COMB-PSO for selecting the smallest subsets of genes with highest classification performance: a case study of Alzheimer's disease

No Author Given

No Institute Given

Abstract. If a biomarker can accurately diagnose Alzheimer's disease (AD) at early stages, it can help developing treatment for early intervention. Nonetheless, the less biomarkers we identify, the easier and more reliable the diagnosis will be. But, due to the small number of samples compared to the huge number of genes (high dimension) with irrelevant and noisy genes, many of the computational techniques have difficulties selecting small subsets with high classification performance. Particle swarm optimization (PSO) algorithms have shown promising results in this active research area. However, preservation of the population diversity to avoid premature convergence has been a long-standing challenge for PSO. In this paper, we present a novel PSO variant based on a combination of continuous and binary (COMB-PSO) model and we introduce a new encoding mechanism to allow the swarm to converge faster to very small feature subsets. We also introduce a diversity preservation strategy to make the swarm investigate unexplored areas of the search space and avoid premature convergence. Additionally, a new inertia weight is suggested, which creates a dynamic balance between exploration and exploitation. None of the proposed methods require new parameters or alter the PSO's principal quality: its simplicity. We experiment with a real gene expression dataset from blood samples for early AD diagnosis, and we find that the proposed method can converge to subsets at least 5-times smaller than the standard binary PSO while displaying high classification performance.

1 Introduction

With an aging population, and age being the biggest risk factor for Alzheimer's, and estimated 135 million people are expected to have dementia by 2050. Presently, there is no effective long-lasting drug treatments for Alzheimer's, since the brain is already severely affected by the time patients are diagnosed with the disease. A blood test, which is substantially easier and less expensive than brain images or spinal fluid samples, could be used for early diagnosis. Nonetheless, we need to study large well-characterized cohorts of patients with multiple markers and consolidate the findings to produce the most accurate diagnostic. Experts believe that biomarkers (short for" biological markers") offer an encouraging path. A

biomarker is a precise and reliable indicator of the presence of disease. An example of a biomarker is fasting blood glucose (blood sugar) level, which indicates the presence of diabetes if it is 126mg/dL or higher. For the treating medical doctor, the less biomarkers there are, the easier and more accurate the diagnosis will be.

However, for small size of samples (few hundreds) in comparison to the high dimensionality (tens of thousands) of gene expressions, it is difficult to implement a particular biological classification problem as well as gain deeper understanding of the functions of particular genes. Gene selection is one of the critical steps in the process of classification of microarray data [1–9]. Feature selection is multi-objective optimization problem, which aims to maximize the prediction accuracy and minimize the cardinality of the genes subset. Feature selection methods were broadly classified into *filter*, *wrapper* and *embedded* methods. Filter methods act as preprocessing to rank features wherein the highly ranked features are selected and applied to a predictor. In wrapper methods, the features selection criteria is the performance of the predictor, i.e. the predictor is wrapped on a search algorithm which will find a subset which gives the predictor's performance. Embedded methods include variable selection as part of the training process without splitting the data into train and test sets. Feature selection is not to be confused with feature extraction (or construction) [10–13] or other dimensionality reduction methods such as PCA. The major difference is that feature selection returns only subsets of original features, while feature extraction creates new features. This paper focuses only on feature selection.

Particle swarm optimization (PSO) is an evolutionary computation technique proposed by Kennedy and Eberhart [14]. It has shown promising results in recent years when wrapped with classifiers such as SVM or RF, and applied to feature selection. The particle swarm concept was inspired by studies on the social behavior of biological organisms. The PSO algorithm mimics the behavior of flying birds and their means of information exchange to solve optimization problems. We can use the idea of PSO for the optimal feature selection problem [15]. Consider a large feature space, where each feature subset can be seen as a point or position in such a space. If there are d total features, then there will be 2^d kinds of subsets, different from each others in the length and features contained in each subset. The optimal position is the subset of smallest size and highest classification quality. Now we put a particle swarm into this feature space, each particle takes one position. The particles fly in this space, their goal is to fly to the best position. Over time, they change their position, communicate with each other, and search around the local best and global best position. Eventually, they should converge on good, possibly optimal, positions. It is this exploration ability of particle swarms that should better equip it to perform feature selection and discover optimal subsets [16].

In this paper, we propose a novel continuous-binary PSO (COMB-PSO) to select a small (near-optimal) subset of informative genes that is most relevant for the Alzheimer's disease (AD) classification. We introduce a new encoding scheme

and a new diversity preservation strategy. We go over the parameters that influence the performance of our algorithm and we propose novel settings for the inertia weight. To test the effectiveness of our approach, we apply COMB-PSO to real clinic AD gene expression datasets provided by King's College in London and Oxford University. The related work are briefly described in section 2. The proposed gene selection method is introduced in section 4. Section 5 gives the experimental results on real life microarray dataset. Finally, the concluding remarks are offered in section 6.

