JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 27, 1667-1686 (2011)

Feature Subset Selection Based on Bio-Inspired Algorithms*

CHULMIN YUN, BYONGHWA OH, JIHOON YANG+ AND JONGHO NANG
Department of Computer Science and Engineering
Sogang University
Seoul, 121-742 Korea

Many feature subset selection algorithms have been proposed and discussed for years. However, the problem of finding the optimal feature subset from full data still remains to be a difficult problem. In this paper, we propose novel methods to find the relevant feature subset by using biologically-inspired algorithms such as Genetic Algorithm and Particle Swarm Optimization. We also propose a variant of the approach considering the significance of each feature. We verified the performance of the proposed methods by experiments with various real-world datasets. Our feature selection methods based on the biologically-inspired algorithms produced better performance than other methods in terms of the classification accuracy and the feature relevance. In particular, the modified method considering feature significance demonstrated even more improved performance.

Keywords: genetic algorithm, particle swarm optimization, feature redundancy and relevance, wrapper approach, inductive learning algorithm

1. INTRODUCTION

Feature subset selection is about finding the optimal subset of features, among the full features, that renders the best performance in terms of well-defined criteria such as the classification accuracy in labeled data and the total cost associated with feature subsets [1-3]. The importance of feature selection in machine learning stems from its ability to improve such learning performance. In other words, through feature selection, we can reduce the cost of learning (both the acquisition cost/risk of feature values and the computational overhead in learning) and obtain higher classification accuracy, compared to the learning with the entire feature set.

In order to handle the inherent exponential complexity of the task (*i.e.* the existence of exponential number of candidate subsets), a number of approaches to feature subset selection have been proposed in the literature [1-21], which are based on diverse search strategies. (See [1-3, 9-12] for surveys.) First, an exhaustive search was employed to find the best feature subset under certain criteria [4, 5]. In this approach, the candidate feature subsets are evaluated with respect to the performance measure and an optimal feature subset is found using exhaustive search. However, exhaustive search is computationally infeasible in practice, except in those rare instances where the total number of features is quite small. Therefore, a number of researchers have explored the use of heuristics or randomized algorithms for feature subset selection. For instance, features were either selected (starting from an empty feature subset) or eliminated (starting from the entire feature set) sequentially to determine the final feature subset [6-8]. Heuristics such as mu-

Received December 7, 2009 revised March 29 & November 1, 2010; accepted December 15, 2010.
Communicated by Chih-Jen Lin.
* This paper was supported by the Special Research Grant of Sogang University to Jihoon Yang.
+ Corresponding author.

tual information, relevance, and relevance of each feature were also employed to find the optimal feature subset [13-17]. In addition, several authors explored the use of randomized population-based heuristic search techniques for feature subset selection [3, 18, 19].

Though each of the various approaches described above has its own rationale and criterion for finding quality feature subsets, it has demonstrated limited success and the feature subset selection still remains as a difficult task. Against this background, we aim to develop novel feature subset selection methods that produce quality solutions in terms of search criteria (*e.g.* classification accuracy, feature costs). More precisely, the new method combines biologically-inspired algorithms and well-defined heuristics expecting to inherit the merits of each approach, based on our previous experimental work on the task [20, 21].

Biologically-inspired (or *bio-inspired* in short) algorithms have been contrived based on the principles of the behavior of organisms, and applied to mainly optimization problems [22-26]. Among the various approaches, we consider two bio-inspired algorithms in this paper: *Genetic Algorithm* (*GA*) and *Particle Swarm Optimization* (*PSO*). *GA* is based on the survival of the fittest and tries to find the best solution through evolutionary processes of crossover and mutation among the individuals in the search space. PSO mimics the phenomenon of the swarm (or flocking) of creatures and attempts to find the best solution by changing the form of the swarm of individuals. As the bio-inspired algorithms are known to be generally quite effective for rapid global search of large search space in difficult optimization problems, we aim to combine the bio-inspired algorithms (as a *wrapper* defined in [3]) with inductive learning algorithms in a bid to find the optimal feature subset that yields the best performance. (See [27-29] for explanations on inductive learning algorithms.) Feature subset selection algorithms are said to follow a *filter* approach if feature selection is performed independently of the learning algorithm used, and said to follow a wrapper approach otherwise. In addition, we compare the performance of bio-inspired approaches to those of several state-of-the-art approaches, and finally propose new algorithms for feature subset selection. To this end, we combine the bio-inspired methods with the well-defined *significance* (or *relevance*) of each feature (by *mRMR* [13] as described in section 3) which was verified to be the best heuristic in [20, 21].

The rest of the paper is organized as follows: Section 2 briefly describes some of the characteristics that need to be considered in feature subset selection. Section 3 includes detailed descriptions on our approach that uses bio-inspired algorithms combined with inductive learning algorithms and the significance of a feature in subset selection. Section 4 explains the various real-world data and experimental setup designed to evaluate the performance of our approaches, followed by the results of experiments presented in section 5. Section 6 concludes with summary and discussion of some directions for future research.

2. CHARACTERISTICS OF FEATURE SUBSET SELECTION

Ideally, feature selection methods should choose the optimal feature subset from the candidates that best describes the target concept inherent in the data. The following aspects need to be considered in the process of feature selection.

2.1 Starting Point

First, we must determine the starting point in the feature space and the direction of search. The search can start with no features (*i.e.* empty subset), and keep adding features in the subset. Or, it can start with all features (*i.e.* full subset), and keep eliminating features. The former is called *forward selection*, and the latter is known as *backward elimination* [3, 10]. Generalizing the idea of forward selection and backward elimination, we can start the search with any number and combination of features.

2.2 Search Strategy

Theoretically, the best feature subset can be found by evaluating all the possible subsets. However, such an exhaustive search of the feature space needs to explore all of 2^n possible subsets of n features, which is impractical for large number of features. Therefore, we have to resort to more realistic approaches. As described in section 1, a variety of approaches have been proposed for this purpose [1-21]. Though these approaches are practical, they are not guaranteed to find the optimal subset of features. This is natural because those approaches sacrifice the quality of the feature subset (to an acceptable degree) for the computational overhead (avoiding the exponential complexity). It is thus of importance to find a good algorithm considering this trade-off.

2.3 Subset Evaluation

After generating candidate feature subsets, we need to evaluate them. As aforementioned, a feature subset selection algorithm is termed either wrapper or filter approach by whether it makes use of a learning algorithm for evaluating feature subsets or not. In other words, the wrapper approach determines the goodness of a feature subset by applying it to a learning algorithm and evaluate the performance (*e.g.* by the classification accuracy). On the other hand, the filter approach evaluates features using some measures independent of the learning algorithm (*e.g.* mutual information [28], *mRMR* [13]).

2.4 Stopping Criteria

Finally, we must decide the criteria for halting the search. For example, we can stop adding or removing features when none of the alternatives improves the performance, or when the number of selected features reaches a pre-determined threshold [11]. We can then choose the best subset among the candidates we have encountered during the search.

3. BIO-INSPIRED APPROACHES TO FEATURE SUBSET SELECTION

Feature subset selection is a hard task and not efficiently manageable when the dimensionality of the feature space is high. Bio-inspired algorithms are appropriate candidates to the task, producing quality solutions within reasonable amount of time and efforts. We propose wrapper-based approaches to feature subset selection based on two kinds of bio-inspired algorithms as described in this section.

