Information Sciences 181 (2011) 4625–4641

Contents lists available at ScienceDirect

Information Sciences

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

mr^2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification

Alper Unler [a], Alper Murat [b,*], Ratna Babu Chinnam [b]

[a] Department of Information Management Systems, KKK, Yucetepe, Ankara, Turkey
[b] Department of Industrial and Manufacturing Engineering, Wayne State University, Detroit, MI, USA

ARTICLE INFO

Article history:
Available online 8 June 2010

Keywords:
Feature selection
Support vector machine
Classification
Mutual information
Filters
Wrappers
Particle swarm optimization

ABSTRACT

This paper presents a hybrid filter–wrapper feature subset selection algorithm based on particle swarm optimization (PSO) for support vector machine (SVM) classification. The filter model is based on the mutual information and is a composite measure of feature relevance and redundancy with respect to the feature subset selected. The wrapper model is a modified discrete PSO algorithm. This hybrid algorithm, called maximum relevance minimum redundancy PSO (mr^2PSO), is novel in the sense that it uses the mutual information available from the filter model to weigh the bit selection probabilities in the discrete PSO. Hence, mr^2PSO uniquely brings together the efficiency of filters and the greater accuracy of wrappers. The proposed algorithm is tested over several well-known benchmarking datasets. The performance of the proposed algorithm is also compared with a recent hybrid filter–wrapper algorithm based on a genetic algorithm and a wrapper algorithm based on PSO. The results show that the mr^2PSO algorithm is competitive in terms of both classification accuracy and computational performance.

1. Introduction

Many practical applications of classification involve a large volume of data and/or a large number of features/attributes. Since these datasets are usually collected for reasons other than mining the data (e.g. classification), there may be some redundant or irrelevant features [13]. This is especially important when a large number of features exist and there are comparably few training sample data points, making feature vector dimensionality reduction an imperative. Examples of this include gene selection from microarray data to separate healthy patients from cancer patients, text categorization to perform automatic sorting of URLs into a web directory, and detection of unsolicited spam email [14]. Extraction of valuable information from these datasets requires exhaustive search over the sample space. This brings about such challenges as managing computational time complexity while extracting compact yet effective models. A common approach for overcoming these challenges is to employ dimensionality reduction (e.g., feature subset selection) techniques.

While some preprocessing procedures (e.g., filtering) can help reduce the effective feature set size, further reduction of the feature subset is required to build good predictor models. Furthermore, feature subset selection can improve accuracy of classification by reducing estimation errors due to finite sample size effects [21]. Other benefits associated with a compact model are the avoidance of over-fitting for better generalization, reduced burden on data collection, and reduced computational effort. Feature selection is the process of defining the most informative and discriminative features in a dataset for the

* Corresponding author.
E-mail addresses: aunler@ttmail.com (A. Unler), amurat@wayne.edu (A. Murat), r_chinnam@wayne.edu (R.B. Chinnam).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.05.037

data mining task (e.g., classification). The two basic steps in a typical feature subset selection process are the specification of the parameter set (that determines the performance of the data mining task) and the search for the best subset. The parameter set often includes the selection algorithm, the learning machine (e.g., classifier), and the process for error estimation. It is commonly observed that there is no single best parameter set combination valid for all data mining tasks and all types/sizes of databases. In addition, the performance of the feature selection process is strongly dependent on the selection algorithm employed.

Feature selection algorithms broadly fall into three categories: *filter* models, *wrapper* models and *hybrid* models [9,14,29]. Filter models generally make use of statistical or probabilistic characteristics of databases and are independent of the learning machines. Given that filters do not involve a learning machine, they are computationally efficient and are preferable for high-dimensional databases. In comparison, wrapper models use learning machines and select feature subsets based on the prediction performance. As a result, their computational overhead is relatively high compared with filters and, thus, they are not effective for high-dimensional databases. The main advantage of wrappers over filters is their prediction accuracy. Since a wrapper's search for the best feature subset is guided by prediction accuracy, the results are generally more promising than results based on filters. Hybrid models, on the other hand, benefit from the advantages of both the filters and the wrappers and thus promise better results.

In most real-world datasets, not all of the attributes contribute to the definition or determination of class labels. In theory, increasing the size of the feature vector is expected to provide more discriminating power. In practice, however, excessively large feature vectors significantly slow down the learning process as well as cause the classifier to over-fit the training data and compromise model generalization [59]. To find good feature subsets, the majority of the search effort should be utilized on identifying relevant and non-redundant features.

Most of the studies that hybridize filters and wrappers use filters either for the ranking of features or for the reduction of the number of candidate features. In particular, these hybrid methods are based on a sequential (e.g., two-step) approach where the first step is usually based on filter methods to reduce the number of features considered in the second stage. Using this reduced set, a wrapper method is then employed to select the desired number of features in the second stage. However, no study truly integrates both methods within a search process. In this study, we propose a hybrid *filter and wrapper* framework for a feature subset selection algorithm based on particle swarm optimization (PSO). This hybrid framework, called maximum relevance minimum redundancy PSO (mr^2PSO), integrates the mutual information based filter model within the PSO based wrapper model. Hence, it brings together the efficiency advantage of filters with the accuracy performance of wrappers.

The rest of the paper is organized as follows: Section 2 gives a review of hybrid models and search algorithms for the feature selection problem. In Section 3, we present the relevance and redundancy of features based on mutual information and the particle swarm optimization (PSO) methodology. In Section 4, we compare the proposed hybrid PSO based filter–wrapper algorithm with the genetic algorithm based hybrid filter–wrapper in [20] and report on performance accuracy and computational efficiency.

2. Overview of feature selection methods

There are a number of studies that provide an overview of feature selection methods as well as guidance on different aspects of this problem [9,21,29,35,43,44]. Most feature subset selection algorithms can be categorized into two types: filter and wrapper algorithms. The main distinction between the two is that filter algorithms select the feature subset before the application of any classification algorithm. By using statistical properties of features, the filter approach eliminates the less important features from the subset [8]. Besides their comparative computational efficiency, there are other compelling arguments for using filter methods. For instance, some filter methods provide a generic selection of variables which are independent of a given learning machine. However, given a classifier, the best feature subset is usually available through the wrapper methods [14].

Lewis [64] and Battiti [65] are some of the earliest authors to propose the use of mutual information (MI) for selecting features in building models. Mutual information of two random variables is a quantity that measures the mutual dependence of the two variables. Chow and Huang [5] propose a method that is based on MI but designed for efficient estimation of MI in high-dimensional datasets. Liu et al. [28] introduce the idea of dynamic mutual information (DMI) in feature selection. They note that a feature is relevant to the classes if it embodies important information about the classes, otherwise it is irrelevant or redundant. While MI is the most commonly employed importance measure in filter methods, several studies have also proposed other measures of importance. Debuse and Rayward-Smith [10] propose an entropic measure based on the information gain and employed a simulated annealing algorithm to address the feature subset selection problem. In the context of feature selection, the notion of irrelevant features is first discussed by Ben-Bassat [2]. Sebban and Nock [45] present a feature selection model based both on information theory and statistical tests. In their method, a feature is selected if and only if the information given by this attribute allows for statistical reduction of class overlaps. Yu and Liu [59] categorize the feature space into three classes; strongly relevant, weakly relevant and irrelevant features. The strong relevance of a feature indicates that the feature is always necessary for an optimal subset; it cannot be removed without affecting the original conditional class distribution. Weak relevance suggests that the feature is not always necessary but may become necessary for an optimal subset under certain conditions. Irrelevance indicates that the feature is not necessary. They argue that an optimal

subset should include all strongly relevant features, a subset of weakly relevant features, and none of the irrelevant features. Mitra et al. [34] describe an unsupervised feature selection algorithm that utilizes feature similarity for redundancy reduction. Liu et al. [5] propose a method that is based on mutual information (MI). Mutual information of two random variables is a quantity that measures the mutual dependence of the two variables. Chow and Huang [28] study the dynamic mutual information (DMI) in feature selection. They note that a feature is relevant to the classes if it embodies important information about the classes; otherwise it is irrelevant or redundant.

Wrapper methods approach the problem of feature subset selection based on the contribution of features to task performance accuracy (e.g., classification). This method uses training data and tests for generalization by employing validation and testing datasets. Since feature selection and learning is concurrently performed under these methods, the prediction power of the resulting classification model tends to be better and more consistent than the alternatives [26]. However, wrapper methods are often criticized for their computational inefficiency caused by the joint processing of learning and feature selection tasks. As a result, most wrapper algorithms are inexact search methods that seek good quality solutions under reasonable computational effort [10,14]. Most wrapper algorithms fall under one of the following categories: exact methods, greedy sequential feature subset selection methods, nested partitioning methods, mathematical programming methods, and metahuristic methods. Since the feature selection problems are *NP-hard*, the optimal solution cannot be guaranteed unless an exhaustive search is carried out. This is only possible for datasets with a small number of features. Although the literature offers some complete search methods and their extensions [36,49,60], finding an optimal feature subset is a combinatorial problem, and hence, suboptimal algorithms are typically used to find acceptable solutions. Sequential forward selection (SFS) [56] and sequential backward selection (SBS) [31,32] are the two basic suboptimal feature selection algorithms. In the last decade, metaheuristic methods such as tabu search [55,62], simulated annealing [10,33], genetic algorithms [17,42,47,54,57,61] have also been applied to solve the feature subset selection problem. In addition, there are feature selection methods based on rough set theory [4,7,19,58,63] and on Boolean independent component analysis [1].

Given our aim to build hybrid feature selection models that strike a good balance between the computational efficiency of filter models and the accuracy performance of wrapper models, we focus our attention in the rest of this section on those studies that employ filter–wrapper hybrid models for feature subset selection. Das [8] propose a hybrid feature selection algorithm that uses boosting and incorporates some of the features of wrapper methods into a fast filter method. The results of comparative study using real-world datasets indicate that proposed method is competitive with wrapper methods while selecting feature subsets much faster. Xing et al. [52] propose a hybrid feature selection based on Markov Blanket filter for high-dimensional genomic microarray data with only 72 data points in a 7130 dimensional space. Their experimental results using different classifiers demonstrate that the proposed method leads to feature subsets outperforming those of regularization methods as well as classification based on all features. Sebban and Nock [45] present a hybrid feature selection model based both on information theory and statistical tests. Using both synthetic and real-world datasets they demonstrate that the hybrid method is able to eliminate irrelevant and redundant features even in very large feature spaces more efficiently than pure wrapper methods. A two-stage hybrid algorithm is presented by Peng et al. [40] to select good features according to the maximal statistical dependency criterion based on mutual information. In the first stage, the method uses maximum relevance minimum redundancy incremental selection to determine the optimal number of features to be selected. In the second stage, it uses classical sequential forward and backward selection algorithms by taking the initial subset as chosen in the first stage. Results of an extensive experimentation using real-world data reveal that the proposed method outperforms the maximum dependency based filter method.