2 Related work

In recent years, particle swarm optimization (PSO), has been used increasingly as an effective technique for global optimization in areas such as microarray data analysis [5, 6, 17]. Compared with genetic algorithm [9, 11], PSO has no complicated evolutionary operators and fewer parameters need to be adjusted. In [18], a combination of integer-coded GA (ICGA) and particle swarm optimization was used to select an optimal set of genes. In [3, 5, 7], BPSO combined with filter method was applied for searching optimal gene subsets. The method in [19], simplified gene selection and obtained a higher classification accuracy compared with some similar gene selection methods based on GA, while the method in [1] could determine the appropriate number of genes and obtained high classification accuracy by support vector machine (SVM). These hybrid methods were capable of selecting a compact subset of predictive genes for sample classification. However, all PSO algorithms including BPSO are easy to lose the diversity of the swarm, which may lead to premature convergence. One main reason for premature convergence is that the updates of the particles have not been guided by some effective prior information of the problems. In [20], the authors presented a Tabou-Search TS-PSO, where particles divided into two swarms, the first for the local search and the second for the global search. The idea is to use a second swarm to push away from local best each particle entering the tabou condition (i.e. getting trapped in local best).

In this paper, to improve prediction accuracy as well as overcome the deficiencies of premature convergence , we propose an improved PSO feature selection method combining both continuous and discrete sub-methods and introduce a new encoding technique, a diversity preservation strategy and a parameter optimization mechanism to allow faster convergence towards smaller gene subsets with higher classification performance.

3 The standard PSO and BPSO algorithms

Particle swarm optimization (PSO) is a population based stochastic optimization technique inspired by studies on the social behavior of biological organisms. In

PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Consider a large feature space, where each feature subset can be seen as a point or position in such a space. The optimal position is the subset of smallest size and highest classification quality. Now we put a particle swarm into this feature space, each particle takes one position. PSO comes in two approaches: continuous (PSO) [14] and binary (BPSO) [21]. Both approaches can be applied to solve feature selection problems in a wrapped/embedded method with classification algorithms such as SVM or random forest (RF). The PSO algorithms are non-deterministic resulting into multiple local minima.

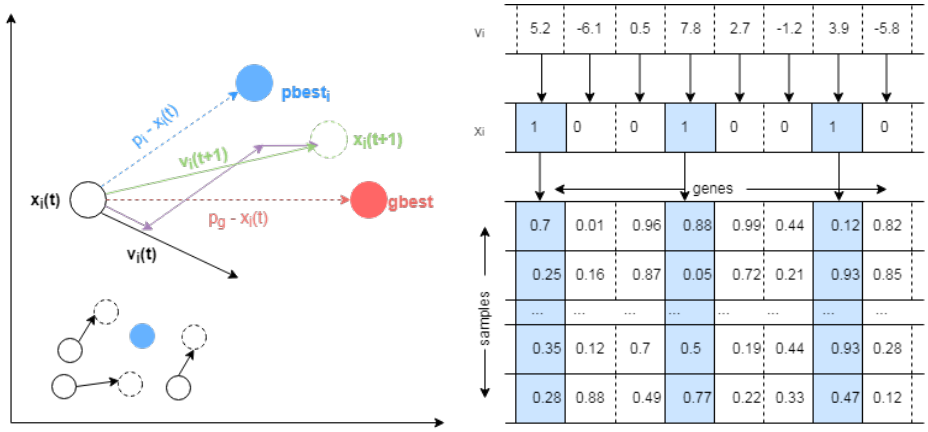
The PSO algorithm is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. This value is called *pbest*. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called *gbest*. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest. After finding the two best values, the particle updates its velocity and positions with following eq. (1) and eq. (2) respectively.

$$\overrightarrow{v}_i = w\overrightarrow{v}_i + c1\overrightarrow{R1}(\overrightarrow{p}_i - \overrightarrow{x}_i) + c2\overrightarrow{R2}(\overrightarrow{g} - \overrightarrow{x}_i) \tag{1}$$

$$\overrightarrow{x}_i = \overrightarrow{x}_i + \overrightarrow{v}_i \tag{2}$$

where $\overrightarrow{v}_i$ and $\overrightarrow{x}_i \in \mathbb{R}^d, 1 \leq i \leq P$ are the i^{th} particle's velocity and position respectively, P is the size of the swarm, d is the dimension of the search space, $\overrightarrow{R}1$ and $\overrightarrow{R}2$ are vectors of dimension d, randomly generated from a uniform distribution in $[0.0, 1.0]$ and $\overrightarrow{p}_i$ is d dimension vector indicating the best position attained by particle i so far, whereas $\overrightarrow{g}$ indicates the best location vector found so far in the entire swarm. The term involving w is the *inertia component* and it controls the importance of the previous velocity in the new velocity. The term involving the constant $c1$ is called the *cognitive component* and it measures the degree of self-confidence of a particle, the degree at which it trusts its performance. The term involving $c2$ is the *social component* and it relies on the capacity of the swarm to discover better candidate solutions. Fig. 1a shows how particle i moves from position $\overrightarrow{x_i}$ a epoch t to a better position at epoch $t+1$, by taking into consideration its current velocity $\overrightarrow{v_i}$, its best position $pbest_i$ and the swarm overall best position *gbest*, and by doing so, it updates its velocity for epoch $t+1$.