3.1 Bio-Inspired Algorithms

Bio-inspired algorithms have been conceived based on the principles of the behavior or the phenomena in living organisms and creatures, such as gene evolution, insect swarming, bird swarming, food foraging, and the like [25]. Bio-inspired algorithms are well-known for their applicability to optimization problems. Each individual in a bio-inspired algorithm represents a candidate solution to the problem, and the algorithm converges to the optimal solution (under certain assumptions) through the evolutionary interactions of the individuals in the solution space.

There exist a variety of bio-inspired algorithms [22-25], among which the *Genetic Algorithm* (*GA*), and the *Particle Swarm Optimization* (*PSO*) are considered in the paper. (Yet another popular bio-inspired algorithm, the Ant Colony Optimization (ACO) was experimentally proven to perform comparable to *GA* in our previous work [19] and is thus not considered in this paper.)

A fitness function is an objective function that quantifies the optimality of a solution (*i.e.* individual) in bio-inspired algorithms including *GA* and *PSO*. In this paper, the fitness function is defined by the accuracy of a learning algorithm. That is, each individual is evaluated by the learning accuracy based on the feature subset it represents. The bio-inspired algorithm attempts to find the best feature subset through evolution by wrapping the evaluation process of candidate solutions.

3.2 Feature Subset Selection using Genetic Algorithm (*GAFSS*)

The *Genetic Algorithm* (*GA*) is one of the bio-inspired algorithms using techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover [23]. Typically, solutions (called individuals or *chromosomes*) are represented as strings in *GA*. The evolution starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of each individual is evaluated, and multiple individuals are stochastically selected from the current population based on their fitness, and modified to form a new population by genetic operations such as crossover and mutation. The new population is then used in the next iteration of the algorithm. The algorithm terminates when either the maximum number of generations has been reached, or a satisfactory fitness level has been obtained for the population.

In our *GA*-based feature subset selection, each individual is represented as a binary string encoding a feature subset. If the data consist of *N* features, an individual will be an *N*-bit binary string. If a bit is 1 the feature is chosen in the feature subset; if 0 it is not. Each individual in the population is thus a candidate feature subset. The initial population, whose size is set to 40 in our experiments, consists of randomly generated such individuals. The selection, crossover, and mutation processes are implemented in a standard way of *GA* [23]. That is, a fitness-proportionate selection is used for choosing mating pairs of individuals; a single-point crossover was adopted for crossover operation with the probability of 0.6; and mutation operation was applied to flip each bit of an individual (*i.e.* either from 1 to 0 or from 0 to 1) with the probability of 0.03. The maximum number of generation was set to 20 in our experiments. The parameter settings were based on results of preliminary runs.

3.3 Feature Subset Selection Using Particle Swarm Optimization (*PSOFSS*)

Particle Swarm Optimization (*PSO*) is a stochastic, population-based evolutionary algorithm introduced by Kennedy and Eberhart [22, 24]. Similar to *GA*, a population of individuals (or *particles*) is initialized as candidate solutions for a given problem in *PSO*. The particles iteratively evaluate the fitness of the candidate solutions and remember the *location* where they had their best fitness. The particle's best solution is called the *particle best* (*pbest*) or the *local best* (*lbest*), and the best single solution among all of the particles is called the *global best* (*gbest*). In the solution space, each particle makes suitable changes in its *position* and *velocity* iteratively with respect to the best solutions.

Let $X_i^{(t)} = (x_{i1}, x_{i2}, x_{iD})$ be the position (vector) of particle i and $V_i^{(t)} = (v_{i1}, v_{i2}, \ldots, v_{iD})$ be the velocity (vector) of particle i at time t in D-dimensional space, respectively. Then particle i changes its position and velocity iteratively as follows,

$$V_i^{(t)} = V_i^{(t\text{-}1)} + c_1(P_i - X_i^{(t\text{-}1)}) + c_2(P_g - X_i^{(t\text{-}1)}) \tag{1}$$

$$X_i^{(t)} = X_i^{(t\text{-}1)} + V_i^{(t)} \tag{2}$$

where $P_i = \{p_{i1}, p_{i2}, \ldots, p_{iD}\}$ is the *pbest* of particle i, $P_g = \{p_{g1}, p_{g2}, \ldots, p_{gD}\}$ is the *gbest* of all particles, and c_1 and c_2 are *cognitive factors*, respectively. As the algorithm continues, the fitness of the global best solution keeps improving.

To handle the feature subset selection task with *PSO*, we followed the same approach that we used in *GA*. Like the chromosomes in *GA*, each particle in the swarm represents a candidate feature subset. We also define the position vector of a particle as a binary string and the fitness function as the accuracy of a learning algorithm.

Because of the binary values in the position vector of particles, we need to modify the operations (1) and (2) of *PSO* accordingly. We followed the modified *PSO* operations suggested by Kennedy and Eberhart [24]. Let $x_{id}^{(t)}$ be the dth element of $X_i^{(t)}$. Then $x_{id}^{(t)}$ changes its value to 0 or 1 iteratively as follows,

$$v_{id}^{(t)} = v_{id}^{(t-1)} + c_1(p_{id} - x_{id}^{(t-1)}) + c_2(p_{gd} - x_{id}^{(t-1)}), \tag{3}$$

$$s(v_{id}^{(t)}) = \frac{1}{1 + \exp(-v_{id}^{(t)})}, \tag{4}$$

$$\text{if } \rho_{id} < s(v_{id}^{(t)}), \text{ then } x_{id}^{(t)} = 1; \text{ else } x_{id}^{(t)} = 0, \tag{5}$$

where $v_{id}^{(t)}$, p_{id} and p_{gd} are the dth elements of $V_i^{(t)}$, P_i and P_g as defined previously, $s(v_{id})$ is a *sigmoid function* to normalize the value of v_{id} into [0.0, 1.0], and ρ_{id} is a vector of random numbers drawn from a uniform distribution between 0.0 and 1.0. In these binary *PSO* operations, the velocity parameter $v_{id}^{(t)}$ acts as an individual's disposition to make one or the other choice (*i.e.* to select the feature or not). If $v_{id}^{(t)}$ is high ($s(v_{id})$ is close to 1.0), the individual $x_{id}^{(t)}$ is more likely to choose 1, and 0 otherwise. c_1 and c_2 are set to 2 in our experiments after some preliminary runs. In addition, the same parameter values for the population size and the number of generations of 40 and 20 were used for a fair comparison with *GAFSS*.

3.4 Combining Feature Relevance with *GAFSS* and *PSOFSS*

We have explained *GAFSS* and *PSOFSS* feature subset selection methods as wrapper approaches with *GA* and *PSO*, respectively. *GAFSS* and *PSOFSS* simply rely on the classification accuracy (of a learning algorithm used in the wrapper) of candidate solutions during the process of evolution. While the accuracy is the appropriate criterion to be considered, the *relevance* (or *significance* or *goodness*) of each feature can provide additional information and thus help produce the best feature subset. For instance, one of such relevance measures, the *mutual information* describes how two features are related to each other [28]. The relevance measures are actually the ones that have been applied in filter approaches for feature subset selection [3]. Here, we extend *GAFSS* and *PSOFSS* by incorporating the relevance of features and modifying the evolution operators accordingly.