A filter supported sequential hybrid feature selection model (FS-SFS) is presented in [30]. FS-SFS reduces the number of features that has to be tested through the classifier. These pre-selected features are considered to be ‘‘informative” and are then evaluated for the accuracy of classification as in the conventional wrapper method. Experimental results using real-world datasets show that the proposed method provides good accuracy performance while significantly reducing computational time. Peng et al. [39] study filter and wrapper methods for biomarker discovery from microarray gene expression data for cancer classification. They propose a hybrid approach where Fisher's ratio is used as the filtering method. The extensive experimentation using real datasets demonstrates that the hybrid approach outperforms the accuracy obtained from the simple wrapper method while being computationally more efficient. Further, the results show that the hybrid approach significantly outperforms the simple filter method with higher classification accuracies. Somol et al. [48] introduce a flexible hybrid feature selection method based on floating search methods to improve flexibility in dealing with the quality-of-result versus computational time trade-off. They test the performance of the hybrid method using real-world datasets and conclude that the proposed method significantly reduces the search time while achieving comparable accuracies with those of the wrapper methods. Huang et al. [20] present a hybrid genetic algorithm for finding a subset of features that are most relevant to the classification task. Rather than optimizing the classification error rate, they optimize the mutual information between the predictive labels of a trained classifier and the true class labels. The results of an experimental study using real-world datasets indicate that the hybrid method outperforms the accuracy performance of filter methods and is much more efficient than wrapper methods. Uncu and Turksen [51] propose a feature selection algorithm that avoids the problem of over-fitting by first filtering the potential significant features or feature subset combinations and then identifying the best input variable combination by means of a wrapper. To improve the processing time and accuracy of the classifier, Hsu et al. [18] combine the filter and wrapper feature selection methods. They use filter methods based on *F*-score and information gain as the preprocessing step and wrappers based on sequential floating search methods as the post-processing step.

Particle swarm optimization (PSO) is a powerful swarm-based metaheuristic method, originally proposed by Kennedy and Eberhart [22,23] and then later improved by Clerc anmd Kennedy [6], Poli et al. [41], Shi and Eberhart [46]. PSO's search principle is based on the information sharing ability of the biological swarms like birds and fish. Over the past decade, it has been successfully applied to address a very large number of applications especially where the objective function has non-convex nature or the search space is very large. The particle swarm optimization approach has recently gained more attention for solving the feature subset selection problem. Wang et al. [53] integrated rough sets and particle swarm intelligence to be able to find high quality feature subsets. Escalante et al. [12] propose another application of PSO to the problem of full model selection (FMS) for classification. A two-phase feature selection algorithm is presented in [38] based on PSO. In the first phase, a core set of features is searched to obtain a good initial solution, and, in the second phase, new promising features are sequentially added to the core set by a selection method.

In summary, a number of studies have demonstrated that hybridization combines the good characteristics of both wrapper and filter methods. They are more efficient than wrapper methods while providing comparable accuracy [8,16,20,27,30,39,45,48]. Alternatively, hybrid methods are superior to filter methods in terms of accuracy performance while still allowing feature selection in large datasets [20,39,40]. These demonstrated merits of hybridization motivate the proposed approach in this paper. This study contributes to the feature subset selection methodology literature by proposing a hybrid filter–wrapper algorithm based on the swarm intelligence based PSO algorithm. The most salient aspect of our proposed approach is the hybridization strategy of encapsulating the filter model within the wrapper method. This hybridization differs from earlier studies where the filter and wrapper methods operate in sequence. We use an adaptation of the filter model based on mutual information in Peng et al. [40] and integrate it within the PSO based wrapper. The choice of PSO is primarily motivated by its ability to integrate the filter model within the wrapper method. Other motivations include the good performance of PSO in comparison with other evolutionary algorithms and that a hybrid feature selection algorithm based on the swarm intelligence does not exist [11,15]. Our experiments comparing the proposed PSO based hybrid approach with the genetic algorithm based hybridization of filters and wrappers in [20] as well as with the PSO based wrapper methods demonstrate its effectiveness.

3. *mr²PSO*-hybrid PSO algorithm for feature selection

This section describes our proposed *filter–wrapper* framework based on the maximum relevance minimum redundancy filter and particle swarm optimization search heuristic. In general, a feature subset selection problem can be described as follows; Let $\mathscr{K}$ be a dataset of $\mathscr{N}$ records with $\mathscr{D}$ dimensions (features) which is a $\mathscr{K} = \mathscr{N} \times \mathscr{D}$ matrix. The goal of the feature subset selection is to obtain d features from the whole feature space where $d < D$, which optimizes a criterion function. Sebban and Nock [45] categorize the feature selection algorithms into three classes according to what is optimized:

1. algorithms that find the feature subset of a specified dimensionality in which the classes of data are most discriminable,
2. algorithms that find the smallest feature dimensionality for which the discriminability exceeds a specified value, and
3. algorithms that find a compromise between a small subset of features and the class discriminability.

In this study, we consider only the first and third algorithm classes. We further consider various discriminability measures such as the classification accuracy, Kappa statistic (which measures the agreement between two raters who each classify N items into C mutually exclusive categories [66]), and mutual information. In the remainder of this section, we first describe the support vector machine classifier used in our hybrid wrapper–filter algorithm. Second, we describe the filter component (based on mutual information) of the hybrid framework. Finally, we describe how the filter component is integrated within the wrapper PSO algorithm.

3.1. Support vector machine (SVM) classification

In this study, we use the support vector machine (SVM) as the classifier. The SVM, based on statistical learning theory and structural risk minimization, selects key data points as its support vectors, and uses these support vectors for prediction. As

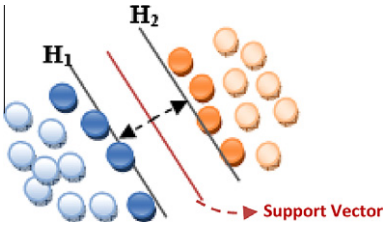


Fig. 1. Illustration of support vector as separating hyperplane between two datasets.

illustrated in Fig. 1, the support vectors help form the separating parallel hyperplanes (H_1 and H_2) that maximize the margin between the two datasets. Intuitively, larger margins lead to lower generalization error of the SVM classifier.

We now give a brief mathematical summary of the classical SVM for binary-class classification. Let K be a dataset, a set of points of the form; $K = \{(x_i, y_i)|x_i \in \mathbb{R}^D,\ y_i \in \{-1, 1\}\}_{i=1}^{N}$ where y_i indicates the class label for point x_i belongs. The goal is to find the maximum-margin hyperplane dividing the points having $y_i = 1$ from those having $y_i = -1$. We can express any hyperplane as the set of points x_i satisfying,

$$W \cdot x_i - b = 0,$$

where $(\cdot)$ denotes the dot product. While the vector W is a normal vector perpendicular to the hyperplane, the parameter $\frac{b}{\|w\|}$ determines the offset of the hyperplane from the origin along the normal vector w. We therefore choose the w and b to maximize distance (margin) between the parallel hyperplanes while still separating the data. The two equations describing these hyperplanes are $(w \cdot x_i - b) = 1$ and $(w \cdot x_i - b) = -1$. For linearly separable training datasets, the distance between these two hyperplanes is $\frac{2}{\|w\|}$, hence, our goal is to minimize $\|w\|$. Furthermore, to prevent data points from falling into the margin, the following constraints are also needed: for each x_i either $(w \cdot x_i - b \geqslant 1)$, of the first class or $(w \cdot x_i - b) \leqslant 1$, of the second and, more compactly $y_i(w \cdot x_i - b \geqslant 1)$, for all $1 \leqslant i \leqslant n$. Putting all of this together, we obtain the following optimization problem:

$$\begin{aligned} &\min \|w\| \\ &\text{s.t.} \quad y_i(w.x_i - b) \geqslant 1, \quad i = 1, \ldots, N. \end{aligned}$$

The optimization problem above is difficult to solve as it depends on $\|w\|$ (e.g., the norm of) that involves a square root. Note that by substituting $\|w\|$ in the objective with $\frac{\|w\|^2}{2}$, the solution remains unchanged as the minimum of the original and the modified equation both have the same w and b. Hence, we obtain the equivalent quadratic programming (QP) optimization problem as follows:

$$\begin{aligned} &\min \quad \frac{1}{2}\|w\|^2 \\ &\text{s.t.} \quad y_i(w.x_i - b) \geqslant 1, \quad i = 1, \ldots, N. \end{aligned}$$

Unfortunately, for linearly non-separable cases, a hyperplane that correctly classifies every training point does not exist. We therefore generalize the optimization idea above by introducing the concept of a soft margin and obtain the following new optimization problem,

$$\begin{aligned} &\min \quad \frac{1}{2}\|w\|^2 + h\sum_{i=1}^{N}\tau(i) \\ &\text{s.t.} \quad y_i(w.x_i - b) \geqslant 1 - \tau(i), \quad i = 1, \ldots, N, \end{aligned}$$

where $\tau(i)$ are called slack variables which are related to the soft margin, and h is the tuning parameter used to balance the margin and the training error. In the classification phase, a point x is assigned a label y according to $y = sgn[w \cdot x + b]$.

3.2. Relevance and redundancy based on mutual information

Contrary to the intuitive interpretation, including more features in a classification model does not necessarily provide more discriminating power. Furthermore, additional features may induce some disadvantageous effects on the classification process. Firstly, they significantly slow down the learning process. Secondly, they deteriorate the classification accuracy by causing the classifier to over-fit the training data as irrelevant or redundant features may confound the learning algorithm.

As outlined in [59], the features of a dataset can be considered to fall into one of three different categories: strongly relevant features, weakly relevant features and irrelevant features. While the strongly relevant features must be included in the optimal subset, the weakly relevant features are not always necessary but may become necessary for an optimal subset at certain conditions. To determine the relevance properties of the feature space, the mutual information concept is first introduced in [3]. Given two random variables x and y, their mutual information $I(x,y)$ is defined in terms of their probability density functions $p(x)$, $p(y)$ and $p(x,y)$:

$$I(x,y) = \int\int \frac{p(x,y)\log(p(x,y))}{p(x)p(y)}\,dx\,dy, \tag{1}$$

$$I(x,y) = \sum_{x\in X}\sum_{y\in Y} \frac{p(x,y)\log(p(x,y))}{p(x)p(y)}, \tag{2}$$

where (1) and (2) are for continuous and discrete cases, respectively.