In the BPSO version, formula (1) remains the same, except that now $\overrightarrow{p_i}$ and $\overrightarrow{x_i}$ are integers in $\{0, 1\}$ and $\overrightarrow{v_i}$, since it is a probability, must be constrained to the



(a) Moving principle of particles

(b) Particle's position decoding into selected features

Fig. 1: Principals of PSO movement and transformation

interval $[0.0, 1.0]$, a logistic transformation $S(\overrightarrow{v_i})$ can be used to accomplish this transformation. The change in position is defined as:

$$S(\overrightarrow{v_i}) = \frac{1}{(1 + e^{-\overrightarrow{v_i}})} \tag{3}$$

$$x_{i,j} = \begin{cases} 1, & \text{if } rand() < S(v_{i,j}). \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

where $S(\overrightarrow{v_i})$ is a sigmoid limiting the transformation, $rand()$ is a random number selected from a uniform distribution in $[0.0, 1.0]$, and $x_{i,j}$ is the j^{th} bit of the binary position of the i^{th} particle. When applying BPSO to feature selection, we consider that each particle position is represented by a vector whose elements are binary values. Every bit represents a feature, the value 1 means the corresponding feature is selected while 0 not selected. Each position is a feature subset.

4 The COMB-PSO algorithm

In this paper, we present our contribution to the PSO algorithm, a novel variant based on a combination of continuous and binary (COMB-PSO) model with a new *diversity preservation strategy* to make the swarm investigate unexplored areas of the search space and avoid premature convergence, a new *encoding mechanism* to allow the swarm to converge faster to very small feature subsets, and a *dynamic inertia weight* to control balance between exploration and exploitation.

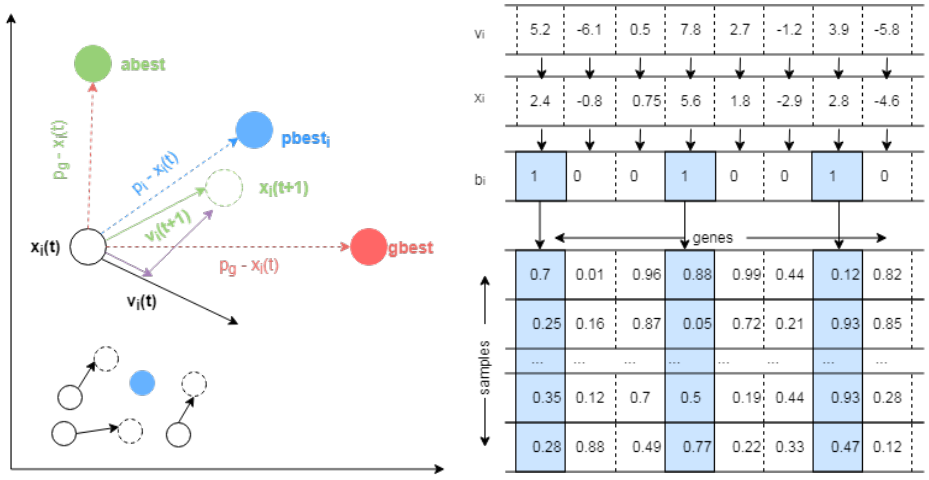
4.1 Diversity preservation strategy

PSO has emerged to be one of the most useful techniques to address high-dimensional optimization problems mainly as a result of its fast convergence property (as contrasted with other population based algorithms). However, its wide application was hampered by the premature stagnation of its candidate solutions. This issue is more articulated for the feature selection optimization problem which involves a mixture of continuous and binary variables. In this case, the velocity tends to go into $\bar{v}$ or $\underline{v}$ by the velocity update if the corresponding position is 1 or 0. In such circumstance, it is hard to redirect the corresponding position with a little change in velocity, which limits the search of the entire swarm. In order to investigate unexplored areas of the search space, we suggest to introduce a *shuffle and archive* mechanism, a diversity preservation strategy to avoid stagnation of the particles. The idea is to make the swarm try new global best positions with different subsets of features, while following the historical best position. We use a random mutation mechanism to modify the global best vector by shuffling its content, the criteria to shuffle the global best is to check for the stagnation of its fitness value for a given number of iterations, usually, around three iterations [3, 7]. In this work, we introduce a new variable *archive best a* to hold the historical best position. Then, to favor variability while preserving consistency and avoid divergence of the swarm, we expand the original velocity eq. (1) and add a to eq. (5), where $c3$ and $\overrightarrow{R3}$ have the same definition as $c1, c2$ and $\overrightarrow{R1}, \overrightarrow{R2}$. We call the term involving $c3$, the *diversity component* and it depends on the capability of the swarm to diversify its candidate solutions. Fig. 2a shows a cartoon of how particle i's position $\overrightarrow{x_i}$ has been impacted by the introduction of archive best. Once the maximum number of epochs reached, it is the archive best that is returned back as best position.