First, we need to determine the method for measuring the relevance of a feature. A variety of filter approaches have been proposed in the literature among which several methods were compared experimentally [3, 20, 21]. Based on the experimental results, we chose the *mRMR* method [13] that was based on mutual information and showed the best performance. The mutual information of two random variables is a quantity that measures their mutual dependence [27]. Generally, the mutual information of two discrete random variables X and Y is defined as:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log\left(\frac{p(x, y)}{p(x)\, p(y)} \right), \tag{6}$$

where $p(x, y)$ is the joint probability distribution function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y, respectively. Intuitively, the mutual information is to measure the information that X and Y share. In other words, it measures how much the knowledge on one variable reduces the uncertainty about the other.

In *mRMR*, the mutual information between a feature and a class is used as the *relevance* of the feature for the class. If there exist data D that consist of N features $x_1, x_2, \ldots, x_N$ and a class c, then the relevance of feature x_n ($1 \le n \le N$) is defined by the mutual information between feature x_n and the class c:

$$relevance(x_n) = I(x_n; c). \tag{7}$$

This reflects the dependency of feature x_n on the target class c. The feature x_i with the largest relevance has the largest dependency on the class.

mRMR also considers the mutual information between features as the *redundancy* of each feature. The redundancy of feature x_n in a feature subset FS is defined by following equation:

$$redundancy(x_n) = \frac{1}{|FS| - 1} \sum_{x_i \neq x_n, x_i \in FS} I(x_i; x_n). \tag{8}$$

This indicates the dependency of feature x_n with other features. For example, when two

features are highly dependent on each other, the class-discriminative power would not change much if one of them were removed. In this case, these two features are said to have high redundancy. We thus need to find features that have low redundancy in order to find mutually exclusive features.

The criterion combining above two conditions is called *Minimal-Redundancy and Maximal-Relevance* (*mRMR*) [13]. It means that the goodness of a feature becomes larger if the feature has lower redundancy and higher relevance. A feature with low redundancy and high relevance implies that the feature is mutually exclusive to other features and highly dependent on the target class. The *mRMR* measure of feature x_n is defined as:

$$mRMR(x_n) = relevance(x_n) - redundancy(x_n). \tag{9}$$

To set the relevance of each feature, we first measure the *mRMR* values of all features. We represent the *mRMR* values of features in $D = \{x_1, x_2, \ldots, x_N\}$ as $\{m_1, m_2, \ldots, m_N\}$. Then we calculate the average ($\bar{m}$) and the standard deviation (s) of the values as:

$$\bar{m} = \frac{1}{N}\sum_{i=1}^{N} m_i, \tag{10}$$

$$s = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (m_i - \bar{m})^2}. \tag{11}$$

Also, we divide the data D into three sets which satisfy $D = D_1 \cup D_2 \cup D_3$:

$$D_1 = \left\{ x_i \mid m_i - \bar{m} > \frac{s}{2} \right\}, D_2 = \left\{ x_i \mid -\frac{s}{2} \le m_i - \bar{m} \le \frac{s}{2} \right\}, D_3 = \left\{ x_i \mid m_i - \bar{m} < -\frac{s}{2} \right\}. \tag{12}$$

This is to separate the features into three different categories: features with higher *mRMR* values are in D_1, lower *mRMR* values in D_3, and others in D_2. Choosing the half of the standard deviation as the threshold for this categorization of data is a simple heuristic determined based on results of several preliminary runs. Our approach to combine *GAFSS* and *PSOFSS* with *mRMR* relevance measure is described in the remaining part of this section.

3.5 *GAFSS + mRMR*

GAFSS combined with *mRMR* (*GAFSS + mRMR*) is the same as simple *GAFSS* except the *mutation*. The mutation processes of *GAFSS* and *GAFSS + mRMR* are shown in Algorithms 1 and 2, respectively. (The overall description on *GAFSS* was given in section 3.2.)

In *GAFSS + mRMR*, the relevance of a feature is checked before performing the mutation operation. When the value of a bit is about to change from 1 to 0, the mutation is actually happening only if the corresponding feature is not in D_1. This means that the feature is not removed from the feature subset if it is presumed to be relevant (in terms of *mRMR*). Similarly, the mutation of a bit from 0 to 1 occurs only if the feature is not in D_3. This means that the feature is not added to the feature subset if it is presumed to be irre-

Algorithm 1: *GAFSS*
N: Size of population
n: Number of bits in an individual
$G_1 = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$: List of features ($x_n$ = 0 or 1) for individual *i*
P_m: Probability of mutation ($0 < P_m < 1$)
…
/* Do Mutation */
For *i* = 1 to *N*
    For *j* = 1 to *n*
        get random value *rand* ($0 < rand < 1$)
        If ($rand < P_m$)
            If ($x_{ij} == 0$) then $x_{ij} = 1$
            Else If ($x_{ij} == 1$) then $x_{ij} = 0$
        End if
    End for
End for
…

Algorithm 2: *GAFSS + mRMR*
N: Size of population
n: Number of bits in an individual
$G_1 = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$: List of features ($x_n$ = 0 or 1) for individual *i*
P_m: Probability of mutation ($0 < P_m < 1$)
…
/* Do Mutation */
For *i* = 1 to *N*
    For *j* = 1 to *n*
        get random value *rand* ($0 < rand < 1$)
        If ($rand < P_m$)
            If ($x_{ij} == 0$)
                If ($x_j \in D_3$) then $x_{ij} = 0$
                Else $x_{ij} = 1$
            Else If ($x_{ij} == 1$)
                If ($x_j \in D_1$) then $x_{ij} = 1$
                Else $x_{ij} = 0$
        End if
    End for
End for
…

levant. By combining *GAFSS* and *mRMR* in this fashion, we control the mutation process and attempt to include features of high significance and to exclude features of low significance, which could improve the performance of *GAFSS*.

3.6 *PSOFSS + mRMR*

Similar to *GAFSS + mRMR*, *PSOFSS* combined with *mRMR* (*PSOFSS + mRMR*) is the same as simple *PSOFSS* except the random process (Eq. (5)) which corresponds to the mutation step of *GAFSS + mRMR* in Algorithm 2. The basic operations of *PSOFSS*

and *PSOFSS + mRMR* (except the *pbest* and *gbest* determination steps) are shown in Algorithms 3 and 4, respectively. (The overall description on *PSOFSS* was given in section 3.3.)