In the case of discrete x and y, it is easy to calculate $I(x,y)$. However, when at least one of the variables is continuous, it becomes difficult to compute their mutual information. To overcome this problem, a data discretization method needs to be incorporated in the process. A density estimation method such as Parzen window (e.g., kernel density estimation) is one of the commonly used alternatives. Parzen window is a non-parametric way of estimating the probability density function of a random variable.

For some datasets in our experimentation, we used the Parzen window to estimate the densities. Parzen method requires two important definitions: window (kernel function) and window width (bandwidth). Let R be a hypercube centered at x where the length of the edge of the hypercube is denoted by h, called bandwidth. Hence, the volume V is defined as $V = h^2$ for a 2-dimensional square, and $V = h^2$ for a 3-dimensional cube and so forth. The kernel function characterizes the local probability density function around each observation. While there is a variety of kernel function alternatives (e.g., Gaussian), we chose to use the uniform density kernel function in our implementations.

Given a set of observations $\{x_\perp, i = 1,\ldots,n\}$, let $\gamma(\cdot)$ denote the window function with bandwidth h,

$$\gamma\left(\frac{(x_i - x)}{h}\right) = \begin{cases} 1, & \frac{|x_{ik} - x_k|}{h} \leqslant \frac{1}{2}, \\ 0, & \text{otherwise} \quad k = 1,\ldots,D. \end{cases}$$

Further, k, the total number observations falling within the region R is expressed as,

$$k = \sum_{i=1}^{n} \gamma\left(\frac{(x_i - x)}{h}\right).$$

Then the kernel density approximation of the probability density function of x in 2-dimensions is calculated as follows:

$$p(x) = \frac{k/n}{V} = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h^2}\gamma\left(\frac{(x_i - x)}{h}\right)$$

Peng et al. [40] propose relevance and redundancy criteria to determine the information property of a feature subset. In particular, they defined the relevance of a feature subset S as $R = \frac{1}{|S|}\sum_{x_i \in S} I(x_i, C)$, which is the mean value of all mutual information values between individual features $x_i \in S$ and the target class c. When the features are selected such that the relevance R is maximized, it is possible to have high dependency (i.e., redundancy) among these features. Given two highly dependent features, removing one of them from the set S would not change the class-discriminative power. Hence, the redundancy of a feature subset S is defined as $B = \frac{1}{|S|^2}\sum_{x_i x_j \in S} I(x_i, x_j)$. They further state that the purpose of feature selection is finding a feature set S with d features $\{x_i\}$ that either jointly have the largest dependency on the target class c or have the minimal redundancy in the selected subset S. Consequently, this leads to a bi-criteria feature subset selection objective. They recommend searching balanced solutions through the composite objective $\max Z(R, B) = R - B$. This criterion combining the two criteria is called "*maximal relevance minimal redundancy*" criterion.

In contrast to [40], our goal is to maximize the prediction accuracy of the selected feature subset. Hence, we use the relevance and redundancy mutual information only as an intermediate measure in the PSO algorithm to improve the speed and performance of the search. For this, we define a "*relevance–redundancy index*", φ_j, for feature j based on the composite objective Z, as discussed in the next section.

3.3. Particle swarm optimization algorithm

Particle swarm optimization (PSO) is a population-based search technique and motivated by the social behavior of organisms such as bird flocking and fish schooling. It is originally proposed by Kennedy and Eberhart [22] for continuous problems and then was extended to discrete problems by Kennedy and Eberhart [23]. It is well suited for combinatorial optimization problems in which the optimization surface possesses many local optimal solutions. The underlying phenomenon of PSO is that knowledge is optimized by social interaction and thinking is not only personal but also social. The particles in PSO resemble the chromosomes in genetic algorithm. However, PSO is usually easier to implement than the GA as there are neither crossover nor mutation operators in the PSO and the movement from one solution set to another is achieved through the velocity functions. We refer the reader to [41] for a recent review of the applications and variations of the PSO.

PSO is based on the principle that each solution can be represented as a particle in a swarm. Each particle has a position and a corresponding fitness value evaluated by the fitness function to be optimized. The particles iterate (fly) from one position to another according to their most recent velocity vector. This velocity vector is determined according to the particle's own experience as well as the experience of other particles by using the best positions encountered by the particle and the swarm. Specifically, the velocity vector of each particle is calculated by updating the previous velocity by following two best values. The first best value is the particle's personal best value (*pbest*) (i.e., the best position it has visited thus far) and is tracked by each particle. The other best value is tracked by the swarm and corresponds to the best position visited by any particle in the population. This best value is called the global best (*gbest*). The effect of personal best and global best on the velocity update is controlled by weights called learning factors. Through the joint self and swarm-based updating, the PSO achieves local and global search capabilities where the intensification and diversification are achieved via relative weighting.

We denote the number of particles with N and refer to each particle with index i, i.e., $i = 1,2,\ldots,N$. Let $X_i^t = \left(x_{i1}^t, x_{i2}^t, \ldots, x_{ij}^t, \ldots, x_{ik'}\right)$ denote the position vector of particle i at iteration t, where the dimension of the particle is the number of features (K) and $x_{ij}^t \in \{0,1\}$. Accordingly, $(X_1^t, X_1^t, \ldots, X_N^t)$ represents the swarm of the particles at iteration t. Let P_i^t denote the personal best position for particle i at iteration t, where $P_i^t = \left(p_{i1}^t, p_{i2}^t, \ldots, p_{ij}^t, \ldots, p_{ik}^t\right)$ and $p_{ij}^t \in \{0,1\}$. In addition,

G^t denotes the global best position for the swarm at iteration t, where $G^t = \left(g_1^t, g_2^t, \ldots, g_j^t, \ldots, g_k^t\right)$ and $g_j^t \in \{0,1\}$. There are two key differences between the discrete and continuous versions of PSO. The first difference is the representation of the particle. In the discrete PSO, every particle is expressed as a binary vector. The second difference is that the velocity of a particle in the discrete PSO is a probability vector, where each probability element determines the likelihood of that binary variable taking a value of one. At the end of each discrete PSO iteration t, the velocity vector of particle i, v_i^{t+1}, is updated as follows:

$$v_i^{t+1} = w v_i^t + c_1 r_1 \left(P_i^t - X_i^t\right) + c_2 r_2 \left(G^t - X_i^t\right), \tag{3}$$

where $v_i^t = \left(v_{i1}^t, v_{i2}^t, \ldots, v_{ij}^t, \ldots, v_{ik^t}\right)$ is the previous iteration's velocity vector, w is the inertia weight, c_1 is the weight factor for local best solution, c_1 is the weight factor for global best solution factor, and r_1 and r_2 are random numbers uniformly distributed in [0,1]. The terms in (3) represent the memory, cognitive learning and social learning of the particle, respectively. The weights (c_1, c_2) are referred as the learning rates since the inertia weight controls the extent to which the memory of the previous velocity influences the new velocity. The pseudo-code for the proposed adaptive PSO algorithm for solving the feature subset selection problem is provided at the end of this section.

There are usually maximum and minimum velocity levels, v_{max} and v_{min}, defined to bound the velocity v_i^{t+1}. If the velocity v_i^{t+1} in (3) exceeds v_{max}, then $v_i^{t+1} \to v_{max}$, or if it is less than v_{min}, then $v_i^{t+1} \to v_{min}$ [6]. The diversification and intensification of the particle is controlled through these velocity bounds as well as the inertia weight [46]. Inertia weight, velocity bounds and learning rates jointly determine the particle's motion. Usually, a high inertia weight is used at the beginning and then gradually decreased to diversify the solution particles. Specifically, at each iteration of the PSO algorithm, the inertia weight w is updated according to the following expression:

$$w^{t+1} = w_{max} - \frac{(w_{max - w_{min}})}{T} t, \tag{4}$$

where w_{max} and w_{min} are the bounds on the inertia weight and T is the maximum number of PSO iterations.

In our application of the PSO to the feature subset selection problem, we define the position of a particle as the binary vector $X_i^t = \left(x_{i1}^t, x_{i2}^t, \ldots, x_{ij}^t, \ldots, x_{ik}^t\right)$ where $x_{ij}^t = 1$ if feature j is to be included in the feature subset, and 0 otherwise. Accordingly, K represents the total number of features in the original data set. Note that for a given number of features to be included in the subset, $k \leqslant K$, we have $\sum_j x_{ij}^t = k$ for $\forall i, t$.

While in the continuous PSO the position of the particle is updated as below;

$$X_i^{t+1} = X_i^t + v_i^{t+1},$$

in the discrete PSO, we first transform the velocity vector into a probability vector through a sigmoid function,

$$s_{ij}^t = \frac{1}{1 + e^{v_{ij}^t}}, \tag{5}$$

where s_{ij}^t represents the probability that the jth bit in X_i^t is 1. Hence, the position of the particle in the discrete PSO is updated as follows,

$$x_{ij}^t = \begin{cases} 1, & \text{if } \delta < s_{ij}^t, \\ 0, & \text{otherwise}, \end{cases} \quad j = 1, \ldots, K, \tag{6}$$

where δ is a uniform random number between 0 and 1.

When the position of a particle is updated as in (6), the inclusion/exclusion decisions of features are made independent of one another. However, it is well known that features possess statistical dependencies (e.g., redundancy) as well as exhibit subset dependent prediction contributions. Accordingly, random selection of features leads to such instances where redundant features are selected or there is a lower predictive contribution [26,59]. Hence, an ideal feature subset selection strategy would select features according not only to their independent likelihood (s_{ij}^t) but also based on their contribution to the subset of features already selected. This can be achieved by admitting features in the feature subset one at a time according to the contribution weighted probability of features. One way to calculate the contribution is to calculate the predictive contribution of the feature j to the current feature subset L_i^t of particle i in iteration t by calculating $max\left[J\left(L_i^t \cup \{j\}\right) - J\left(L_i^t\right), 0\right]$. This procedure requires successively constructing the particle's position by incrementally adding the features into the subset L_i^t. However, since the predictive contribution calculations need to be done for each unselected feature $(j \notin L_i^t)$ and until $|L_i^t| = d$, the computational effort necessary for classification criterion function $J(\cdot)$ is prohibitive. Alternatively, in our hybrid PSO algorithm for feature selection, we take advantage of the efficiency of filter type methods, which provide valuable redundancy and relevance information for each candidate feature to be included in the subset.

