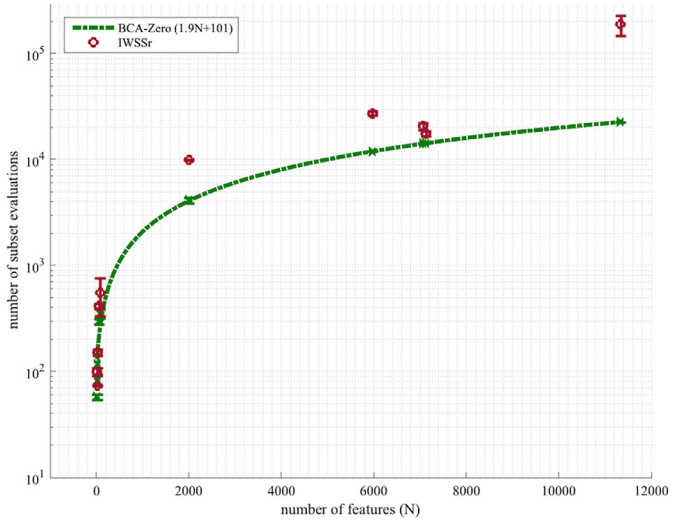

Fig. 4. Number of wrapper subset evaluations as a linear function of the number of input attributes (N) for BCA-Zero, as well as the error bars obtained from the experimental results for both BCA-Zero and IWSSr. Note the log scale in the vertical axis.

therefore, about 70% of layers were redundantly scanned by the SFS and SFFS techniques. On the other hand, BCA-based FSS never goes far away from the optimal or sub-optimal subsets in its search space; thus, substantially more efficient.

In Section 3, we investigated the performance of each algorithm based on its AUC obtained on an independent test set, but we did not discuss the results obtained in the 10-fold CV scheme (5-fold for IWSSr) on the validation set through the FSS process. Table 9 shows the performance comparison in the validation test. Our exploration indicated that the validation results were also comparable. To confirm that, we performed statistical analysis with the methodology we used earlier. There was no statistically significant difference among our comparisons of the validation AUC results. In fact, we might conclude that our BCA optimization algorithm successfully obtained a local optimum in the space of feature subsets with the objective function value (i.e., the validation AUC) compa-

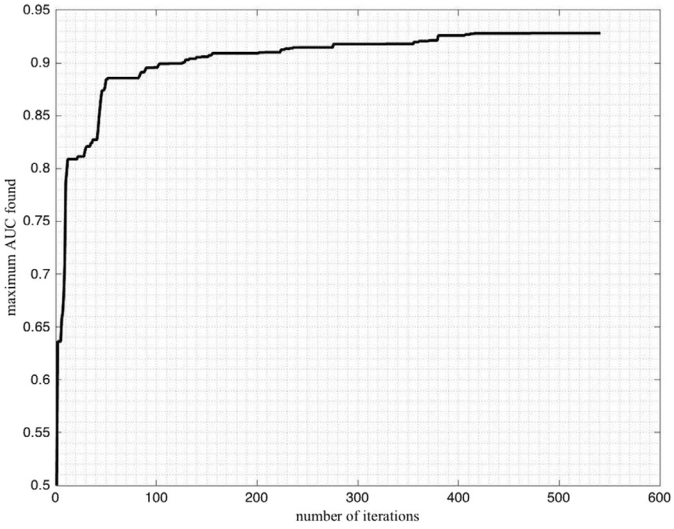

Fig. 5. Optimization trend of the objective AUC function over the number of iterations in our BCA optimization algorithm. The figure shows the trend for the case of the Sonar dataset and the SVM classifier.

rable to that of the reference techniques, while reducing the number of iterations. Investigating the validation results, one might notice that although some approaches reached a better AUC on the validation set, the corresponding performance on the independent test set was lower. This is consistent with the study on the performance evaluation of the FSS techniques, conducted in [51].