Algorithm 3: *PSOFSS*

N: Number of particles
D: Number of bits in particle (number of full features)
T: Number of iteration
$X_i^{(t)} = (x_{i1}^{(t)}, x_{i2}^{(t)}, \ldots, x_{iD}^{(t)})$: List of features ($x_n$ = 0 or 1) for particle *i*
$V_i^{(t)} = (v_{i1}^{(t)}, v_{i2}^{(t)}, \ldots, v_{iD}^{(t)})$: Velocity of *i*th particle at iteration *t*
$P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$: *pbest* of particle *i*
$P_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$: *gbest*
c_1, c_2: cognitive parameters
…
For *t* = 1 to *T*
  For *i* = 1 to *N*
    For *d* = 1 to *D*

$$v_{id}^{(t)} = v_{id}^{(t-1)} + c_1(p_{id} - x_{id}^{(t-1)}) + c_2(p_{gd} - x_{id}^{(t-1)})$$

$$s(v_{id}^{(t)}) = \frac{1}{1+\exp(-v_{id}^{(t)})}$$

      Get random a value ρ_{id} ($0.0 \le \rho_{id} \le 1.0$)
      If ($\rho_{id} < s(v_{id}^{(t)})$) then $x_{id}^{(t)} = 1$
      Else $x_{id}^{(t)} = 0$
    End for
  End for
End for
…

Algorithm 4: *PSOFSS + mRMR*

N: Number of particles
D: Number of bits in particle (number of full features)
T: Number of iteration
$X_i^{(t)} = (x_{i1}^{(t)}, x_{i2}^{(t)}, \ldots, x_{iD}^{(t)})$: List of features ($x_n$ = 0 or 1) for particle *i*
$V_i^{(t)} = (v_{i1}^{(t)}, v_{i2}^{(t)}, \ldots, v_{iD}^{(t)})$: Velocity of *i*th particle at iteration *t*
$P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$: *pbest* of particle *i*
$P_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$: *gbest*
c_1, c_2: cognitive parameters
…
For *t* = 1 to *T*
  For *i* = 1 to *N*
    For *d* = 1 to *D*

$$v_{id}^{(t)} = v_{id}^{(t-1)} + c_1(p_{id} - x_{id}^{(t-1)}) + c_2(p_{gd} - x_{id}^{(t-1)})$$

$$s(v_{id}^{(t)}) = \frac{1}{1+\exp(-v_{id}^{(t)})}$$

```
                Get a random value ρ_id (0.0 ≤ ρ_id ≤ 1.0)
                If (ρ_id < s(v_id^(t)))
                    If (x_d ∈ D_3) then x_id^(t) = 0
                    Else x_id^(t) = 1
                Else
                    If (x_d ∈ D_1) then x_id^(t) = 1
                    Else x_id^(t) = 0
                End if
            End for
        End for
    End for
…
```

As shown in Algorithm 4, the relevance of a feature is checked if it is in D_1 or D_3 in *PSOFSS + mRMR*. When the value of a bit is to be set 1 (*i.e.* for high velocity), the relevance of the feature is checked to see if it is in D_3, in which case the value is set to 0. Similarly, when the value of a bit is to be set 0 (*i.e.* for low velocity), the relevance of the feature is checked to see if it is in D_1, in which case the value is set to 1. By following this method, we attempt to include features of high significance and to exclude features of low significance, which could improve the performance of *PSOFSS*.

4. EXPERIMENTS

4.1 Dataset

20 real-world datasets were used in our experiments. All the datasets are from the *UCI Machine Learning Repository* [30] and summarized in Table 1. Though some of the datasets consist of about 20 features, most of them have large number of features, so they are appropriate to the feature subset selection task. The datasets are also diverse in terms of the number of classes and samples, as well. (For detailed descriptions on each dataset, see [30].)

4.2 Experimental Setup

Our experiments were conducted with three purposes. First, we compared the performance of *GAFSS* and *PSOFSS* with some of the existing feature subset selection methods. For this comparison, we chose the following five methods: *mRMR* [13], *Mutual Information Measurement* (*MI*) [28], *I-RELIEF* [14], *INTERACT* [16], *and PAM* [17]. These methods have been recently introduced and demonstrated as competitive feature subset selection methods. The five methods were used as a filter to inductive learning algorithms to construct the classifier. In other words, each method produced a feature subset based on its selection criterion that would be used in the learning algorithm (See the references and our previous publications [20, 21] for detailed descriptions on the selection methods and preliminary experimental results.) So we compared the performance of seven approaches (*i.e. GAFSS, PSOFSS, MI, mRMR, I-RELIEF, INTERACT, PAM*). We compared them in terms of the number of selected features and the classification ac-

Table 1. Datasets.

Name	Features	Classes	Samples
Audiology	69	24	226
Dermatology	33	6	366
Musk	166	2	475
Spambase	57	2	4601
Arrhythmia	277	16	452
Ionosphere	34	2	351
Waveform	21	3	5000
Sonar	60	2	208
Image Segmentation	19	7	2310
Flag	28	6	194
Hepatitis	19	2	155
Lung Cancer	56	3	32
Promoter	56	2	106
Splice	60	3	3190
Optdigits	64	10	3823
SpectF	44	2	267
Connect-4	42	3	691
Water Treatment	38	13	527
Isolet	617	26	6236
HDR Multifeature	649	10	2000

curacy with the feature subset. For the five filter approaches, we reorganized the datasets using selected (reduced) features by each method on which the learning algorithms were applied. The number of selected features for the filter approaches was determined as follows: For *MI*, *mRMR*, and *I-RELIEF*, features were added to the feature subset in an incremental way (starting from an empty set) with respect to their selection criteria and the best feature subset (with the highest classification accuracy) was produced; for *INTERACT* and *PAM*, features were selected as suggested by the authors in the references and available software. As mentioned earlier, we went through twenty iterations (*i.e.* generations) with forty individuals (*i.e.* chromosomes or particles) for the two bio-inspired approaches. This work is an extension of our previous research in [3, 19] where the former is a *GA*-based approach that is similar to *GAFSS*, and the latter is an ACO-based approach that produced fairly comparable performance to *GAFSS*. Note that the main contribution of this paper is the introduction of *GAFSS + mRMR* and *PSOFSS + mRMR* which are improvements over *GAFSS* and *PSOFSS*.

In the second experiment, we compared the performance of *GAFSS* and *PSOFSS* with their variants (*i.e. GAFSS + mRMR*, *PSOFSS + mRMR*). For the variants, the same number of iterations, the size of population, and other parameters were used as in *GAFSS* and *PSOFSS*. In addition to the feature subset size and the classification accuracy, we also compared the learning curves in this experiment. This is to check the evolution speed and to figure out how fast each approach converges to the best solution.

In the third experiment, we explored the effectiveness of our approaches in selecting an appropriate subset of relevant features in the presence of redundant or useless features so as to maximize the accuracy of the resulting classifiers. Since it is not so easy to judge the usefulness of the features in real-world datasets without domain knowledge, we chose

one of the common artificial datasets, *parity* (3-bit and 6-bit), constructed as follows: The original features are replicated once (to introduce redundancy) thereby doubling the number of features. Then an additional set of irrelevant features are generated and are assigned random boolean values. For 3-bit parity, 100 7-bit random vectors were generated and augmented with the 6-bit vectors (corresponding to the original 3 bits plus and identical set of 3 bits). In the same way, 100 12-bit random vectors were generated and augmented with 12-bit vectors (6 original bits and 6 identical bits) for 6-bit parity.

We used Weka [31] for our experiments. Weka is a well-known open-source data mining software based on JAVA, equipped with a variety of machine learning algorithms. Among those algorithms [27-29], three popular ones are adopted in our experiments: Naïve Bayes (NB) [32], Decision Tree (C4.5) [33], and Support Vector Machines (SVM) [34] with linear kernel. (See the references for detailed descriptions on the algorithms.) We conducted 10-fold cross-validation in all experiments.