In particular, we use the "*relevance-redundancy index*" for feature j, φ_j, defined in the preceding section. For each particle i, we first initialize the feature subset $L_i^t = \emptyset$ and then select the first feature according to feature's independent probabilities (e.g., s_{ij}). For the remaining features, we first calculate the redundancy and relevance index φ_{ij} for each candidate feature $j \in F \backslash L_i$ via the following expression:

$$\varphi_{ij}^t = I(x_j, c) - \frac{1}{L_i^t} \sum_{x_l \in L_i^t} I(x_j, x_l), j \in F \setminus L_i^t \tag{7}$$

Next, we calculate the redundancy-relevance index weighted probabilities, $\Psi_{ij}^t = s_{ij}^t \varphi_{ij}^t$ for $j \in F \setminus L_i^t$ and apply the random proportional rule using Ψ_{ij}^t for $\forall j \in F \setminus L_i^t$ to select the next feature to be included in the subset. The algorithm terminates once the required number of features are selected (i.e.,$|L_i^t| = d$, $|L_i^t| = d$).

A commonly occurring behavior in the binary discrete PSO is when a feature's bits in $X_i^t, P_i^t, and\ G^t$ all have the same value (i.e., either 0 or 1). This may lead to an event where the probability that the feature will be included (or excluded) is *0.5*. For small problems, where the binary vector length is small compared to the number of bits allowed to be 1, this event improves the diversification. For large problems, however, this event causes excessive diversification as a result of single particle movement. Therefore, we modified the social learning in the velocity update (3) by using two best neighbors: global best and the iteration best. The iteration best, *ibest*, is the position of the best particle at each iteration. Accordingly, we use the following velocity update formula:

$$v_i^{t+1} = w v_i^t + c_1 r_1 \left(P_i^t - X_i^t\right) + c_2 r_2 \left(G^t - X_i^t\right) + c_3 r_3 \left(I^t - X_i^t\right), \tag{8}$$

where c_3 is the weight factor for the best solution in iteration t, r_3 is a random number uniformly distributed in $[0,1]$, and $I^t = \left(i_1^t, i_2^t, \ldots, i_j^t, \ldots, i_k^t\right)$ is the position of the best particle in iteration t among all particles. The pseudo-code for the proposed hybrid algorithm (*mr²PSO*) based on feature relevance and redundancy filter and PSO wrapper follows:

mr²PSO-hybrid PSO algorithm for feature selection:

Initialize

- Set parameters: c_1, c_2, c_3, ω_{min}, ω_{max}, v_{min}, v_{max}
- Initialize $L_i^1 = \emptyset,\ M = \emptyset,\ t = 1,\ \omega = \omega_{max},\ v_i^1 = 0\ \forall i = 1, 2, \ldots, N$
- Initialize particles X_i^t for $i = 1,2,\ldots,N$ randomly such that $\sum_j x_{ij}^1 = k$ for $\forall i = 1, 2, \ldots, N$
- Set $P_i^t = X_i^1$ and determine G_i^1 and I^1
  i. $lbest_i = J\left(L_i^1\right)$ where $j \in L_i^1$ if $x_{ij}^1 = 1\ \forall i = 1, 2, \ldots, N$
  ii. $G^1 = argmax_{x_i^1}\{lbest_i\}$ and $gbest = J(L)$ where $j \in L$ if $g_j^1 = 1$
  iii. $I^1 = argmax_{x_i^1}\{lbest_i\}$ and $ibest = J(L)$ where $j \in L$ if $i_j^1 = 1$

Repeat, While $t \leqslant T$

$t = t + 1$; Update ω using (4)

For each particle $i = 1,2,\ldots,N$, Repeat.

Calculate velocity v_i^t using (8)

Determine the particle*i*'s position X_i^t:

- Set $L_i^t = \emptyset$
- Repeat, while $|L_i i\dagger t| < k$
  - For each $j \in F \setminus L_i^t$, Repeat
    - Calculate independent selection probability s_{ij}^t using (5)
    - Calculate the relevance-redundancy index φ_{ij}^t using (7)
    - Calculate $\Psi_{ij}^t = s_{ij}^t \times \varphi_{ij}^t$
  - Apply random proportional rule in set $F \setminus L_i^t$
    - Generate a uniform random number $\delta \in [0,1]$
    - Select a feature $f \in F \setminus L_i^t$ based on δ and $x_{ij}^t = 1$
    - Update $L_i^t \to L_i^t \cup \{f\}$

Calculate feature subset selection criterion $J\left(L_i^t\right)$ where $j \in L_i^t$ if $x_{ij}^t = 1$

Update $lbest_i$ if $lbest_i < J\left(L_i^t\right)$ where $j \in L_i^t$ if $x_{ij}^t = 1$

Update *gbest* and *ibest*

- If $gbest < argmax_{x_i^t}\{lbest_i\}$
  - $G_i^t = argmax_{x_i^t}\{lbest\}$ and $gbest = J(L)$ where $j \in L$ if $g_j^t = 1$
- If $ibest < argmax_{x_i^t}\{J(L_i) : j \in L_i \text{ if } x_{ij}^t = 1\}$,
  - $I^t = argmax_{x_i^t}\{J(L_i) : j \in L_i \text{ if } x_{ij}^t = 1\}$ $ibest = J(L)$ where $j \in L$ if $i_j^t = 1$

Terminate with the feature subset L where $j \in L$ if $g_j^t = 1$.

The random proportional rule used in the adaptive feature subset selection procedure is defined as follows:

Definition (*Random proportional rule*). The random proportional rule for selecting a subset Γ of a set Ω based on a uniform random number vector $\delta = \{\delta_k \in [0,1],\ k = 1,\ldots,|\Gamma|\}$ and member values $\theta = \{\theta_j,\ j \in \Omega\}$ is as follows:

- Create a vector of normalized member values of set Ω, i.e.$\hat{\theta} = \{\hat{\theta}_j, j \in \Omega\} \cup \{0, 1\}$, where $\hat{\theta}_j = \theta_j / \sum_{k\in\Omega}\theta_k$, $\hat{\theta}_0 = 0$, and $\hat{\theta}_{|\omega|+1} = 1$.
- Select member$j \in \Omega$ to be included in set Γ, if $\delta_k \in [\hat{\theta}_{j-1}, \hat{\theta}_j$ for any $k = 1, \ldots, |\Gamma|$.

In our implementation of the maximum relevance minimum redundancy PSO, we follow two types of algorithmic configurations. First, we take the number of features to be selected as a pre-specified parameter d and then search for the best d features out of N in the search space. The second configuration searches for the best compromise solution that maximizes accuracy/mutual information while minimizing the number of features. Accordingly, the corresponding d with respect to maximum fitness value is regarded as the optimal number of features.

4. Experimental studies

4.1. Experimental setup

In this section, we present the results from a series of experiments carried out to test and compare the proposed method. Six real-world datasets used in the experimentation were all obtained from the well-known data repository of the University of California (UCI) Machine Learning Repository [37]. Each of the datasets selected has enough data instances for every degree of freedom to avoid the risk of over-fitting. The following datasets were used in our experiments;

- *Glass* (GL): The study of classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence if it is correctly identified. There are 214 instances in this dataset defined by 10 attributes, each instance belonging to one of seven classes.
- *Wine* (WN): These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. There are 178 instances in this dataset.
- *Wisconsin Breast Cancer-Diagnostic* (BC): This is medical dataset. Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. There are 569 instances with 30 real valued attributes. Each instance has one of two possible diagnosis classes: benign or malignant.
- *Ionosphere* (IO): In this dataset there are 34 continuous attributes and 351 instances, each belonging to one of two classes. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those with signals that pass through the ionosphere.
- *Sonar* (SO): This database contains 208 patterns, each being a set of 60 numbers in the range 0.0 to 1.0. Each number represents the energy within a particular frequency band, integrated over a certain period of time. There are two distinct class labels in this database. The label associated with each record contains the letter "R" if the object is a rock and "M" if it is a mine.
- *Heart* (HE): This database contains 267 instances with 76 attributes, but none of the published experiments refer to using all attributes. We have considered only 13 attributes from the heart dataset. Although there are five classes in this database, experiments have concentrated on simply attempting to distinguish the presence (values 1, 2, 3, 4) from the absence (value 0) of heart disease.

Wrapper type feature selection algorithms are very dependent on classifier, re-sampling methods, and even on the parameter settings of the algorithm. To make fair comparisons, we set the parameters of the SVM classifier the same as in [20], such that our feature selection algorithm uses SVM classifier with error cost parameter of 100 and Radial Basis Function with $\sigma = 2$ as the kernel. When the number of classes exceeds 2 we used *one-against-rest* strategy. The PSO parameter setting is such that maximum and minimum inertia weight, ω_{max} and ω_{min} are 0.9 and 0.4, respectively, learning rates, $c_1 = c_2 = 2$, maximum and minimum velocity allowed $v_{max} = 6$ and $v_{min} = -6$, and the maximum number of iterations was $t_{max} = 300$. We selected this setting of parameters after several parameter tuning experiments. Obviously, the selection of best set of parameters is a challenging task. However, there are a number of earlier empirical and theoretical studies on the PSO parameter selection which guided us in the parameter tuning process [22,6,46,50]. The results are compared in terms of three performance measures presented in [20]. *2-Fold cross-validation* is used as re-sampling method for all experiments. We implemented the maximum relevance minimum redundancy feature selection for the SVM classification algorithm on a PC with Intel DualCore CPU, 6400 at 2.13 GHz and 2 GB RAM. Finally, we run our algorithm 10 times for all datasets and always report average performance unless otherwise stated.

4.2. Results and discussion

We performed two types of experiments to evaluate the performance of the proposed method. In the first set of experiments, we considered the reported d values in [20] as prespecified (i.e., fixed the number of features to be selected) and searched for the best d features out of N available features. Classification accuracy is taken as the fitness function in this first set of experiments. In the second set of experiments, our algorithm searched for the best set of features that maximize fitness

value (either classification accuracy or mutual information, as required) while minimizing the number of features. The corresponding d is accepted as the optimal number of features. More details regarding these experiments are presented in the following sections.