$$\begin{aligned} \overrightarrow{v_i} = w\overrightarrow{v_i} + c1\overrightarrow{R1}(\overrightarrow{p}_i - \overrightarrow{x_i}) \\ +c2\overrightarrow{R2}(\overrightarrow{g} - \overrightarrow{x_i}) + \underbrace{c3\overrightarrow{R3}(\overrightarrow{a} - \overrightarrow{x_i})}_{\text{diversity component}} \end{aligned} \tag{5}$$

4.2 Encoding

Before attempting to propose a new method, it would be careful to find the limitations of the standard BPSO by analyzing the main formulas (3) and (4). At each iteration, the logistic function in eq. (3) represents the probability for $\overrightarrow{x_i}$ to be 0 or 1. As analyzed in [2], using the standard sigmoid function in high-dimensional data only reduces the number of genes to about half the total number of genes. Another drawback of eq. (4) is that by moving particles in the hamming space, we loose precious information about the actual position. In this paper, we present a modification of the standard BPSO. The search in



(a) Particle's velocity and position updates in COMB-PSO

(b) Particle's position decoding in COMB-PSO

Fig. 2: PSO movement and transformation in COMB-PSO

continuous space is implemented as in conventional PSO, afterward, we introduce a new discrete variable $\overrightarrow{b_i}$ to the update the movement in Hamming space.

In this paper, we introduce a new encoding mechanism for the particles. In the COMB-PSO algorithm, the particle's velocity and position will still be updated using eq. (1) and eq. (2) respectively, nonetheless, we present a new binary position vector in eq. (6).

$$b_{i,j} = \begin{cases} 1, & \text{if } rand() < S(x_{i,j}). \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

This approach is expected to avoid undesirable discrepancy generated by using a mixture of continuous velocity and discrete position. We experiment with this method and we show that particles converge faster to very small feature subsets, while achieving better classification performance since the personal best and global best are likewise positioned in continuous space. Fig. 2b depicts the transformation of the particle's position from continuous domain to the binary domain and then to feature selection.

The second improvement is to avoid filtering out critical genes highly related to the sample classes. We introduce a *clipping* mechanism to limit the velocity and position vectors to $[\underline{v}, \bar{v}]$ and $[\underline{x}, \bar{x}]$ respectively. However, the choice of these boundaries is important, if $\underline{v}$ or $\bar{x}$ are too large, many irrelevant genes will be selected, inversely, if $\underline{v}$ or $\bar{x}$ are too small, some critical genes will be missed

in the selection process. We present the values selected for our experiments in subsection **??**.

4.3 Inertia weight

The inertia weight proposed in this paper, is neither set to a constant value nor set as linearly decreasing time-varying function. It has been proven in [**?**] that a dynamic inertia weight improves the performance of PSO over constant value. However, a time-varying linearly decreasing weight function implies that all particles have to adopt the same search pattern (exploration vs exploitation) at each iteration, which is usually not the case. Some particles may be in exploration mode while others in exploitation. In this paper, we introduce a novel inertia weight function based on the Jaccard index which represents the similarity coefficient between the particle and the global best. Less similarities with the global best induces a higher inertia weight to encourage exploration. On the other hand, when the particle gets closer to its local optimum, a lower inertia weight is derived to privilege exploitation. We used the Jaccard index instead of the Hamming distance (i.e. the number of bits that are different between the two binary vectors) because the former ignores the 0-0 matches, and therefore, is more appropriate for comparing two genes subsets; since we want to see how many genes the two subsets share. The formula in eq. (7) represents the inertia weight as a linear non monotonic function that fluctuates between predefined boundaries.

$$\begin{aligned} w_i = \overline{w} - (\overline{w} - \underline{w}) \quad J(\overrightarrow{g_i}, \overrightarrow{x_i}) \\ J(\overrightarrow{g_i}, \overrightarrow{x_i}) = \frac{M_{11}}{n - M_{00}} \end{aligned} \tag{7}$$

where w_i is the inertia weight of the i^{th} particle, $\underline{w}$, and $\overline{w}$ are the min, max boundaries of the inertia weight, $\overrightarrow{x_i}$ and $\overrightarrow{g_i}$ are the particle's position and its local best, J is the Jaccard index, where M_{11} is the number of 1-1 matches between $\overrightarrow{g_i}$ and $\overrightarrow{x_i}$, while M_{00} is the number of 0-0 matches and n is the particle size.

4.4 Objective function

Gene selection is a multi-objective optimization problem where we have to maximize the classification accuracy and minimize the size of the feature subset, two conflicting objectives. One of the most common general scalarization methods for multi-objective optimization is the *the weighted sum method* in which all objective functions are combined to form a single function as generally defined in eq. (8).

$$maximize \quad F(x) = \sum_{i=1}^{k} w_i f_i(x) \tag{8}$$

In this paper, the sensitivity performance and the number of features selected are balanced using a weight coefficient in the fitness function, which is expected to solve the problem of selecting a redundant feature subset. The formula for the fitness function is defined in eq. (9):

$$f(X, y, b, \alpha) = \alpha P_b + (1 - \alpha)\frac{|X| - |b|}{|X|} \tag{9}$$

where b is the subset of features selected, $|b|$ is the number of features (number of bits = 1) in b, X is the set of all the features in the dataset, $|X|$ is the number of features (dimensions) of X, y is the set of class label values, P_b is the classification performance using only the features decoded by b given class label y, α and $(1-\alpha)$ are weight factors that denote the importance of the number of features and the classification performance. As proposed in [2], $\alpha \in [0.6, 0.9]$, we set $\alpha = 0.8$ in this paper, to denote the importance of the classification performance compared to the number of features selected. This formula is practically penalizing the classification performance with the α weight and compensating that penalty with a reward in favor of small size feature subsets.