Most of sequential search techniques have to be executed completely in order to obtain a sub-optimum point as the best subset of features. As mentioned earlier, our BCA optimization algorithm is an anytime iterative algorithm, i.e., it can be stopped earlier by considering early stopping criteria. Fig. 5 illustrates a trend of the objective function (AUC) maximization through the iterations (i.e., the subset candidates) for an example case of the FSS for the Sonar dataset and SVM classifier. This single case was only selected for the ease of illustration, while curves for all other BCA-based FSS experiments followed the same trend. In this example, with a total of 540 iterations, there was almost no significant improvement of the objective function after the 400th iteration; therefore, one might stop the algorithm at this point based on the application needs. We might also consider this property as one of the limitations of our BCA optimization algorithm. In other words, our BCA algorithm might end up in a local optimum such that there is no way of improving the objective functions even by increasing the number of iterations. In fact, this is a common limitation of most local search methods. This can be studied further for the case of our BCA optimization algorithm by considering techniques that make trade-off between the number of iterations and the objective function results. For the classification application, we showed that even though this limitation exists, the performance results were convincing while significant reductions in the computational complexity were obtained.

5. Conclusion

We proposed and developed an efficient iterative deterministic local optimization algorithm, namely, binary coordinate ascent (BCA). Our BCA algorithm can be utilized in the optimization frameworks with a binary representation of the input variables and continuous objective function values. To study the efficiency and the performance of our BCA optimization algorithm, we investigated this algorithm for the task of FSS in classification applications. To this end, we first represented the space of

Table 9
Validation performance (in terms of AUC) comparison of our BCA algorithms with zero and feature-rating-based initializations with the SFS, SFFS, and IWSSr FSS techniques for seven datasets using the SVM, MLP, and NB classifiers.

Dataset	BCA-Zero	BCA-Initialized	SFS	SFFS	IWSSr
SVM					
Polyp	0.91	0.93	0.95	0.96	0.80
Sonar	0.93	0.91	0.93	0.93	0.92
Landsat	0.98	0.98	0.98	0.98	0.98
Parkinson	0.95	0.95	0.95	0.95	0.94
Breast Cancer	0.99	0.99	0.99	0.99	0.99
Segmentation	0.99	0.99	0.99	0.99	0.99
Climate	0.96	0.95	0.95	0.96	0.95
Mean ± SD	**0.96 ± 0.03**	**0.96 ± 0.03**	**0.96 ± 0.03**	**0.96 ± 0.03**	**0.96 ± 0.03**
Average rank	**2.71**	**3.36**	**2.79**	**2.29**	**3.86**
		NS	**NS**	**NS**	**NS**
MLP					
Polyp	0.88	0.93	0.90	0.94	0.86
Sonar	0.96	0.96	0.96	0.99	0.92
Landsat	0.98	0.98	0.98	0.98	0.98
Parkinson	0.99	0.99	0.99	0.99	0.98
Breast Cancer	0.99	0.99	0.99	0.99	0.99
Segmentation	0.99	0.99	0.99	0.99	0.99
Climate	0.95	0.97	0.95	0.98	0.96
Mean ± SD	**0.98 ± 0.02**	**0.98 ± 0.01**	**0.98 ± 0.02**	**0.99 ± 0.02**	**0.96 ± 0.03**
Average rank	**3.29**	**2.64**	**3.14**	**2.07**	**3.86**
		NS	**NS**	**NS**	**NS**
NB					
Polyp	0.90	0.90	0.88	0.90	0.89
Sonar	0.93	0.93	0.92	0.94	0.95
Landsat	0.96	0.96	0.96	0.96	0.97
Parkinson	0.95	0.95	0.95	0.95	0.94
Breast Cancer	0.99	0.99	0.99	0.99	0.99
Segmentation	0.99	0.99	0.99	0.99	0.98
Climate	0.95	0.95	0.95	0.95	0.94
Mean ± SD	**0.95 ± 0.03**	**0.95 ± 0.03**	**0.95 ± 0.04**	**0.95 ± 0.03**	**0.95 ± 0.03**
Average rank	**2.79**	**2.79**	**3.43**	**2.57**	**3.43**
		NS	**NS**	**NS**	**NS**

Note: when considering the results for Holm post hoc statistical analysis, NS and S under the results of a method indicate that there is no significant or a significant difference between the corresponding method and the BCA-Zero as the control algorithm, respectively.