4.2.1. Selection of best d features – case of predetermined feature set size

Table 1 shows, for each dataset, the predetermined feature set size (d), dimensionality reduction percentage (r), and the mean performances of HGA_p method in [20] and the proposed method mr^2PSO_{ACC} (including the standard deviations in performance). The performance measures reported are mutual (output) information (MI), Kappa statistic (KS), and classification (estimation) accuracy (ACC). Note that the mutual information performance measure is calculated between actual and predicted class values (e.g., Y and Y_f). All results reported for the proposed method are based on 10 runs. As stated above, the criterion for search by the proposed method is accuracy. To make a fair comparison between algorithms, we restricted the feature subset sizes d to those recommended by Huang et al. [20] during this set of experiments. Our aim in this set of experiments was to compare the proposed hybrid PSO based filter–wrapper approach to the HGA_p method.

When we compare the performances of the two algorithms according to the three criteria, the proposed mr^2PSO_{ACC} method outperforms HGA_p in terms of mutual information for three out of six datasets, for five out of six datasets in terms of Kappa statistics, and for all datasets in terms of classification accuracy. Since we have used different objective functions when searching for the best feature subset, we consider the Kappa statistic a better measure in making a fair comparison.

4.2.2. Selection of best features – case of optimal feature set size

Different than the experiments in Section 4.2.1, we herein utilize the proposed mr^2PSO to search for the least number of features that achieve the highest fitness value. In searching for the best features, we employed two criteria, classification accuracy (ACC) and MI, denoting the respective proposed methods as mr^2PSO_{ACC} and mr^2PSO_{MI}. Figs. 2 and 3 report the run time behavior from a particular run of mr^2PSO_{ACC} and mr^2PSO_{MI} algorithms for Sonar dataset, respectively. Note that we observed similar results for other datasets, however, we are not reporting them for brevity. The plots in Figs. 2(a) and 3(b) clearly indicate that there is monotone performance improvement during the search, a desirable characteristic. These two plots correspond to the respective search criteria (e.g., accuracy and mutual information), where the search is guided

Table 1
Performance comparison between HGA_p and mr^2PSO_{ACC} under the case of predetermined feature set size.

	d	r	MI[$I(Y,Yf)$]		KS		ACC	
			HGA_p	mr^2PSO_{ACC}	HGA_p	mr^2PSO_{ACC}	HGA_p	mr^2PSO_{ACC}
GL	5	0.4444	**0.8208**	0.2083 ± 0.067	0.5213	**0.7433 ± 0.024**	0.6551 ± 0.024	**0.7977 ± 0.020**
WN	6	0.5385	**1.4628**	0.9000 ± 0.027	0.9711	**0.9964 ± 0.004**	0.9831 ± 0.006	**0.9972 ± 0.003**
BC	3	0.9000	0.6353	**0.9565 ± 0.004**	**0.8755**	0.7665 ± 0.032	0.9424 ± 0.004	**0.9664 ± 0.003**
IO	6	0.8235	0.5693	**0.6719 ± 0.071**	0.8393	**0.9347 ± 0.006**	0.9276 ± 0.011	**0.9492 ± 0.004**
SO	15	0.7500	0.4461	**0.5004 ± 0.083**	0.7373	**0.8435 ± 0.017**	0.8702 ± 0.000	**0.8815 ± 0.013**
HE	3	0.7692	**0.3266**	0.2948 ± 0.047	0.6449	**0.7702 ± 0.023**	0.8259 ± 0.021	**0.8267 ± 0.017**

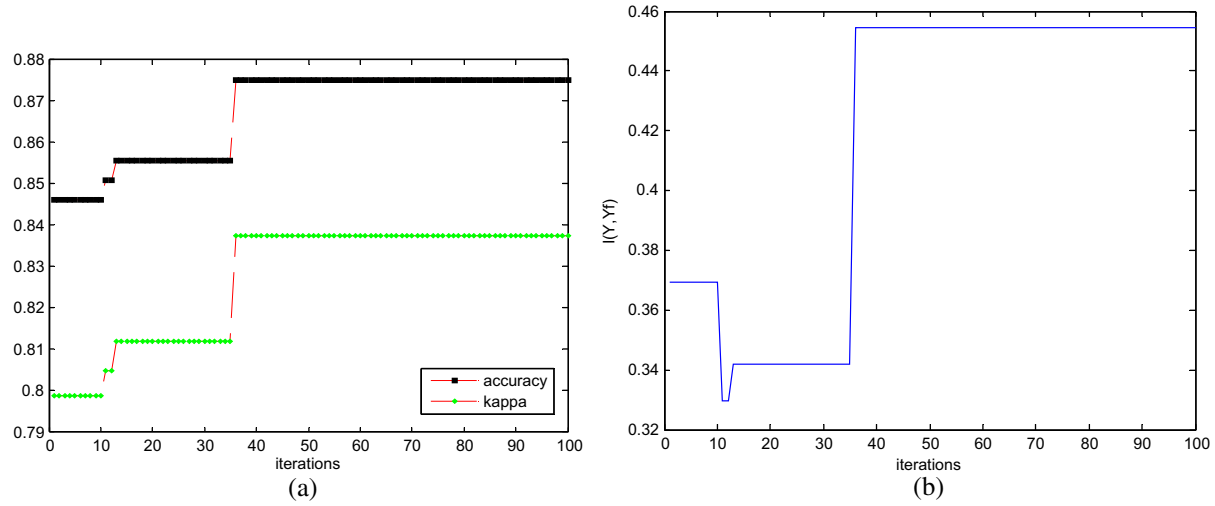


Fig. 2. Runtime behavior of mr^2PSO_{ACC} for a particular run of the Sonar dataset: (a) classification accuracy and Kappa statistic performance by iteration and (b) mutual Information performance by iteration.

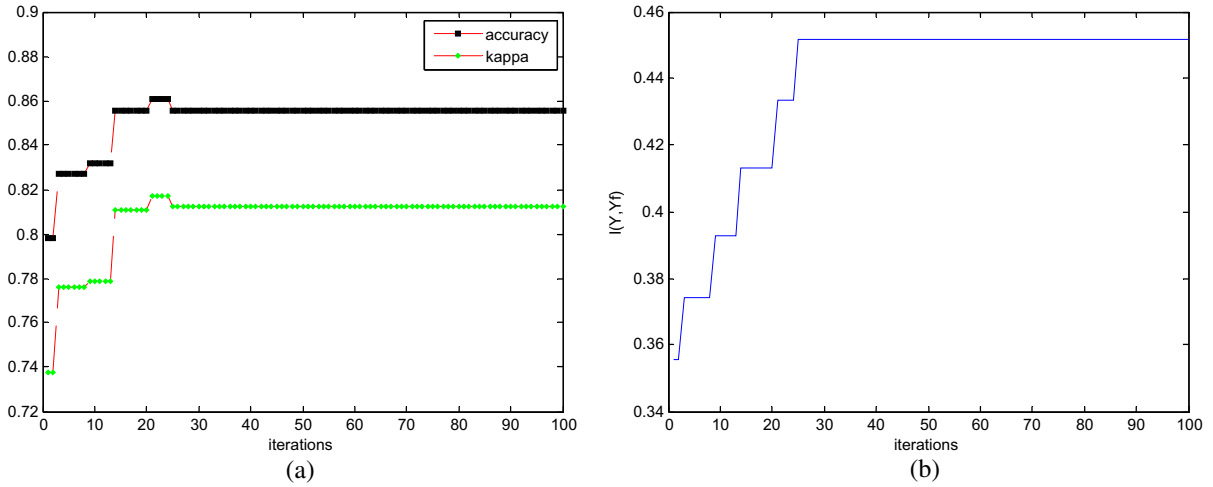


Fig. 3. Runtime behavior of mr^2PSO_{MI} for a particular run of the Sonar dataset: (a) classification accuracy and Kappa statistic performance by iteration and (b) mutual information performance by iteration.

with these criteria. This is not the case for Figs. 2(b) and 3(a), where there are some non-monotonic behaviors. These non-monotonic behaviors in trajectories in Figs. 2(a) and 3(b) could be attributed to a lack of a symmetrical relationship between classification accuracy and mutual information, at least for this dataset. However, once the algorithm converges, we see agreement between accuracy and MI (i.e., they both peak in the end) in both Figs. 2 and 3, which is also predicted in [20]. In particular, authors in [20] conclude that when a classifier is trained by the objective function of minimal classification error rate, its output information (mutual information) also achieves its maximum in the end.

Table 2 shows the results for the second set of experiments comparing HGA_p and mr^2PSO where we executed the proposed mr^2PSO with two feature subset selection criteria, classification accuracy (*ACC*) and MI. At first glance, it seems that nearly for all cases, HGA_p reduces dimensionality better than the proposed algorithm for both criteria. However, the proposed method mr^2PSO significantly dominates HGA_p in mutual information for four out of six datasets and dominates in accuracy for five out of six datasets. For example, in the case of glass dataset, with barely 0.9 additional features on the average, it lifts average classification accuracy from 65.5% to over 80%. The only time HGA_p outperforms the proposed method in terms of accuracy is in the case of Sonar dataset, where the performance is slightly better than the proposed method (with 1.35% better accuracy over mr^2PSO_{ACC}) with 1.1 additional features on the average. The results from the table seem to indicate that HGA_p can terminate prematurely by getting trapped in local optimal solutions.

Table 2 also shows that the classification accuracy seems to be a better fitness measure than the mutual information when searching for best feature subsets. For example, mr^2PSO_{ACC} dominates mr^2PSO_{MI} in all experiments in terms of KS statistic. As for the size of the recommended feature set, the difference in mr^2PSO_{ACC} and mr^2PSO_{MI} recommended set sizes are statistically insignificant. In contrast to HGA_p, it appears that the proposed methods (e.g., mr^2PSO_{ACC} and mr^2PSO_{MI}) are far more effective in locating the optimal feature subset, namely the feature subset with peak performance. In theory, having more features implies more classification accuracy. However, in reality, this is not always true due to data sparsity issues, noise, and other factors. For example, some features may not represent the underlying phenomena of interest, but their introduction increases model complexity and, hence, can compromise performance. This can lead the so called peaking effect in feature selection problem domain (i.e., performance can peak for feature subsets) as can be seen in Fig. 4.

We now demonstrate the performance of the proposed method over different feature subset sizes using heart and wine datasets (Figs. 4–6). In these experiments, we use the mr^2PSO_{ACC} to search for the subset of a given size with the best classification accuracy (*ACC*). Note that since we observed similar results for other datasets, we are not reporting them for brevity. Fig. 4 plots the average classification accuracy as a function of feature subset size for heart and wine datasets. A very desirable trend is apparent from the plots. The classification accuracy grows approximately monotonic as a function of feature subset size until peak performance is achieved. This allows termination of search upon peaking with confidence. Results from Table 2 suggest that the search strategies in HGA_p do not offer this property since the optimal d values are mostly smaller than those suggested by mr^2PSO. Unfortunately, Huang et al. [20] does not offer these plots for a direct comparison.