P_b is a performance measure where *accuracy*, defined as the ratio of the total number of good classifications over the total number of examples, has been commonly utilized. In this work, we experiment with a binary classification problem (patients and controls), and we apply both *accuracy* and *sensitivity* (ratio of true positive over all the positive class samples) measure. We found similar results with both performance measures which will be presented in section 5.

4.5 Architecture

COMB-PSO is used as a wrapper method, that is solely a feature selection method in which every candidate solution is evaluated using a learning machine model. The basic idea is to apply a classifier to each subset of the features selected, then, a 10-fold cross validation is applied and the score mean value is retained. In this work, we experiment with both SVM and Random Forest (RF) as a predictive model to evaluate the subset of features. The two models displayed almost equivalent classification measures, except that RF was almost 10-times faster. Fig.3 shows the architecture of the wrapper COMB-PSO feature selection approach.

As a preprocessing step we applied scaling to the dataset to prevent feature values in greater numeric ranges from dominating those in smaller numeric ranges, and to prevent numerical difficulties in the calculation. Each feature was linearly

scaled to $[-1.0, 1.0]$ interval. Scaling does not always guarantee improvement, but in this work, our experimental results demonstrate that scaling the feature value improves the classification performance.

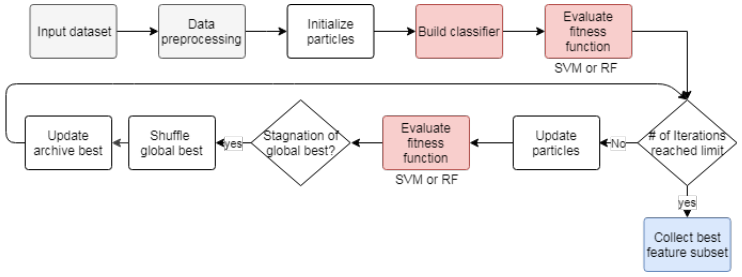


Fig. 3: The process diagram of COMB-PSO as a wrapper method

Algorithm 1 Novel Continuous-Binary PSO Algorithm

Require: $N, \bar{t}, \underline{w}, \bar{w}, \underline{v}, \bar{v}, \underline{x}, \bar{x}, c1, c2, c3$
Ensure: list of *gbest* subsets of selected features
1: **function** ICBPSO(X, y, f)
2: Initialize swarm velocities, positions, binary positions, costs, personal bests, global best and historical best
3: $t \leftarrow 0$.
4: **while** $t < \bar{t}$ **do** ▷ Update swarm block
5: randomly set $R \in [0.0, 1.0]$
6: **for** $i \leftarrow 0, N$ **do**
7: $\overrightarrow{w_i} \leftarrow \bar{w} - (\bar{w} - \underline{w})J(\overrightarrow{g} - \overrightarrow{x_i})$ using eq. (7)
8: randomly set $\overrightarrow{R1}, \overrightarrow{R2}, \overrightarrow{R3} \in [0.0, 1.0]$
9: update $\overrightarrow{v_i}$ using eq. (5)
10: clip $\overrightarrow{v_i} \in [\underline{v}, \bar{v}]$
11: $\overrightarrow{x_i} \leftarrow \overrightarrow{x_i} + \overrightarrow{v_i}$ using eq. (2)
12: clip $\overrightarrow{x_i} \in [\underline{x}, \bar{x}]$
13: $S \leftarrow 1/1 + e^{-\overrightarrow{x_i}}$ using eq. (3)
14: update $\overrightarrow{b_i}$ using eq. (4)
15: $cost_i \leftarrow f(X, y, \overrightarrow{b_i}, \alpha)$
16: update $\overrightarrow{p_i}$, $\overrightarrow{g}$ and $\overrightarrow{a}$
17: **end for**
18: **if** $\overrightarrow{g}$ stagnates for 3 iterations **then**
19: randomly shuffle position $\overrightarrow{g}$
20: update $\overrightarrow{g}$ and $\overrightarrow{a}$
21: **end if**
22: $t \leftarrow t + 1$
23: **end while**
24: **return** $\overrightarrow{a}$ ▷ return best feature subset
25: **end function**