feature subsets using binary vectors of 0 s and 1 s. Based on that, we proposed two efficient wrapper-based FSS techniques, namely, BCA-Zero and BCA-Initialized. The task of our wrapper-based FSS approaches was to efficiently find the best subset of features in terms of the AUC. We experimented on seven datasets (18–79 attributes) using three classifiers, i.e., the SVM, MLP, and NB, and compared the performance and efficiency results of our proposed algorithms with those of two of the most popular wrapper-based FSS techniques, i.e., the SFS and SFFS. Additionally, to gain further knowledge of our BCA optimization search method, we compared the proposed algorithms with an efficient filter-wrapper-based FSS technique, i.e., IWSSr. Five more datasets (2000–11,340 attributes) were explored for this purpose. The performance analysis demonstrated that the AUC results for our proposed algorithms were comparable to those of the SFS, SFFS, and IWSSr techniques. Investigating the efficiency of the approaches, we demonstrated that with our BCA optimization algorithm, the number of subset evaluations and the processing time of the FSS algorithm were reduced substantially, compared with those of the SFS and SFFS techniques. We also showed that the number of subset evaluations was reduced for BCA-based FSS algorithms when comparing with filter-wrapper IWSSr, where the difference was greater for the datasets with higher dimensionalities. We also performed statistical analysis for all of the experiments and comparisons to confirm our statements. Additionally, our detailed analysis of the BCA based algorithms suggested a further study on the BCA-based FSS system in terms of the initialization strategy and the feature ordering technique. Through our analysis and experiments, we found that our BCA-based algorithms follow a linear complexity (the number of subset evaluations) with respect to the number of initial attributes, whereas the SFS and SFFS techniques follow a quadratic form. Correspondingly, one can use the BCA optimization algorithm as an efficient alternative approach in applications in which efficiency is a requirement, specifically in FSS for classification of datasets with a high number of initial attributes.

Acknowledgements

The authors are grateful to Nima Tajbakhsh, Yisong Chen, Nazanin Makkinejad, Junchi Liu, Paul Forti, and other members in the Suzuki Lab for their valuable suggestions and discussions.

References

[1] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[2] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (2014) 16–28.
[3] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, A review of feature selection methods on synthetic data, Knowl. Inf. Syst. 34 (2013) 483–519.
[4] M. Dash, H. Liu, Feature selection for classification, Sience 1 (1997) 131–156.
[5] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Springer Science & Business Media, 2012.
[6] H. Liu, L. Yu, S.S. Member, L. Yu, S.S. Member, Toward integrating feature selection algorithms for classification and clustering, Knowl. Data Eng. IEEE Trans. 17 (2005) 491–502.
[7] K. Suzuki, Machine learning in computer-aided diagnosis of the thorax and colon in ct: a survey, IEICE Trans. Inf. Syst. 96 (2013) 772–783.
[8] S. Chen, K. Suzuki, H. MacMahon, Development and evaluation of a computer-aided diagnostic scheme for lung nodule detection in chest radiographs by means of two-stage nodule enhancement with support vector classification, Med. Phys. 38 (2011) 1844–1858.