Obviously, the number of features to be selected ultimately depends on the trade-off between model compactness, execution speed, and desired accuracy. For example, if the classification accuracy is the most important factor, then Fig. 4(b) indicates that approximately eight features will be needed for the Heart dataset. However, if the threshold value for classification accuracy is 80%, then three features would be adequate.

Fig. 5 is very similar to Fig. 4, except it plots Kappa statistic performance as a function of the number of selected features. The plot patterns are in close resemblance to those in Fig. 4 such that the peaks are aligned and trajectories look alike. This is

Table 2
Performance comparison between HGA_p, mr^2PSO_{ACC} and mr^2PSO_{MI} with d unrestricted.

	HGA_p	d		MI[$I(Y,Yf)$]			KS			ACC		
		mr^2PSO_{ACC}	mr^2PSO_{MI}	HGA_p	mr^2PSO_{ACC}	mr^2PSO_{MI}	HGA_p	mr^2PSO_{ACC}	mr^2PSO_{MI}	HGA_p	mr^2PSO_{ACC}	mr^2PSO_{MI}
GL	5	5.9 ± 1.2	4.9 ± 0.8	**0.8208**	0.2264 ± 0.062	0.2402 ± 0.088	0.5213	**0.7483** ± 0.023	0.7287 ± 0.048	0.655 1 ± 0.024	**0.8028 ± 0.019**	0.7850 ± 0.039
WN	6	8.6 ± 1.6	8.3 ± 2.12	**1.4628**	0.8963 ± 0.035	0.9165 ± 0.004	0.9711	**0.9964 ± 0.003**	0.9896 ± 0.005	0.983 1 ± 0.006	**0.9972 ± 0.003**	0.9919 ± 0.004
BC	3	12.2 ± 2.3	11.1 ± 1.9	0.6353	0.7657 ± 0.041	**0.8033 ± 0.039**	0.8755	**0.9695 ± 0.001**	0.9588 ± 0.002	0.9424 ± 0.004	**0.9766 ± 0.001**	0.9683 ± 0.001
IO	6	9.25 ± 0.7	9.6 ± 0.8	0.5693	0.7123 ± 0.054	**0.7287 ± 0.046**	0.8393	**0.9427 ± 0.005**	0.9413 ± 0.006	0.9276 ± 0.011	**0.9554 ± 0.004**	0.9544 ± 0.004
SO	15	13.9 ± 2.8	14.3 ± 3.0	0.4461	0.3969 ± 0.078	**0.4853 ± 0.048**	0.7373	**0.8117 ± 0.023**	0.7936 ± 0.019	**0.8702 ± 0.000**	0.8567 ± 0.017	0.8428 ± 0.013
HE	3	7.5 ± 1.5	8.6 ± 1.34	0.3266	0.3879 ± 0.063	**0.4271 ± 0.059**	0.6449	**0.8150 ± 0.009**	0.7922 ± 0.031	0.825 9 ± 0.021	**0.8601 ± 0.007**	0.8430 ± 0.023

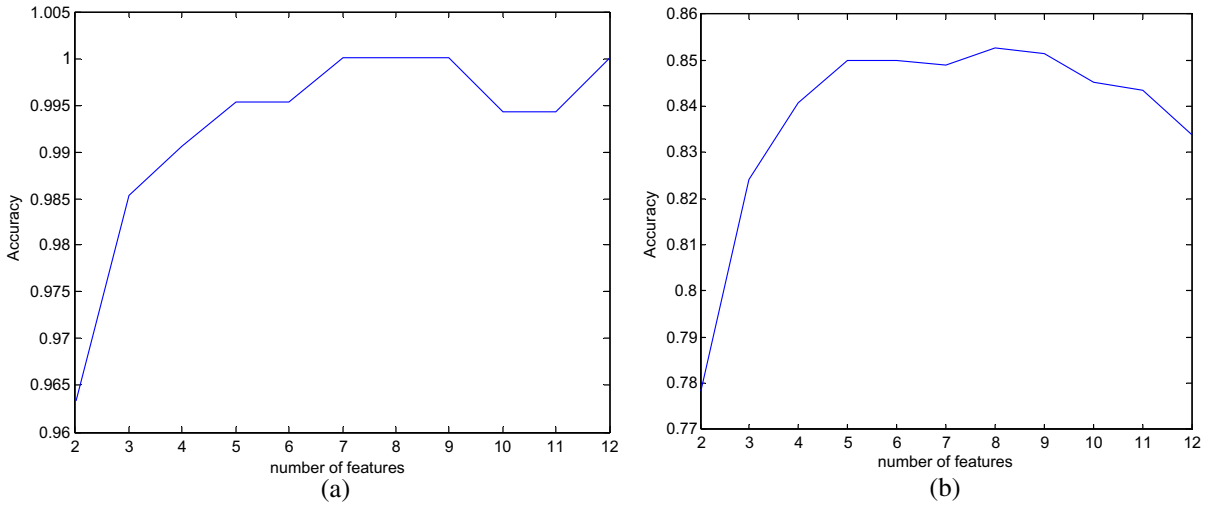


Fig. 4. Variation of classification accuracy as a function of the number of selected features during a particular run: (a) wine dataset and (b) heart dataset.

because both measures take into account the number of correct classifications but from different perspectives. Fig. 6 shows the behavior of mutual information as a function of the number of selected features for the two datasets. Although the classification accuracy and Kappa statistic plot characteristics (peaks and trajectories) are very similar, it is not the case with the mutual information. Since we used the mr^2PSO_{ACC} to search for the subset with the best classification accuracy, the dissimilarity of MI plots in Fig. 6 with those in Figs. 4 and 5 are expected. Interestingly, one can observe that the MI plot is not monotonic for the heart dataset and peaks at about 6 features, with MI = 0.39. In comparison with the result in Table 2, the peak MI is attained with 8–9 features, with MI = 0.427 on the average.

Fig. 7 plots the commonality of features selected during 10 different runs of the search algorithm (mr^2PSO_{ACC}) for wine and heart datasets. Note that these runs are with optimal feature set size (e.g., at peak accuracy performance). Overall, in both cases, all of the features are selected in one run or the other. This confirms that the search for the optimal feature subset is strongly sample dependent. For example, in heart dataset, the features 5, 6 or 7 are included in the optimal subset only two out of 10 times. This can be interpreted as some features in the feature space can be predictive only when combined with some other specific features. As mentioned in [40], the d best features are not always the best d features. On the other hand, some features are very dominant, which means they are highly relevant to the class labels and have a great chance to be selected during the search process. For example, note the consistent presence of first and last features of the wine dataset in the selected feature subset and features 3, 12 and 13 of the heart dataset.

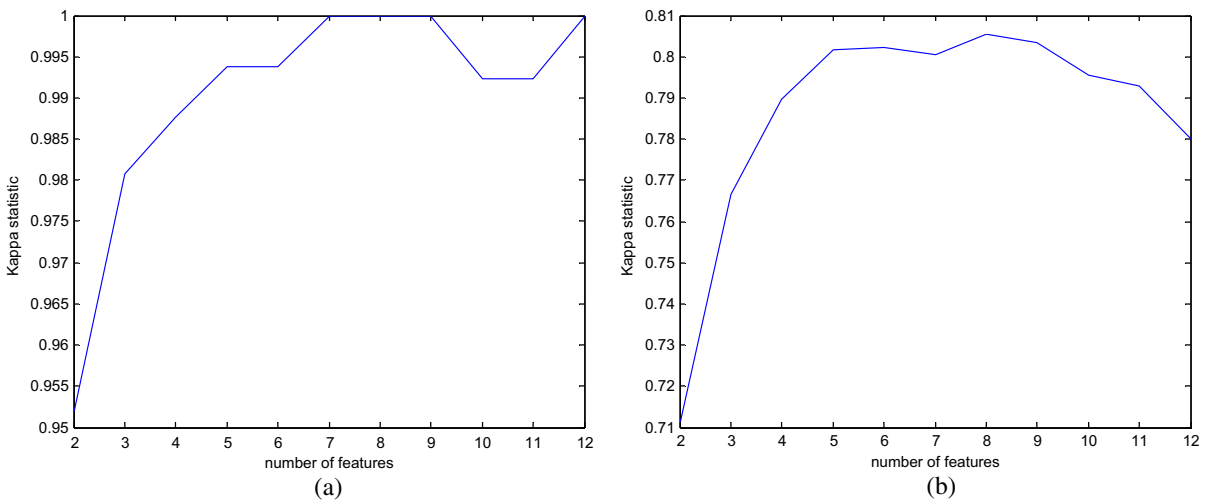


Fig. 5. Variation of Kappa statistic as a function of the number of selected features during a particular run: (a) wine dataset and (b) heart dataset.

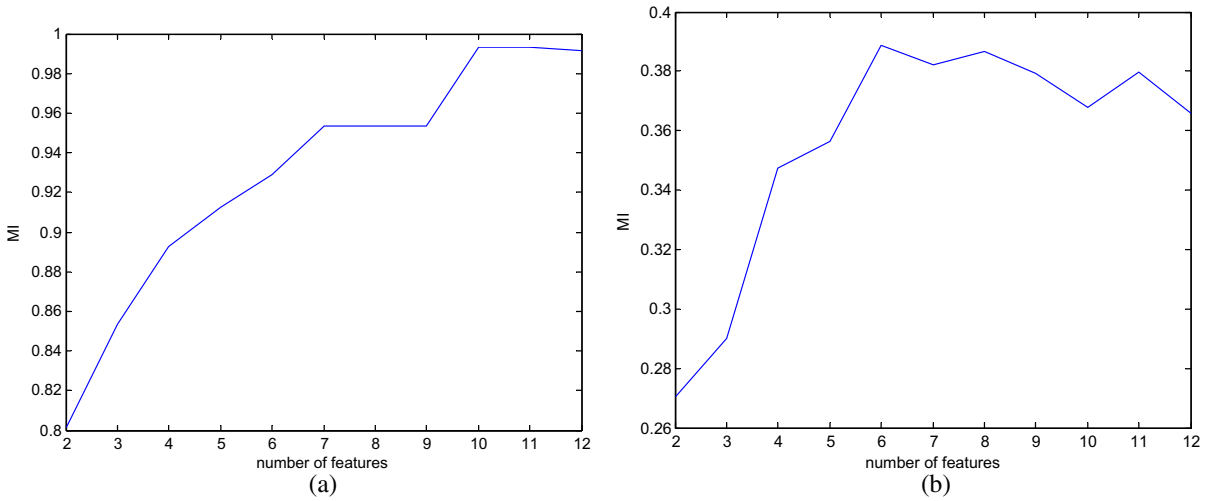


Fig. 6. Variation of MI (output information) as a function of the number of selected features during a particular run: (a) wine dataset and (b) heart dataset.