5. RESULTS

5.1 Comparison Between Bio-Inspired Methods and Existing Methods

First, we computed the classification accuracy of data with all features (*i.e*. full feature set) using the three learning algorithms. The results are shown in Table 2. Then we carried out feature subset selection on the datasets using the seven methods (shown in Tables 3-5 with the classification accuracy and the feature subset size), and compared their performance with the results in Table 2.

Table 2. Classification accuracies with all features (%).

Name	Acc (NB)	Acc (C4.5)	Acc (SVM)
Audiology	73.45	77.88	81.86
Dermatology	97.54	90.98	95.36
Musk	73.68	78.95	82.74
Spambase	79.29	92.61	90.42
Arrhythmia	59.07	66.37	67.70
Ionosphere	84.62	88.60	89.17
Waveform	78.20	75.60	81.22
Sonar	65.38	69.23	73.56
Image Segmentation	79.87	87.84	85.80
Flag	56.19	70.62	64.43
Hepatitis	85.16	79.35	85.81
Lung Cancer	62.50	59.38	50.00
Promoter	82.08	82.08	82.08
Splice	91.25	92.60	84.55
Optdigits	91.66	89.62	98.14
SpectF	68.54	73.78	79.18
Connect-4	54.85	63.68	63.82
Water Treatment	74.76	70.02	78.75
Isolet	84.35	83.24	96.81
HDR Multifeature	95.35	94.60	98.40
Average	76.89	79.35	81.49

Table 3. Classification accuracies of feature subset selection methods (%)-NB.

NB	MI		*mRMR*		I-RELIEF		INTERACT		PAM		*GAFSS*		*PSOFSS*	
	Acc	#F	Acc	#F	Acc	#F	Acc	#F	Acc	#F	Acc	#F	Acc	#F
Audiology	75.7	21	75.7	21	75.2	45	72.6	21	74.8	39	76.6	39	**78.3**	36
Dermatology	98.4	23	98.6	24	98.1	24	97.3	18	97.5	33	**99.2**	25	**99.2**	15
Musk	76.8	55	80.2	80	75.4	69	71.6	16	75.4	61	79.6	66	**81.7**	74
Spambase	90.1	30	**90.4**	22	84.4	15	82.6	30	79.8	46	89.5	29	90.1	26
Arrhythmia	68.4	41	69.9	54	67.5	16	68.1	23	63.5	124	67.7	88	**70.7**	119
Ionosphere	89.2	13	90.3	12	88.6	18	88.3	18	85.5	24	91.0	15	**91.7**	16
Waveform	78.2	16	78.6	16	78.6	16	78.3	19	78.3	19	79.4	15	**79.5**	13
Sonar	69.7	6	**79.2**	9	74.5	17	69.2	17	70.2	8	75.5	25	76.9	15
Image Segmentation	80.8	13	82.4	6	81.3	17	82.6	13	80.7	12	**84.9**	9	**84.9**	10
Flag	68.0	6	69.6	4	69.6	4	58.3	9	56.2	16	70.6	11	**72.2**	14
Hepatitis	84.5	16	86.5	4	84.5	17	83.9	9	83.2	11	89.7	8	**90.3**	7
Lung Cancer	78.1	5	84.4	6	75.0	10	75.0	8	62.5	22	84.4	28	**87.5**	21
Promoter	91.5	23	**98.1**	40	**98.1**	38	89.6	13	85.9	30	92.5	33	94.3	27
Splice	**91.7**	43	91.7	31	91.7	53	91.0	31	91.4	57	91.4	33	91.6	38
Optdigits	91.6	63	92.4	45	92.5	45	91.6	21	92.3	47	92.7	36	**93.2**	33
SpectF	74.9	3	74.9	9	74.9	3	73.0	15	71.2	26	**79.4**	1	79.0	7
Connect-4	65.3	6	65.4	4	65.0	4	63.7	9	54.9	40	66.	12	**68.7**	11
Water Treatment	74.2	32	75.9	22	75.9	17	75.3	16	73.2	32	78.9	15	**79.3**	20
Isolet	84.4	357	82.6	495	X	X	85.3	57	84.7	594	89.1	284	**89.6**	284
HDR Multifeature	95.6	420	95.4	402	X	X	96.6	131	95.1	564	96.4	313	**96.7**	301
Average	81.4	–	83.1	–	80.6	–	79.7	–	77.8	–	83.7	–	**84.8**	–

Table 4. Classification accuracies of feature subset selection methods (%)-C4.5.

C4.5	MI		*mRMR*		I-RELIEF		INTERACT		PAM		*GAFSS*		*PSOFSS*	
	Acc	#F	Acc	#F	Acc	#F	Acc	#F	Acc	#F	Acc	#F	Acc	#F
Audiology	77.9	21	77.9	21	77.9	26	77.9	21	77.9	39	**78.3**	33	**78.3**	41
Dermatology	93.2	22	92.9	23	92.9	23	92.4	18	91.0	33	**95.9**	11	**95.9**	19
Musk	82.7	89	82.7	76	83.2	131	77.1	16	81.1	61	84.0	84	**88.0**	79
Spambase	93.0	17	93.3	26	93.2	49	92.8	30	92.9	46	93.1	37	**93.8**	26
Arrhythmia	70.6	21	70.6	18	69.7	34	70.1	23	67.7	124	72.4	111	**73.2**	112
Ionosphere	89.2	28	89.7	27	88.6	18	86.9	18	88.6	24	**92.0**	14	91.2	15
Waveform	76.4	11	76.2	15	76.4	15	75.7	19	75.8	19	76.8	15	**77.6**	13
Sonar	78.4	11	80.3	8	74.5	3	76.4	17	71.2	8	78.9	23	**84.1**	24
Image Segmentation	88.1	14	88.1	14	88.1	14	87.8	13	87.7	12	88.1	14	**88.2**	11
Flag	73.2	3	73.7	6	73.2	8	73.7	9	73.2	16	73.7	9	**74.2**	12
Hepatitis	81.9	13	83.2	1	81.9	12	82.6	9	81.9	11	**85.8**	5	**85.8**	4
Lung Cancer	68.8	5	59.4	5	65.6	6	62.5	8	59.4	22	**71.9**	5	**71.9**	25
Promoter	87.7	4	**100**	38	**100**	38	83.0	13	72.6	30	86.8	12	86.8	25
Splice	94.4	12	94.4	13	94.5	14	93.2	31	92.7	57	94.5	33	**94.7**	27
Optdigits	89.8	60	89.7	56	89.6	53	89.8	21	89.4	47	90.0	33	**90.4**	35
SpectF	79.4	1	82.4	20	79.4	2	74.2	15	74.5	26	83.5	10	**86.1**	15
Connect-4	67.0	3	66.7	6	67.4	2	65.0	9	64.0	40	70.2	16	**70.3**	19
Water Treatment	71.2	36	72.3	24	73.4	11	68.9	16	66.8	32	72.9	19	**76.3**	19
Isolet	80.3	429	83.1	273	X	X	80.4	57	83.1	594	83.6	325	**84.6**	313
HDR Multifeature	94.1	458	94.9	55	X	X	93.7	131	93.8	564	96.4	312	**96.7**	320
Average	81.9	–	82.6	–	81.6	–	80.2	–	79.3	–	83.4	–	**84.4**	–

Tables 3-5 are with the NB, C4.5, and SVM algorithms, respectively. The highest classification accuracy is underlined and bold-faced for each dataset. Note that the X's in those tables are due to the limited capability of *I-RELIEF* in subset selection for high-

Table 5. Classification accuracies of feature subset selection methods (%) – SVM.