5 Experiments

COMB-PSO involves a set of hyper parameters that regulate the inertia, the personal behavior, the social behavior, and the diversity preserving behavior of the swarm. Parameter selection, in any heuristic algorithm, is far from trivial. For instance, a minor change in the sigmoid function, may have a great effect on the rate of convergence of the whole algorithm. Nonetheless, parameters depend also on the dataset at hand. In any case, the parameters suggested in this paper are mostly empirical, and their detailed analysis is not within the scope of this work. Parameters are presented in Table 1. Since there has no guidance on how to select the population size and maximum iteration number in PSO [22], we determine the values of these parameters within the cross-validation runs on the validation dataset. We choose to set the population size to 100 and the number of iterations to 100. According to [23,24], a recommended choice for the acceleration

constants $c1$ and $c2$ is 2, and a better decrease for the inertia weight, w, is form 1.4 to 0.5. In this study, based on the conclusions in [24] and the cross-validation runs on the validation dataset, the initial and final inertia weight are set as 0.9 and 0.4, respectively on all data, and the acceleration constants $c1$ and $c2$ are both selected as 2.1 on all data, so that particles are attracted toward the averages of $\overrightarrow{p_i}$, $\overrightarrow{g}$ and $\overrightarrow{a}$.

Table 1: Parameter settings for COMB-PSO

Parameters	Values
Nbr particles	100
Nbr iterations	100
Nbr runs	30
Nbr running cycles	9
($\underline{w}$, $\bar{w}$)	(0.4, 0.9)
(C1, C2)	(2.1, 2.1)
($\underline{v}$, $\bar{v}$)	(-4.0, 0.25)
($\underline{x}$, $\bar{x}$)	(-6.0, 6.0)

To verify the effectiveness and efficiency of the proposed gene selection method and before applying our algorithm on the Alzheimer's blood dataset, we first experimented on public domain datasets downloaded from http://www.gems-system.org/ which include binary as well as multiclass gene expression datasets, as summarized in Table 2. All experimental results reported in this paper were implemented using Python 3.6 on a Linux platform. The minimum hardware requirement for executing the implementation of COMB-PSO is a processor with at least 2.0-GHz clock speeds and a minimum random access memory of 512 MB.

Table 2: Summary of public gene expression datasets

Dataset	Nbr samples	Nbr genes	Nbr of classes
Leukemia 2	72	11 225	3
Prostate Tumor	102	10 509	2
DLBCL	77	5469	2

Experimental results produced by COMB-PSO are compared to the ones generated by BPSO on the same datasets, applying the same hyper parameters (Except for the inertia weight) and using the same environment. Two criteria are

examined in evaluating the performance of the two algorithms. Small number of selected genes and high accuracy are required to achieve high performance. Accuracy in all experiments is collected after implementing a 10-fold cross-validation on *Random Forest* classifier. Several experiments are independently conducted thirty times on each dataset using COMB-PSO and BPSO. Next, an average result of the top ten independent runs is obtained. Based

Table 3: Comparison of performance results between COMB-PSO and BPSO on public datasets

	Leukemia 2				Prostate Tumor				DLBCL			
	COMB-PSO		BPSO		COMB-PSO		BPSO		COMB-PSO		BPS	
Run	Nbr genes	Acc.(%)	Nbr genes	Acc.(%)	Nbr genes	Acc.(%)	Nbr genes	Acc.(%)	Nbr genes	Acc.(%)	Nbr genes	A
1	25	94.82	198	95.65	18	90.09	169	92.09	9	91.25	91	
2	26	92.80	202	95.71	23	89.18	181	92.09	9	91.25	92	
3	28	93.21	208	96.07	25	92.18	193	93.18	10	96.25	93	
4	28	95.89	211	97.50	26	87.27	193	91.18	12	92.50	95	
5	31	94.23	212	96.90	27	89.18	196	93.09	13	93.75	95	
6	31	94.64	214	95.89	27	90.18	200	90.27	13	91.07	97	
7	32	95.89	217	97.32	27	89.18	204	93.09	14	92.32	97	
8	32	93.57	217	98.57	29	90	204	92.09	16	92.32	99	
9	33	95.83	221	96.07	29	90	205	93.09	16	91.90	100	
10	34	96.25	221	98.57	29	92.09	205	93.09	16	92.50	100	
Avg	30	94.71	212	96.83	26	89.94	195	92.33	13	92.51	96	
±S.D.	±2.90	±0.01	±7.27	±0.01 ±3.2	±1.4	±11.3	±0.01	±2.64	±0.01	±3.08	±0.01	

For Leukemia 2, we obtained a 84.88% accuracy for the whole dataset of 11 225 genes. Likewise, for Prostate tumor, we o and accuracy of 81.36% for the complete set of 10 509 genes, and for the DLBCL, we obtained an accuracy of 84.58% genes.

Next, we conducted experiments on clinic microarray datasets including gene expression collected from whole blood samples (2.5 ml) after 2 h of fasting into Paxgene Blood RNA tubes (BD). Illumina Human HT-12 Expression BeadChips were used to analyze the whole transcriptome according to the manufacturers protocol. The gene expression analysis was run in two batches at two different sites. The raw gene expression data are available as GEO datasets (Accession number GSE63060 for batch 1 and GSE63061 for batch 2). The datasets are summarized in Table 4.