In Fig. 8, we plot the effect of hybridization in terms of time and performance over PSO wrapper using the sonar and heart datasets. It is clear that the hybridization of PSO wrapper with mutual information based filter (e.g., giving redundancy-relevance weights to candidate features in the search) increases both the accuracy performance and computational efficiency. In particular, the hybrid method finds feature subsets with better classification accuracy in far less iterations. The effect of hybridization is more pronounced when the original feature set is larger. For example, the total number of features in the Sonar dataset is 60 and it is clearly seen in Fig. 8(a) that *mr²PSO* has significantly much better performance than PSO based wrapper. In comparison, the total number of features in the heart dataset is 13 and the effect of hybridization is manifested only in the short run, e.g., the classification accuracies in 100 iterations are similar. However, for Heart dataset with $d = 6$, the hybrid method attains near optimal classification accuracy in far less iterations (Fig. 8(b)). Similar results are observed for other datasets but are excluded for brevity.

The average CPU time (in s) for 100 iterations of PSO search for sonar dataset is 116.53 when seeking five features and 146.59 for 10 features. In the case of *mr²PSO*, the times are 139.57 when seeking five features and 286.49 when seeking 10 features. In the case of heart dataset, the PSO search times are 216.91 for three features and 642.94 for six features whereas *mr²PSO* takes 249.89 for three features and 704.17 for six features. CPU results for other datasets were similar. Therefore, we can conclude that, given the significant performance improvement potential under *mr²PSO* and somewhat comparable CPU times, the proposed method is rather effective for the task of feature selection.

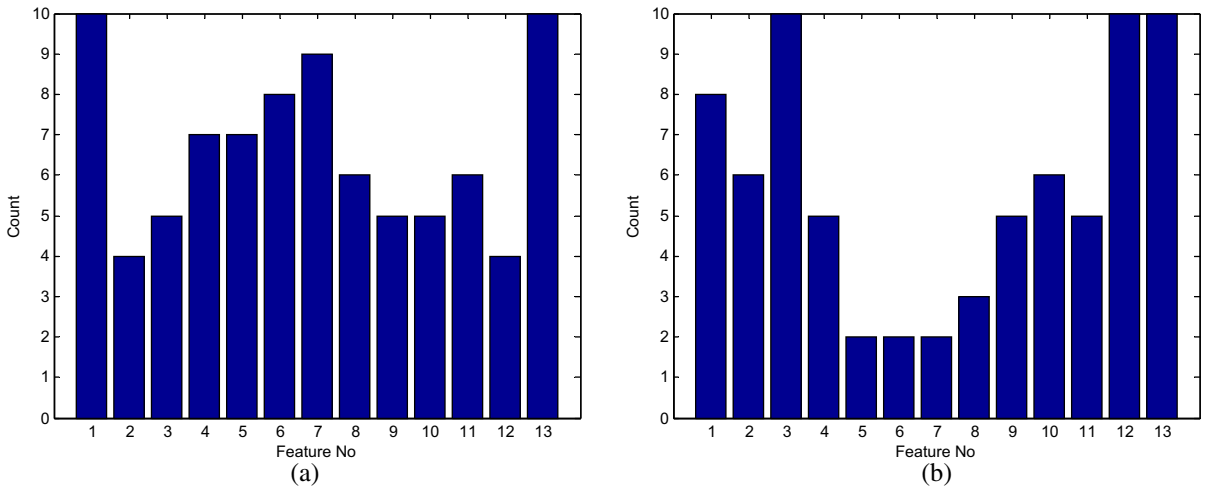
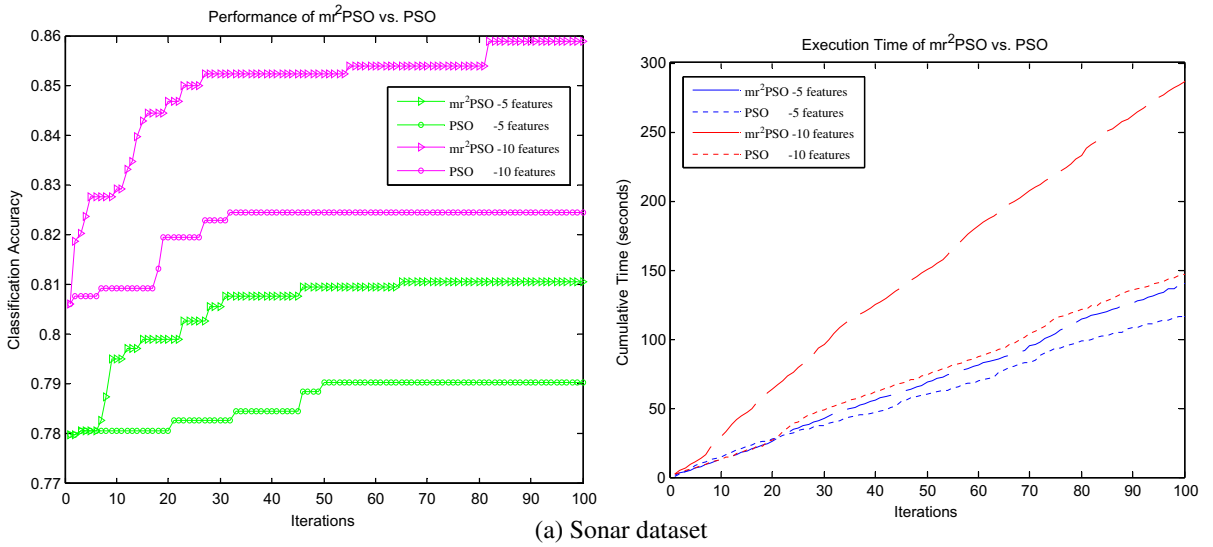
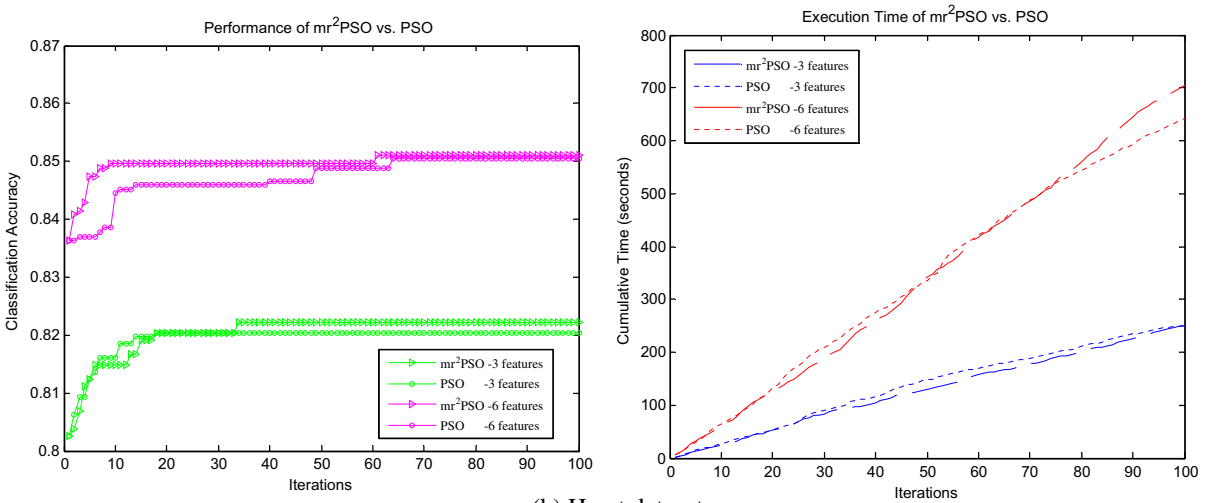


Fig. 7. Commonality of features selected during 10 different runs of the search algorithm: (a) wine dataset and (b) heart dataset.



(a) Sonar dataset



(b) Heart dataset

Fig. 8. Performance of hybrid filter–wrapper *mr²PSO* and *PSO*-based wrapper over 10 experiments.

5. Conclusion

In this study, we proposed a framework that combines the advantages of filter and wrapper type of feature subset selection algorithms and embedded this framework into the particle swarm optimization heuristic. Different than the earlier hybrid filter–wrapper algorithms that use filter and wrapper models in sequence, our approach structurally integrates the filter model within the PSO based wrapper model. The filter model is based on the mutual information and is expressed as a composite measure of feature relevance and redundancy. This relevance and redundancy composite criterion is used to weigh the probabilities of features to be included in the feature subset. Hence, it enhances the convergence rate as well as the solution quality of the feature subset selection problem. We compared the performance of the proposed hybrid algorithm, in terms of both quality as well as speed, with a recently proposed hybrid filter–wrapper method based on a genetic algorithm as well as a PSO based wrapper method. The results indicated that the proposed method is superior with respect to both alternatives. While we adopted identical SVM and RBF kernel parameters (C and σ) as in [20] for comparison purposes, the best parameter settings depend on the given problem. For best results, we recommend a parameter search to identify good settings so that the classifier's prediction accuracy is improved [24]. Further, we used 2-fold cross-validation method as in [20], which is known to be pessimistically biased and increase the variance of the prediction accuracy [25]. Hence, we recommend stratified 10-fold cross-validation with larger number of samples than the ones used in this study.

Future extensions of this proposed methodology aim to improve the efficiency of the hybrid feature selection algorithm. In the current implementation, we sequentially construct the feature subset by including one feature at a time. As a result, when a feature's mutual information is calculated and used to weigh the feature's probability (e.g., velocity), this weight is

based on the feature subset available thus far. Hence, there is a dependence on the feature subset construction sequence. One way to remove this dependency is to perform backtracking in the end and randomly re-evaluate the features in the subset. This corresponds to excluding features from the subset and replacing them with better alternatives. Another interesting extension of this study would be to compare the proposed method with feature extraction methods.

Acknowledgements

We thank two reviewers and the editor whose constructive suggestions have significantly improved the content and the presentation of this paper. The authors are grateful for financial support from the Turkish National Science Foundation (TUBITAK-BIDEB 2219).