SVM	MI		*mRMR*		I-RELIEF		INTERACT		PAM		*GAFSS*		*PSOFSS*	
	Acc	#F	Acc	#F	Acc	#F	Acc	#F	Acc	#F	Acc	#F	Acc	#F
Audiology	81.9	48	82.3	46	82.3	49	76.1	21	78.3	39	**85.8**	43	**85.8**	42
Dermatology	98.1	23	97.5	24	97.5	24	96.7	18	95.4	33	98.6	22	**98.9**	19
Musk	83.8	110	84.8	153	83.4	157	71.2	16	77.5	61	85.3	95	**85.9**	79
Spambase	90.4	54	**90.5**	54	**90.5**	56	88.4	30	89.9	46	90.1	44	90.4	36
Arrhythmia	70.8	30	71.5	27	71.2	45	69.7	23	69.5	124	73.0	133	**74.3**	128
Ionosphere	89.2	17	89.2	25	89.2	33	88.3	18	87.2	24	90.6	19	**92.3**	17
Waveform	81.4	18	81.6	19	81.6	18	81.4	19	81.4	19	81.4	19	**81.6**	18
Sonar	77.4	23	80.3	15	78.4	33	75.0	17	75.0	8	81.7	21	**84.6**	28
Image Segmentation	85.8	18	85.8	16	85.8	17	85.1	13	85.0	12	86.1	13	**86.2**	16
Flag	63.9	18	66.5	23	66.0	21	57.2	9	62.4	16	66.5	17	**69.1**	22
Hepatitis	83.9	15	85.2	13	83.9	9	84.5	9	82.6	11	88.4	9	**89.7**	13
Lung Cancer	75.0	7	78.1	38	81.3	25	65.6	8	75.0	22	78.1	26	**87.5**	18
Promoter	86.8	18	**100**	38	**100**	38	88.7	13	80.2	30	96.2	33	95.3	31
Splice	85.0	44	84.8	21	85.0	18	84.5	31	84.5	57	**85.4**	39	85.2	34
Optdigits	98.2	63	98.3	47	98.3	45	96.2	21	98.2	47	98.2	44	**98.3**	44
SpectF	**79.8**	36	**79.8**	38	79.4	1	79.4	15	79.4	26	79.4	1	79.4	8
Connect-4	64.7	1	64.7	5	64.7	3	64.3	9	63.8	40	**64.**	20	**64.8**	21
Water Treatment	78.8	36	79.1	36	78.6	26	73.4	16	76.3	32	79.5	26	**79.9**	27
Isolet	94.8	354	96.9	402	X	X	92.8	57	96.7	594	**96.8**	325	**96.8**	313
HDR Multifeature	98.5	460	98.5	410	X	X	98.5	131	98.3	564	99.1	312	**99.5**	320
Average	83.4	–	84.8	–	83.2	–	80.9	–	81.8	–	85.2	–	**86.3**	–

dimensional datasets like Isolet and HDR Multifeature.

As we can see from the tables, most of the methods produced better performance with a reduced feature set than with the full feature set, regardless of the learning algorithm. In particular, *GAFSS* and *PSOFSS* showed improved performance than filter approaches for most of the datasets. Furthermore, *PSOFSS* performed better than *GAFSS* for majority of the datasets with the three learning algorithms, producing higher average accuracy over all datasets. Therefore, we can conclude that our bio-inspired approaches, combined with popular learning algorithms, are very effective for feature subset selection, and *PSOFSS* is particularly the best approach for the task.

As far as the learning time, filter approaches were faster due to its non-iterative characteristic. Precisely, *GAFSS* and *PSOFSS* are slower than *mRMR* proportional to the size of the population and the number of generations (*e.g.* $40 \times 20 = 800$ in our experiments). However, this problem can be mitigated by one-time, offline learning that selects the optimal feature subset with the best classification accuracy. In addition, the evolutionary approaches produced best solutions, swiftly converging to quality solutions in a small number of generations as shown in section 5.2. This verifies the practicality of our proposed approaches.

5.2 Comparison Between Bio-Inspired Methods and Their Variants

After the experiments were conducted under the same experimental setup as explained in section 4, the results of bio-inspired methods and their variants with *mRMR* are summarized in Tables 6-8 for the three learning algorithms. *mRMR* was chosen since its performance was either better than or comparable to other filter methods as shown Tables 3-5.

Table 6. Performance comparison of *GAFSS* and *PSOFSS* with their variants-NB.

NB	*GAFSS*		*GAFSS + mRMR*		*PSOFSS*		*PSOFSS + mRMR*	
	Acc	#F	Acc	#F	Acc	#F	Acc	#F
Audiology	76.55	39	**77.43**	36	**78.32**	36	77.88	33
Dermatology	99.18	25	99.18	22	99.18	15	**99.45**	19
Musk	79.58	66	**83.37**	53	81.68	74	**84.00**	72
Spambase	**89.52**	29	89.11	30	90.11	26	**91.31**	21
Arrhythmia	67.70	88	**68.14**	99	69.69	119	**72.35**	95
Ionosphere	90.88	15	**91.45**	17	91.74	16	**92.02**	18
Waveform	**79.36**	15	79.14	17	79.48	13	79.48	16
Sonar	75.48	25	**76.44**	26	76.92	15	**79.81**	23
Image Segmentation	84.85	9	84.85	9	84.85	10	84.85	10
Flag	**70.62**	11	70.10	18	72.16	14	**73.20**	11
Hepatitis	89.68	8	89.68	8	90.32	7	90.32	10
Lung Cancer	84.38	28	84.38	21	87.50	21	**90.63**	25
Promoter	92.45	33	**93.40**	32	94.34	27	**95.28**	30
Splice	91.38	33	**91.44**	35	91.63	38	**91.97**	34
Optdigits	92.73	36	**93.60**	39	93.23	33	**93.70**	39
SpectF	79.40	1	79.40	1	79.03	7	**79.40**	10
Connect-4	66.71	12	66.70	11	68.74	11	**69.03**	11
Water Treatment	78.94	15	**79.13**	22	**79.32**	20	77.99	21
Isolet	89.10	284	**89.57**	321	89.64	284	89.64	297
HDR Multifeature	**96.35**	313	96.25	370	96.70	301	**97.30**	286
Average	83.74	–	84.14	–	84.73	–	**85.48**	–

Table 7. Performance comparison of *GAFSS* and *PSOFSS* with their variants-C4.5.