[9] K. Suzuki, M. Zarshenas, J. Liu, Y. Fan, N. Makkinejad, P. Forti, et al., Development of computer-aided diagnostic (CADx) system for distinguishing neoplastic from nonneoplastic lesions in CT colonography (CTC): toward CTC beyond detection, in: 2015 IEEE Int. Conf. Syst. Man, Cybern., IEEE, 2015, pp. 2262–2266.
[10] M. Bacauskiene, A. Verikas, A. Gelzinis, D. Valincius, A feature selection technique for generation of classification committees and its application to categorization of laryngeal images, Pattern Recognit. 42 (2009) 645–654.
[11] S. Li, H. Wu, D. Wan, J. Zhu, An effective feature selection method for hyperspectral image classification based on genetic algorithm and support vector machine, Knowl. Based Syst. 24 (2011) 40–48.
[12] P. Somol, J. Novovicová, P. Pudil, Efficient feature subset selection and subset size optimization, Pattern Recognit. Recent Adv. (2010) 1–24.
[13] J. Tang, S. Alelyani, H. Liu, Feature selection for classification: a review, Data Classif. Algorithms Appl. (2014) 37–64.
[14] D. Zhang, S. Chen, Z. Zhou, Constraint score: a new filter method for feature selection with pairwise constraints, Pattern Recognit. 41 (2008) 1440–1451.
[15] H. Liu, J. Sun, L. Liu, H. Zhang, Feature selection with dynamic mutual information, Pattern Recognit. 42 (2009) 1330–1339.
[16] C. Shang, M. Li, S. Feng, Q. Jiang, J. Fan, Feature selection via maximizing global information gain for text classification, Knowl. Based Syst. 54 (2013) 298–309.
[17] A.K. Uysal, S. Gunal, A novel probabilistic feature selection method for text classification, Knowl. Based Syst. 36 (2012) 226–235.
[18] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, et al., A survey on filter techniques for feature selection in gene expression microarray analysis, IEEE/ACM Trans. Comput. Biol. Bioinform. 9 (2012) 1106–1119.
[19] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Trans. Neural Netw. 5 (1994) 537–550.
[20] R. Kohavi, R. Kohavi, Wrappers for feature subset selection, Artif. Intell. 97 (1997) 273–324.
[21] W. Ng, D. Yeung, M. Firth, E. Tsang, X. Wang, Feature selection using localized generalization error for supervised classification problems using RBFNN, Pattern Recognit. 41 (2008) 3706–3719.
[22] H. Uğuz, A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm, Knowl. Based Syst. 24 (2011) 1024–1032.
[23] P. Bermejo, L. De La Ossa, J.A. Gámez, J.M. Puerta, Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking, Knowl. Based Syst. 25 (2012) 35–44.
[24] P. Bermejo, J.A. Gámez, J.M. Puerta, Incremental wrapper-based subset selection with replacement: an advantageous alternative to sequential forward selection, in: 2009 IEEE Symp. Comput. Intell. Data Mining, CIDM 2009 - Proc., 2009, pp. 367–374.
[25] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1986) 81–106.
[26] J. Zhu, S. Rosset, T. Hastie, R. Tibshirani, 1-norm support vector machines, Nips. (2003) 49–56.
[27] S.-W. Lin, Z.-J. Lee, S.-C. Chen, T.-Y. Tseng, Parameter determination of support vector machine and feature selection using simulated annealing approach, Appl. Soft Comput. 8 (2008) 1505–1512.
[28] P.-W. Huang, C.-H. Lee, Automatic classification for pathological prostate images based on fractal analysis, IEEE Trans. Med. Imag. 28 (2009) 1037–1050.
[29] L. Böröczky, L. Zhao, K.P. Lee, Feature subset selection for improving the performance of false positive reduction in lung nodule CAD, IEEE Trans. Inf. Technol. Biomed. 10 (2006) 504–511.
[30] P. Campadelli, E. Casiraghi, D. Artioli, A fully automated method for lung nodule detection from postero-anterior chest radiographs, IEEE Trans. Med. Imaging. 25 (2006) 1588–1603.
[31] R. Wang, K. Tang, Feature selection for maximizing the area under the ROC curve, in: 2009 IEEE Int. Conf. Data Min. Work., 2009, pp. 400–405.
[32] J. Canul-Reich, L.O. Hall, D. Goldgof, S.A. Eschrich, Feature selection for microarray data by AUC analysis, in: Syst. Man Cybern. 2008. SMC 2008. IEEE Int. Conf., 2008, pp. 768–773.