References

[1] B. Apollonia, S. Bassisa, A. Brega, Feature selection via Boolean independent component analysis, Information Sciences 179 (22) (2009) 3815–3831.
[2] M. Ben-Bassat, Irrelevant features in pattern recognition, IEEE Transactions on Computers 27 (8) (1978) 746–749.
[3] B.V. Bonnlander, A.S. Weigend, Selecting input variables using mutual information and nonparametric density evaluation, ISSAN 94 (1994) 42–50.
[4] D. Chen, C.Z. Wang, Q.H. Hu, A new approach to attribute reduction of consistent and inconsistent covering decision systems with covering rough sets, Information Sciences 17 (1) (2007) 3500–3518.
[5] T.W.S. Chow, D. Huang, Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information, IEEE Transactions on Neural Networks 16 (1) (2005) 213–224.
[6] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation 6 (1) (2002) 58–73.
[7] C. Cornelis, R. Jensen, G. Hurtado, D. Ślezak, Attribute selection with fuzzy decision reducts, Information Sciences 180 (2) (2010) 209–224.
[8] S. Das, Filters, wrappers and a boosting-based hybrid for feature selection, in: Proceedings of the 18th International Conference on Machine Learning, 2001, pp. 74–81.
[9] M. Dash, H. Liu, Feature selection for classification, Intelligent Data Analysis 1 (1997) 131–156.
[10] J.C.W. Debuse, V.J. Rayward-Smith, Feature subset selection within a simulated annealing data mining algorithm, Journal of Intelligent Information Systems 9 (1) (1997) 57–81.
[11] E. Elbeltagi, T. Hegazy, D. Grierson, Comparison among five evolutionary-based optimization algorithms, Advanced Engineering Informatics 19 (1) (2005) 43–53.
[12] H.J. Escalante, M. Montes, L.E. Sucar, Particle swarm model selection, Journal of Machine Learning Research 10 (2009) 405–440.
[13] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery: an overview, Advances in Knowledge Discovery and Data Mining (1996) 1–34.
[14] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.
[15] R. Hassan, B. Cohanim, O. Weck, A comparison of particle swarm optimization and the genetic algorithm, in: The Proceedings of 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 2005, pp. 18–21.
[16] E.R. Hruschka, E.R. Hruschka, T.F. Covões, N.F.F. Ebecken, Feature selection for clustering problems: a hybrid algorithm that iterates between *k*-means and a Bayesian filter, in: The Proceedings of the International Conference on Hybrid Intelligent Systems, vol. 5, 2005, pp. 405–410.
[17] W.H. Hsu, Genetic wrappers for feature selection in decision tree induction and variable ordering in Bayesian network structure learning, Information Sciences 163 (2004) 103–122.
[18] H.-H. Hsu, C.-W. Hsiesh, M.-D. Lu, A hybrid features selection mechanism, in: The Proceedings of Eighth International Conference on Intelligent Systems Design and Applications, 2008, pp. 271–277.
[19] Q. Hu, D. Yu, J. Liu, C. Wu, Neighborhood rough set based heterogeneous feature subset selection, Information Sciences 178 (18) (2008) 3577–3594.
[20] J. Huang, Y. Cai, X. Xu, A hybrid genetic algorithm for feature selection wrapper based on mutual information, Pattern Recognition Letters 28 (13) (2007) 1825–1844.
[21] A.K. Jain, B. Chandrasekaran, Dimensionality and sample size considerations in pattern recognition in practice, in: P.R. Krishnaiah, L.N. Kanal (Eds.), Handbook of Statistics, vol. 2, North-Holland, Amsterdam, 1982, pp. 835–855.
[22] J. Kennedy, R. Eberhart, Particle swarm optimization, in: The Proceedings of the 1995 IEEE International Conference on Neural Network, 1995, pp. 1942–1948.
[23] J. Kennedy, R. Eberhart, A discrete binary version of the particle swarm algorithm, The Proceedings of the International Conference on Systems, Man and Cybernetics, vol. 5, IEEE Press, 1997, pp. 4104–4108.
[24] S. Keerthi, C.J. Lin, Asymptotic behaviors of support vector machines with Gaussian kernel, Neural Computation 15 (7) (2003) 1667–1689.
[25] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: The Proceedings of 14th International Joint Conference on Artificial Intelligence, 1995, pp. 1137–1143.
[26] R. Kohavi, G. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1-2) (1997) 273–324 (Special Issue on Relevance).
[27] M. Kudo, J. Sklansky, Comparison of algorithms that select features for pattern classifiers, Pattern Recognition 33 (1) (2000) 25–41.
[28] H. Liu, J. Sun, L. Liu, H. Zhang, Feature selection with dynamic mutual information, Pattern Recognition 42 (7) (2009) 1330–1339.
[29] H. Liu, L. Yu, Towards integrating feature selection algorithms for classification and clustering, IEEE Transactions on Knowledge and Data Engineering 17 (4) (2005) 491–502.
[30] Y. Liu, Y.F. Zheng, FS-SFS: a novel feature selection method for support vector machines, Pattern Recognition 39 (2006) 1333–1345.
[31] S. Maldonado, R. Weber, A wrapper method for feature selection using support vector machines, Information Sciences 179 (13) (2009) 2208–2217.
[32] T. Marill, D.M. Green, On the effectiveness of receptors in recognition systems, IEEE Transactions on Information Theory 9 (1963) 11–17.
[33] R. Meiri, J. Zahavi, Using simulated annealing to optimize the feature selection problem in marketing applications, European Journal of Operational Research 171 (3) (2006) 842–858.
[34] P. Mitra, C.A. Murthy, S.K. Pal, Unsupervised feature selection using feature similarity, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (3) (2002) 301–312.
[35] S. Nakariyakul, D. Casasent, An improvement on floating search algorithms for feature subset selection, Pattern Recognition 42 (9) (2009) 1932–1940.
[36] P.M. Narendra, K. Fukunaga, A branch and bound algorithm for feature subset selection, IEEE Transactions on Computers 26 (9) (1977) 917–922.
[37] D.J. Newman, S. Hettich, C.L. Blake, C.J. Merz, UCI repository of machine learning databases. Irvine, CA: Dept. Inf. Computer. Sci., Univ. California, 1998. [Online] Available from: <http://archive.ics.uci.edu/ml/datasets.html>.
[38] W. Pedrycz, B.J. Park, N.J. Pizzi, Identifying core sets of discriminatory features using particle swarm optimization, Expert Systems with Applications 36 (2009) 4610–4616.
[39] Y. Peng, W. Li, Y. Liu, A hybrid approach for biomarker discovery from microarray gene expression data for cancer classification, Cancer Informatics 2 (2006) 301–311.
[40] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8) (2005) 1226–1238.

[41] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization. An overview, Swarm Intelligence 1 (1) (2007) 33–57.
[42] M.L. Raymer, W.F. Punch, E.D. Goodman, L.A. Kuhn, A.K. Jain, Dimensionality reduction using genetic algorithms, IEEE Transactions On Evolutionary Computation 4 (2) (2000) 164–171.
[43] T. Reinartz, A unifying view on instance selection, Data Mining and Knowledge Discovery 6 (2) (2002) 191–210.
[44] J. Reunanen, Overfitting in making comparisons between variable selection methods, Journal of Machine Learning Research 3 (2003) 1371–1382.
[45] M. Sebban, R. Nock, A hybrid filter/wrapper approach of feature selection using information theory, Pattern Recognition 35 (4) (2002) 835–846.
[46] Y. Shi, R. Eberhart, Parameter selection in particle swarm optimization, in: The Proceedings of 7th Annual Conference on Evolutionary Programming, 1998, pp. 591–600.
[47] R. Sikora, S. Piramuthu, Framework for efficient feature selection in genetic algorithm based data mining, European Journal of Operational Research 180 (2007) 723–737.
[48] P. Somol, J. Novovičová, P. Pudil, Flexible hybrid sequential floating search in statistical feature selection, Lecture Notes in Computer Science 4109 (2006) 632–639.
[49] P. Somol, P. Pudil, J. Kittler, Fast branch and bound algorithms for optimal feature selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (7) (2004) 900–912.
[50] I.C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Information Processing Letters 85 (6) (2003) 317–325.
[51] O. Uncu, I.B. Turksen, A novel feature selection approach: combining feature wrappers and filters, Information Sciences 177 (2) (2007) 449–466.
[52] E.P. Xing, M.I. Jordan, R.M. Karp, Feature selection for high-dimensional genomic microarray data, in: The Proceedings of 18th International Conference on Machine Learning, 2001, pp. 601–608.
[53] X. Wang, J. Yang, X. Teng, W. Xia, R. Jensen, Feature selection based on rough sets and particle swarm, Optimization Pattern Recognition Letters 28 (4) (2007) 459–471.
[54] Y. Wang, L. Li, J. Ni, S. Huang, L. Rokach, Genetic algorithm-based feature set partitioning for classification problems, Pattern Recognition 41 (5) (2008) 1676–1700.
[55] Y. Wang, L. Li, J. Ni, S. Huang, Feature selection using tabu search with long-term memories and probabilistic neural networks, Pattern Recognition Letters 30 (2009) 661–670.
[56] A.W. Whitney, A direct method of nonparametric measurement selection, IEEE Transactions on Computers C-20 (9) (1971) 1100–1103.
[57] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, IEEE Intelligent Systems and their Applications 13 (2) (1998) 44–49.
[58] Y. Yao, Y. Zhao, Attribute reduction in decision-theoretic rough set models, Information Sciences 178 (17) (2008) 3356–3373.
[59] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, Journal of Machine Learning Research 5 (2004) 1205–1224.
[60] B. Yu, B. Yuan, A more efficient branch and bound algorithm for feature subset selection, Pattern Recognition 26 (6) (1993) 883–889.
[61] M. Zhang, J. Peñac, V. Robles, Feature selection for multi-label naive Bayes classification, Information Sciences 179 (19) (2009) 3218–3229.
[62] H. Zhang, G. Sun, Feature selection using tabu search method, Pattern Recognition 35 (2002) 701–711.
[63] S. Zhao, E.C.C. Tsang, On fuzzy approximation operators in attribute reduction with fuzzy rough sets, Information Sciences 178 (16) (2007) 3163–3176.
[64] D. Lewis, Feature selection and feature extraction for text categorization, in: The Proceedings of a Workshop on Speech and Natural Language, Morgan Kaufmann, San Mateo, CA, 1992, pp. 212–217.
[65] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Transactions on Neural Networks 5 (4) (1995) 537–550.
[66] J. Cohen, A coefficient of agreement for nominal scales, Educational and Psychological Measurement 20 (1) (1960) 37–46.