Table 4: Blood gene expression datasets

Dataset	Nbr samples	AD	CTL	Nbr genes	Nbr of classes
Batch 1	217	104	113	5437	2
Batch 2	264	125	139	5307	2

On the Alzheimer's datasets, the outcome of the experiments shows COMB-PSO achieving superior performance over BPSO as presented in Table 5. For instance, for batch 1, the average subset size is 18 for COMB-PSO with S.D. of ± 3.03, while it is 123 in BPSO with S.D. of ± 14.94. Furthermore, the accuracy is 80.29% for COMB-PSO with S.D. of ± 0.48, while it is only 79.73% in BPSO with S.D. of ± 0.72.

Table 5: Comparison of performance results between COMB-PSO and BPSO

	Batch 1				Batch 2			
	COMB-PSO		BPSO		COMB-PSO		BPSO	
Run	Nbr genes	Accuracy(%)	Nbr genes	Accuracy(%)	Nbr genes	Accuracy(%)	Nbr genes	Accuracy(%)
1	13	80.81	122	81.35	15	73.14	108	72.70
2	18	80.79	143	80.36	14	73.11	142	72.10
3	24	80.79	127	80.11	15	72.86	117	72.00
4	18	80.58	125	79.96	19	72.71	129	71.69
5	17	80.35	125	79.90	19	72.12	97	71.23
6	14	80.34	110	79.57	22	71.72	101	70.84
7	20	80.34	113	79.07	14	71.36	122	70.58
8	19	79.94	153	79.03	24	71.29	100	70.52
9	21	79.49	109	79.03	16	71.28	94	70.51
10	19	79.43	102	78.99	22	71.23	116	70.47
Avg	18	80.29	123	79.73	18	72.08	113	71.26
$\pm$S.D.	± 3.03	± 0.48	± 14.94	± 0.72	± 3.52	± 0.76	± 14.69	± 0.008

For batch 1, we obtained a 70.19% accuracy for the whole dataset of 5435 genes. Likewise, for batch 2, we obtained and accuracy of 63.68% for the complete set of 5305 genes.

The biological outcome of this paper include the determination of Alzheimer's disease based on the selected informative genes. Table 6 shows the top selected genes after 30 independent runs of the algorithm. Results show consistency in the outcome of the top genes selected in batch 1 and batch 2. Furthermore, most of the top genes identified in this work are already identified to be potential clinical markers for AD by biologicial research, for instance, genes RPL36AL, RPS27A, MRPL51, CETN2, NDUFA1, RPS27A were selected by the methods proposed in [25] with highest boostrap count, and genes RPL36AL, MRPL51,

ING3, CETN2, NDUFA1, LOC653658, AK2, LOC646200, NDUFS5, RPS27A were likewise selected by the methods proposed in a more recent paper [26] as most relevant genes. Some of the remaining genes may be good candidates for further clinical and medical investigation. Biologists can save time since they can directly refer to the genes that are the best prospects for AD classification and drug discovery.

Table 6: Top selected genes in Blood dataset

	Batch 1	Batch 2	Found in [26]	Description
1	**NDUFA1**	**RPL36AL**	**RPL36AL**	Ribosomal protein L36a-like
2	**RPL36AL**	ATP5I	**RPL36AL**	
3	**ING3**	LOC401206	**MRPL51**	Mitochondrial ribosomal protein L51
4	**MRPL51**	**MRPL51**	**ING3**	Inhibitor of growth family, member 3
5	**LOC646200**	**NDUFS5**	**CETN2**	Centrin, EF-hand protein, 2
6	**AK2**	RPS25	**NDUFA1**	NADH dehudrogenase (ubiquinone) 1 alpha subcomplex
7	**RPS27A**	UQCRH	**LOC653658**	Ribosomal protein S23 pseudogene 8
8	**CETN2**	JAK1	**AK2**	Adenylate Kinase 2
9	C20orf11	LOC646508	**LOC646200**	
10	**LOC653658**	**NDUFA1**	**NDUFS5**	NADH dehydrogenase (ubiquinone) Fe-S protein 5
11	RPA3	PBX2	LOC654121	
12	CALML4	AATF	ZMAT2	Zinc finger, matrin-type 2
13	EXOSC9	SF3B14	**RPS27A**	Ribosomal protein S27a
14	CHMP4A	ACTB		
15	DNAJC7	SFRS5		
16	GNL2	C10orf26		
17	**NDUFS5**	SCARB2		
18	SNTB2	**CETN2**		
19	PPIH	RBM15		
20	ARFIP1	LOC654155		

6 Discussion and conclusion

In this work, we try to provide a better support for the feature selection problem when applied to high-dimensional gene expression datasets. We tweaked the well known Particle Swarm Optimization (PSO) algorithm. To better probability for selecting the most relevant genes, we introduced a new coding mechanism based on a hybrid continuous and binary particle position. Furthermore, we present a new diversity preservation strategy which guides the swarm away from being trapped into a pseudo-local optimum situation and push it towards other search opportunities. We provide an updated inertia component that helps preserve a good equilibrium between exploration and exploitation. On the performance

level, we compared our work to the standard BPSO and obtain superior results. On the genetic level, we checked the outcome, the set of the most dominant genes with equivalent biological research and observed similar comforting results.