[33] C. Marrocco, R.P.W. Duin, F. Tortorella, Maximizing the area under the ROC curve by pairwise feature combination, Pattern Recognit. 41 (2008) 1961–1974.
[34] B. Sahiner, N. Petrick, H.P. Chan, L.M. Hadjiiski, C. Paramagul, M.A. Helvie, et al., Computer-aided characterization of mammographic masses: Accuracy of mass segmentation and its effects on characterization, IEEE Trans. Med. Imag. 20 (2001) 1275–1284.
[35] J. Xu, K. Suzuki, Max-AUC feature selection in computer-aided detection of polyps in CT colonography, Biomed. Heal. Informatics, IEEE J. 18 (2014) 585–593.
[36] H. Vafaie, I.I.F. Imam, Feature selection methods : genetic algorithms vs . greedy-like search, Proc. Int. Conf. Fuzzy Intell. Control Syst. 1 (1994).
[37] J. Kennedy, R. Eberhart, Particle swarm optimization, Neural Networks, 1995, in: Proceedings., IEEE Int. Conf. 4, vol.4, 1995, pp. 1942–1948.
[38] S.S. Mohamed, M.M.A. Salama, Prostate cancer spectral multifeature analysis using TRUS images, IEEE Trans. Med. Imag. 27 (2008) 548–556.
[39] Y. Zhang, S. Wang, P. Phillips, G. Ji, Binary PSO with mutation operator for feature selection using decision tree applied to spam detection, Knowl. Based Syst. 64 (2014) 22–31.
[40] R. Meiri, J. Zahavi, Using simulated annealing to optimize the feature selection problem in marketing applications, Eur. J. Oper. Res. 171 (2006) 842–858.
[41] J.C.W. Debuse, V.J.R. Smith, Feature subset selection within a simulated annealing data mining algorithm, J. Intell. Inf. Syst. 9 (1997) 57–81.
[42] P. Pudil, J. Novovičová, J. Kittler, Floating search methods in feature selection, Pattern Recognit. Lett. 15 (1994) 1119–1125.
[43] P. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach, Prentice Hall, 1982.
[44] Y. Liu, Y. Zheng, FS_SFS: a novel feature selection method for support vector machines, Pattern Recognit. 39 (2006) 1333–1345.
[45] J. Hua, W. Tembe, E. Dougherty, Performance of feature-selection methods in the classification of high-dimension data, Pattern Recognit. 42 (2009) 409–424.
[46] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S.S. Keerthi, S. Sundararajan, A dual coordinate descent method for large-scale linear SVM, in: Proc. 25th Int. Conf. Mach. Learn. - ICML '08., 2008, pp. 408–415.
[47] Z.Q. Luo, P. Tseng, On the convergence of the coordinate descent method for convex differentiable minimization, J. Optim. Theor. Appl. 72 (1992) 7–35.
[48] C. CORTES, V. VAPNIK, Support-vector networks, Chem. Biol. Drug Des. 297 (2009) 273–297.
[49] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1998.
[50] H. Zhang, The optimality of naive bayes, in: Proc. Seventeenth Int. Florida Artif. Intell. Res. Soc. Conf. FLAIRS 2004. 1, 2004, pp. 1–6.
[51] J. Reunanen, Overfitting in making comparisons between variable selection methods, J. Mach. Learn. Res. 3 (2003) 1371–1382.
[52] M. Cavazzuti, Optimization Methods: From Theory to Scientific Design and Technological Aspects in Mechanics, Springer Science & Business Media, 2012.
[53] W. Banzhaf, P. Nordin, R. Keller, F. Francone, Genetic Programming: An Introduction, Morgan Kaufmann Publishers, San Francisco, 1998.
[54] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, 1999.
[55] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (2006) 861–874.
[56] K. Bache, M. Lichman, UCI Machine Learning Repository, University of California, School of Information and Computer Science., Irvine, CA, 2013 Phys. Rev. E.
[57] C. Chang, C. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27.
[58] M. Hall, E. Frank, G. Holmes, The WEKA data mining software: an update, ACM SIGKDD Explor. Newsl. 11 (2009) 10–18.
[59] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[60] S. Garcia, F. Herrera, An extension on " Statistical Comparisons of Classifiers over Multiple Data Sets " for all Pairwise Comparisons, J. Mach. Learn. Res. 9 (2008) 2677–2694.
[61] S. Garcia, A. Fernnndez, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Inf. Sci. (Ny). 180 (2010) 2044–2064.
[62] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, et al., Feature Selection: A Data Perspective, 2016, pp. 1–73.