C4.5	*GAFSS*		*GAFSS + mRMR*		*PSOFSS*		*PSOFSS + mRMR*	
	Acc	#F	Acc	#F	Acc	#F	Acc	#F
Audiology	78.32	33	78.32	33	78.32	41	78.32	36
Dermatology	95.90	11	95.90	14	95.90	19	95.90	15
Musk	84.00	84	**85.26**	74	88.00	79	**89.26**	82
Spambase	93.13	37	**93.70**	32	**93.81**	26	93.61	25
Arrhythmia	72.35	111	72.35	126	73.23	112	73.23	118
Ionosphere	92.02	14	92.02	11	91.17	15	**92.31**	8
Waveform	76.84	15	76.84	12	77.64	13	77.64	13
Sonar	78.85	23	78.85	23	84.13	24	84.13	27
Image Segmentation	88.10	14	**88.14**	15	88.23	11	88.23	12
Flag	73.71	9	**74.23**	6	74.23	12	**75.26**	13
Hepatitis	**85.81**	5	84.52	5	85.81	4	85.81	6
Lung Cancer	71.88	5	71.88	16	71.88	25	71.88	22
Promoter	86.79	12	86.79	17	86.79	25	86.79	26
Splice	**94.48**	33	93.98	32	**94.70**	27	94.61	22
Optdigits	**90.03**	33	89.69	36	90.40	35	**91.43**	35
SpectF	83.52	10	83.52	10	**86.14**	15	84.64	5
Connect-4	70.19	16	**70.62**	20	70.33	19	**70.62**	19
Water Treatment	72.87	19	**73.06**	21	**76.28**	19	74.95	20
Isolet	83.55	325	**85.55**	321	84.56	313	**86.57**	310
HDR Multifeature	96.40	312	**97.10**	385	96.65	320	96.65	340
Average	83.44	–	83.62	–	84.41	–	**84.59**	–

In most cases, the modified methods using *mRMR* produced higher learning accuracies than plain *GAFSS* and *PSOFSS*. And overall, *PSOFSS + mRMR* showed the best performance among all the methods we considered.

Table 8. Performance comparison of *GAFSS* and *PSOFSS* with their variant – SVM.

SVM	*GAFSS*		*GAFSS + mRMR*		*PSOFSS*		*PSOFSS + mRMR*	
	Acc	#F	Acc	#F	Acc	#F	Acc	#F
Audiology	85.84	43	85.84	43	85.84	42	**87.15**	39
Dermatology	98.63	22	**98.91**	18	98.91	19	98.91	17
Musk	**85.26**	95	85.05	97	85.89	79	85.89	86
Spambase	90.09	44	**90.39**	39	90.39	36	**90.48**	36
Arrhythmia	73.01	133	**75.45**	158	74.25	128	**75.45**	158
Ionosphere	**90.60**	19	90.31	17	**92.31**	17	91.45	15
Waveform	81.42	19	81.42	19	81.60	18	81.60	18
Sonar	81.73	21	**82.21**	26	**84.62**	28	83.65	31
Image Segmentation	86.10	13	86.10	17	86.15	16	86.15	14
Flag	66.49	17	**69.55**	15	69.07	22	**71.65**	17
Hepatitis	**88.39**	9	86.45	11	89.68	13	89.68	10
Lung Cancer	78.13	26	**84.38**	22	87.50	18	87.50	26
Promoter	**96.23**	33	93.40	32	95.28	31	95.28	35
Splice	85.36	39	85.36	33	85.17	34	**85.30**	31
Optdigits	98.17	44	**98.22**	42	98.33	44	**98.38**	47
SpectF	79.40	1	79.40	1	79.40	8	79.40	21
Connect-4	64.83	20	64.83	20	64.83	21	64.83	16
Water Treatment	79.51	27	**80.08**	27	79.89	27	**81.21**	24
Isolet	96.75	356	**96.90**	356	96.75	313	**97.43**	324
HDR Multifeature	99.10	315	**99.45**	315	99.53	320	99.53	325
Average	85.25	–	85.69	–	86.27	–	**86.55**	–

In addition to the analysis on the accuracy, we also compared the learning speed in order to see how fast the algorithm converges to the best solution. For *PSOFSS* and *PSOFSS + mRMR*, we chose four datasets and derived the learning curves (*i.e.* trace of the current best accuracy) as shown in Fig. 1. The graphs in Fig. 1 indicate that the accuracy of *PSOFSS + mRMR* increases faster than *PSOFSS*. This shows another merit of the modified method that it quickly finds the optimal feature subset.

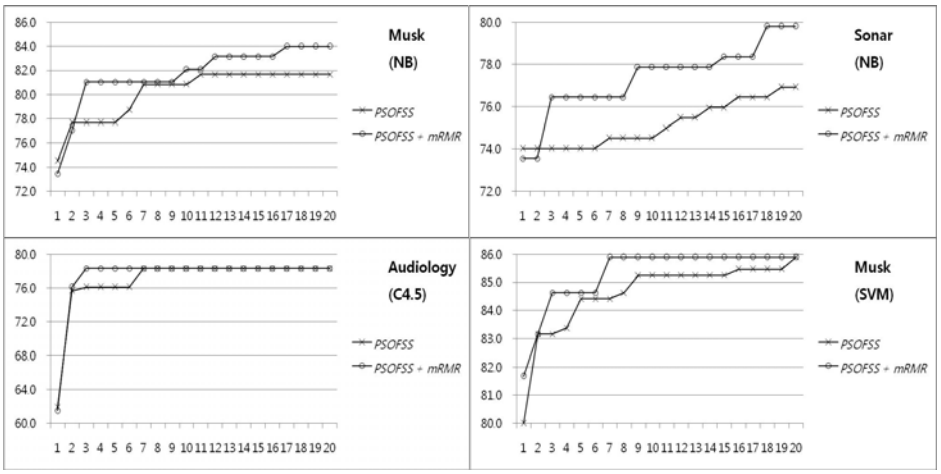

Fig. 1. Comparisons of evolution speed between *PSOFSS* and *PSOFSS + mRMR*. The horizontal axis means the number of iterations, and the vertical axis indicates the learning accuracy (%).

5.3 Quality of Feature Subsets

Table 9 compares the performance of *mRMR* and *PSOFSS + mRMR* on the parity datasets (*#rel* means the number of relevant features in the selected subset of size *#F*). While the subsets produced by *mRMR* included irrelevant features which caused low classification accuracy, *PSOFSS + mRMR* successfully found appropriate sets of features with high accuracy. This demonstrates the outstanding capability of our approach in feature subset selection.

Table 9. Feature subset comparison of *mRMR* and *PSOFSS + mRMR*.

Dataset	Classifier	*mRMR*			*PSOFSS + mRMR*		
		Acc	#F	#rel	Acc	#F	#rel
3-bit Parity	NB	57.00	1	0	**59.00**	4	1
	C4.5	58.00	2	0	**93.00**	3	3
	SVM	**57.00**	1	0	**57.00**	2	1
6-bit Parity	NB	**54.69**	1	0	**54.69**	4	3
	C4.5	56.25	12	4	**62.81**	7	3
	SVM	54.69	1	0	**55.00**	6	4

6. CONCLUSION AND FUTURE WORK

In this paper, we designed efficient methods, *GAFSS* and *PSOFSS*, for the feature subset selection task based on bio-inspired algorithms such as the genetic algorithm and the particle swarm optimization. We also proposed variants of those approaches considering the relevance of each feature, which produced novel algorithms called *GAFSS + mRMR* and *PSOFSS + mRMR*.