References

1. E. Alba, J. Garcia-Nieto, L. Jourdan, and E.-G. Talbi, "Gene selection in cancer classification using pso/svm and ga/svm hybrid algorithms," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. IEEE, 2007, pp. 284–290.
2. M. S. Mohamad, S. Omatu, S. Deris, and M. Yoshioka, "A modified binary particle swarm optimization for selecting the small subset of informative genes from gene expression data," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 6, pp. 813–822, 2011.
3. L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, "Improved binary pso for feature selection using gene expression data," *Computational Biology and Chemistry*, vol. 32, no. 1, pp. 29–38, 2008.
4. W. A. Freije, F. E. Castro-Vargas, Z. Fang, S. Horvath, T. Cloughesy, L. M. Liau, P. S. Mischel, and S. F. Nelson, "Gene expression profiling of gliomas strongly predicts survival," *Cancer research*, vol. 64, no. 18, pp. 6503–6510, 2004.
5. F. Han, C. Yang, Y.-Q. Wu, J.-S. Zhu, Q.-H. Ling, Y.-Q. Song, and D.-S. Huang, "A gene selection method for microarray data based on binary pso encoding gene-to-class sensitivity information," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 14, no. 1, pp. 85–96, 2017.
6. C.-S. Yang, L.-Y. Chuang, C.-H. Ke, and C.-H. Yang, "A hybrid feature selection method for microarray classification." *IAENG International Journal of Computer Science*, vol. 35, no. 3, 2008.
7. S. M. Vieira, L. F. Mendonça, G. J. Farinha, and J. M. Sousa, "Modified binary pso for feature selection using svm applied to mortality prediction of septic patients," *Applied Soft Computing*, vol. 13, no. 8, pp. 3494–3504, 2013.
8. J. C. Ang, A. Mirzal, H. Haron, and H. N. A. Hamed, "Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 13, no. 5, pp. 971–989, 2016.
9. Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.
10. Y.-W. Chen and C.-J. Lin, *Combining SVMs with Various Feature Selection Strategies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 315–324.
11. J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," in *Feature extraction, construction and selection*. Springer, 1998, pp. 117–136.
12. W. A. A. Albukhanajer, "Multi-objective feature extraction and ensembles of classifiers for invariant image identification," Ph.D. dissertation, University of Surrey (United Kingdom), 2015.
13. U. Kamath, K. De Jong, and A. Shehu, "Effective automated feature construction and selection for classification of biological sequences," *PloS one*, vol. 9, no. 7, p. e99982, 2014.
14. J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

15. X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 459 – 471, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865506002327
16. B. Tran, B. Xue, and M. Zhang, "A new representation in pso for discretization-based feature selection," *IEEE Transactions on Cybernetics*, 2017.
17. F. Han, W. Sun, and Q.-H. Ling, "A novel strategy for gene selection of microarray data based on gene-to-class sensitivity information," *PloS one*, vol. 9, no. 5, p. e97530, 2014.
18. B. S. Mahanand, S. Suresh, N. Sundararajan, and M. A. Kumar, "Identification of brain regions responsible for alzheimer's disease using a self-adaptive resource allocation network," *Neural Networks*, vol. 32, pp. 313–322, 2012.
19. C.-H. Yang, L.-Y. Chuang, C. H. Yang *et al.*, "Ig-ga: a hybrid filter/wrapper method for feature selection of microarray data," *Journal of Medical and Biological Engineering*, vol. 30, no. 1, pp. 23–28, 2010.
20. S. Nakano, A. Ishigame, and K. Yasuda, "Particle swarm optimization based on the concept of tabu search," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. IEEE, 2007, pp. 3258–3263.
21. J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 5, Oct 1997, pp. 4104–4108 vol.5.
22. J. M. Keller, D. Liu, and D. B. Fogel, *Fundamentals of Computational Intelligence: Neural Networks, Fuzzy Systems, and Evolutionary Computation*. John Wiley & Sons, 2016.
23. Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. IEEE, 1998, pp. 69–73.
24. Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *International conference on evolutionary programming*. Springer, 1998, pp. 591–600.
25. K. Lunnon, M. Sattlecker, S. J. Furney, G. Coppola, A. Simmons, P. Proitsi, M. K. Lupton, A. Lourdusamy, C. Johnston, H. Soininen *et al.*, "A blood gene expression marker of early alzheimer's disease," *Journal Of Alzheimer's Disease*, vol. 33, no. 3, pp. 737–753, 2013.
26. N. Voyle, A. Keohane, S. Newhouse, K. Lunnon, C. Johnston, H. Soininen, I. Kloszewska, P. Mecocci, M. Tsolaki, B. Vellas *et al.*, "A pathway based classification method for analyzing gene expression for alzheimer's disease diagnosis," *Journal of Alzheimer's Disease*, vol. 49, no. 3, pp. 659–669, 2016.