The performance of *GAFSS* and *PSOFSS* are experimentally verified to outperform other state-of-the-art feature selection methods in terms of the classification accuracy and the quality of the feature subsets. Furthermore, *GAFSS + mRMR* and *PSOFSS + mRMR* demonstrated more improved performance over *GAFSS* and *PSOFSS* making use of the relevance information, in terms of both the learning accuracy and the evolution speed. Extensive experiments on various real-world data concluded the best performance with *PSOFSS + mRMR*.

There are some avenues for future research. First, the relevance measure we considered in this paper is *mRMR* which maximizes the relevance and minimizes the redundancy of features using the mutual information criterion. Based on *mRMR*, the data is divided into three different subsets which affect the feature selection process. Further studies on this procedure can be pursued. Also, research on various relevance measures can lead to the improvement of current algorithms.

Second, we might need to consider more than a single criterion for the fitness function. For instance, we can consider the cost (or risk) associated with each feature as well, and attempt to minimize the total cost while maximizing the learning accuracy. This is basically multi-objective optimization, which occurs frequently in our daily lives (*e.g.* medical diagnosis). Solving a multi-objective optimization problem in the feature subset selection task using bio-inspired algorithms is clearly of interest.

Third, we can apply our idea to other existing bio-inspired algorithms (*e.g.* Ant Colony Optimization (ACO) [24-26]) or develop new bio-inspired algorithms.

Fourth, we can develop hybrid approaches for the feature subset selection task. While the bio-inspired algorithms are very successful in finding good solutions, they may require quite amount of time for evolution. We might as well develop hybrid approaches (*e.g.* combining bio-inspired approaches with filter approaches or simple randomized search) to reduce the computational overhead.

Lastly, we can look into the feature subsets produced by different approaches and elicit the optimal subset by coalescing them based on the consensus among the approaches (*e.g.* ensemble learning).

REFERENCES

1. W. Siedlecki and J. Sklansky, "On automatic feature selection," *International Journal of Pattern Recognition*, Vol. 2, 1988, pp. 197-220.
2. A. Jain and D. Zongker, "Feature selection: evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, 1997, pp. 153-158.
3. J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," *IEEE Intelligent Systems*, Vol. 13, 1998, pp. 44-49.
4. H. Almuallim and T. Dietterich, "Learning Boolean concepts in the presence of many irrelevant features," *Artificial Intelligence*, Vol. 69, 1994, pp. 279-305.
5. J. Sheinvald, B. Dom, and W. Niblack, "A modeling approach to feature selection," in *Proceedings of the 10th International Conference on Pattern Recognition*, 1990, pp. 535-539.
6. P. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Transactions on Computers*, Vol. 26, 1977, pp. 917-922.
7. I. Foroutan and J. Sklansky, "Feature selection for automatic classification of non-gaussian data," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 17, 1987, pp. 187-198.
8. D. Koller and M. Sahami, "Hierarchically classifying documents using very few words," in *Proceedings of International Conference on Machine Learning*, 1997, pp. 535-539.
9. M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, Vol. 1, 1997, pp. 131-156.
10. A. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, Vol. 97, 1997, pp. 245-271.
11. I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, Vol. 3, 2003, pp. 1157-1182.
12. H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, 2005, pp. 491-502.
13. H. Peng, F. Long, and 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*, Vol. 27, 2005, pp. 1226-1238.

14. Y. Sun and J. Li, "Iterative RELIEF for feature weighting: algorithms, theories, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, 2007, pp. 1035-1051.
15. G. Claeskens, C. Croux, and J. Kerckhoven, "An information criterion for variable selection in support vector machines," *Journal of Machine Learning Research*, Vol. 9, 2008, pp. 541-558.
16. Z. Zhao and H. Liu, "Searching for interacting features," in *Proceedings of the 2nd International Joint Conference on Artificial Intelligence*, 2007, pp. 1156-1161.
17. R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Diagnosis of multiple cancer types by shrunken centroids of gene expression," in *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 99, 2002, pp. 6567-6572.
18. W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, Vol. 10, 1989, pp. 335-347.
19. K. Lee, J. Joo, J. Yang, and V. Honavar, "Experimental comparison of feature subset selection using a GA and ACO algorithm," in *Proceedings of the 2nd International Conference on Advanced Data Mining and Applications*, 2006, pp. 465-472.
20. C. Yun, D. Shin, H. Jo, J. Yang, and S. Kim, "An experimental study on feature subset selection methods," in *Proceedings of the 7th International Conference on Computer and Information Technology*, 2007, pp. 77-82.
21. C. Yun and J. Yang, "Experimental comparisons of feature subset selection methods," in *Proceedings of Workshops on International Conference on Data Mining*, 2007, pp. 367-372.
22. J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of International Conference on Neural Networks*, 1995, pp. 1942-1948.
23. M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, 1996.
24. J. Kennedy and R. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, San Francisco, 2001.
25. S. Olariu and A. Y. Zomaya, *Handbook of Bioinspired Algorithms and Applications*, CRC Press, Boca Raton, 2006.
26. M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, Cambridge, 2004.
27. T. Mitchell, *Machine Learning*, McGraw Hill, New York, 1997.
28. R. Duda, P. Hart, and D. Stork, *Pattern Classification*, Wiley-Interscience, New York, 2000.
29. C. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.
30. A. Asuncion and D. Newman, UCI Machine Learning Repository, School of Information and Computer Science, University of California, 2007, http://www.ics.uci.edu/~mlearn/MLRepository.html.
31. I. Witten and E. Frank, *Data Mining*: *Practical Machine Learning Tools and Techniques*, 2nd ed., Morgan Kaufmann, San Francisco, 2005, http://www.cs.waikato.ac.nz/ml/weka, Weka software.
32. P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," *Machine Learning*, Vol. 29, 1997, pp. 103-137.
33. J. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco, 1993.
34. V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1995.

Chulmin Yun (尹哲敏) received his M.S. degree in Engineering from Sogang University, Seoul, Korea, in 2008. He was a graduate student and a member of Data Mining Laboratory in Computer Science and Engineering Department from 2006 to 2008. He is currently a research engineer at Diquest Inc., Seoul, Korea. His research interests focus on machine learning, pattern analysis and computational intelligence. In particular, he is interested in feature selection, evolutionary algorithms, and link mining.

Byonghwa Oh (吳炳華) received the M.S. degree in Engineering from Sogang University, Seoul, Korea, in 2009. He is currently working toward his Ph.D. degree in Computer Science at the same university. He has been a member of Data Mining Research Laboratory in Computer Science and Engineering Department from 2007 to present. His research interests focus on machine learning, computational intelligence, and data mining. In particular, he is interested in evolutionary algorithms, reinforcement learning and semi-supervised learning.

Jihoon Yang (楊枝勳) is Professor of Computer Science and Engineering Department at Sogang University. His research interests include machine learning, data mining and knowledge discovery, artificial intelligence, pattern recognition, evolutionary computation, and bioinformatics. He holds a B.S. in Computer Science from Sogang University, and M.S. and Ph.D. degrees in Computer Science from Iowa State University.

Jongho Nang (浪鍾鎬) received his Ph.D. and M.S. degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Daejon, Korea, in 1992 and 1988, respectively, and his B.S. degree in Computer Science from Sogang University, Seoul, Korea, in 1986. He has been a Professor of Computer Science and Engineering Department, Sogang University since 1993. His research interests include multimedia system, parallel processing, and internet technology.