2009 International Conference on Machine Learning and Computing
IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

An Improved Binary Particle Swarm Optimization with Complementary Distribution Strategy for Feature Selection

Li-Yeh Chuang[1], Chih-Jen Hsiao[2], Cheng-Hong Yang[2+]

[1]Department of Chemical Engineering, I-Shou University, Kaohsiung, Taiwan

[2]Department of Computer Science and Information Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan

Abstract. Feature selection is a preprocessing technique with great importance in the fields of data analysis, information retrieval processing, pattern classification, and data mining applications. It process constitutes a commonly encountered problem of global combinatorial optimization. This process reduces the number of features by removing irrelevant, noisy, and redundant data, thus resulting in acceptable classification accuracy. This paper presents a novel optimization algorithm called complementary binary particle swarm optimization (CBPSO), in which using complementary distribution strategy to improve the search capability of binary particle swarm optimization (BPSO) by facilitates global exploration and local exploitation via complementary particles and original particles, respectively. This complementary approach is introduction new particles ("complementary particles") into the search space, which generated by half of the whole particles selected randomly, and replaces the selected particles when the fitness of the global best particle has not improved for a number of consecutive iterations. In this study, the K-nearest neighbor (K-NN) method with leave-one-out cross- validation (LOOCV) was used to evaluate the quality of the solutions. CBPSO was applied and compared to ten classification problems taken from the literature. Experimental results show that CBPSO simplifies the feature selection process effectively, and either obtains higher classification accuracy or uses fewer features than BPSO and other feature selection methods.

Keywords: Feature Selection, Complementary Binary Particle Swarm Optimization, K-nearest neighbor, Leave-one-out Cross-Validation.

1. Introduction

Feature selection is a preprocessing technique for effective data analysis. The purpose of feature selection is the selection of optimal subsets, which are necessary and sufficient for solving the problem. Feature selection improves the predictive accuracy of algorithms by reducing the dimensionality, removing irrelevant features, and reducing the amount of data needed for the learning process. This can be done because not all available features are relevant for the classification process. Recently, feature selection has been successfully employed to effectively solve classification problem in various areas, such as pattern recognition [1], data mining [2, 3], and multimedia information retrieval [4, 5], and other areas where feature selection can be applied to. The filter methods and wrapper methods are two general approaches of feature selection. Filter methods define the relevant features without prior classification of the data [6]. Wrapper methods on the other hand incorporate classification algorithms to search for and select relevant features [7]. The wrapper methods generally outperform filter methods in terms of classification accuracy. This CBPSO introduced in this paper belongs to the wrapper method. A high number of selected features do not necessarily translate into high classification accuracy for many pattern classification problems. In some cases, the performance of algorithms devoted to speed and predictive accuracy of the data characterization can even decrease because features may be irrelevant or misleading, or due to spurious correlations. These factors can have a negative

+ Corresponding author. Tel.: + 886 7 381 4526 ext 5639; fax: +886 7 383 6844.
E-mail address: chyang@cc.kuas.edu.tw

impact on the classification process during the learning stage. Ideally, feature selection method reduces the cost of feature measurement, and increase classifier efficiency and classification accuracy.

The present study proposes a novel optimization algorithm for feature selection called complementary binary particle swarm optimization (CBPSO). In CBPSO, we employ the strategy of complementary distribution particles in the search space to improve the search efficiency of original BPSO. Complementary particles are introduced into the search space if the fitness of gbest is not improved (i.e. unchanged) a number of consecutive iteration. These complementary particles are introduced at complementary positions of the half of whole particles selected randomly and replace the selected particles. The complementary particles will initialize a new search from these complementary positions in search apace, and open up new opportunities for finding better solutions, they usually guide the entire swarm to promising new regions of the search space by their global exploration. In our study, the K-nearest neighbor method (K-NN) with leave-one-out cross-validation (LOOCV) based on Euclidean distance calculations was used to evaluate the quality of solutions for ten classification problems taken from the literature. Experimental results show that CBPSO simplifies the feature selection process effectively, and either obtains higher classification accuracy or uses fewer features than BPSO and other feature selection methods.

2. Method

2.1. Binary Particle Swarm Optimization (BPSO)

PSO is a population based optimization tool, which was originally introduced as an optimization technique for real-number spaces. In PSO, each particle is analogous to an individual “fish” in a school of fish. A swarm consists of N particles moving around a D-dimensional search space. The process of PSO is initialized with a population of random particles and the algorithm then searches for optimal solutions by continuously updating generations. Each particle makes use of its own memory and knowledge gained by the swarm as a whole to find the best solution. The position of the ith particle can be represented by xi = (xi1, xi2, …, xiD). The velocity for the ith particle can be written as vi = (vi1, vi2, …, viD). The positions and velocities of the particles are confined within [Xmin, Xmax]D and [Vmin, Vmax]D, respectively. The best previously visited position of the ith particle is denoted its individual best position pi = (pi1, pi2, …, piD), a value called pbesti. The best value of the all individual pbesti values is denoted the global best position g = (g1, g2, …, gD) and called gbest. At each generation, the position and velocity of the ith particle are updated by pbesti and gbest in the swarm. However, many optimization problems occur in a space featuring discrete, qualitative distinctions between variables and between levels of variables. For this reason, Kennedy and Eberhart introduced binary PSO (BPSO), which can be applied to discrete binary variables. In a binary space, a particle may move to near corners of a hypercube by flipping various numbers of bits; thus, the overall particle velocity may be described by the number of bits changed per iteration [8]. In BPSO, each particle is updated can be based on the following equations:

$$v_{id}^{new} = w \times v_{id}^{old} + c_1 \times r_1 \times \left(pbest_{id} - x_{id}^{old}\right) + c_2 \times r_2 \times \left(gbest_d - x_{id}^{old}\right) \quad (1)$$

$$\text{if } v_{id}^{new} \notin (V_{\min}, V_{\max}) \text{ then } v_{id}^{new} = \max(\min(V_{\max}, v_{id}^{new}), V_{\min}) \quad (2)$$

$$S\left(v_{id}^{new}\right) = \frac{1}{1 + e^{-v_{id}^{new}}} \quad (3)$$

$$\text{if } \left(r_3 < S\left(v_{id}^{new}\right)\right) \text{ then } x_{id}^{new} = 1 \text{ else } x_{id}^{new} = 0 \quad (4)$$

In these equations, w is the inertia weight that controls the impact of the previous velocity of a particle on its current one, r1 r2 and r3 are random numbers between (0, 1), and c1 and c2 are acceleration constants, which control how far a particle will move in a single generation. Velocities v_{id}^{new} and v_{id}^{old} denote the velocities of the new and old particle, respectively. x_{id}^{old} is the current particle position, and x_{id}^{new} is the new, updated particle position. In Eq. 2, particle velocities of each dimension are tried to a maximum velocity Vmax. If the sum of accelerations causes the velocity of that dimension to exceed Vmax, then the velocity of that dimension is limited to Vmax. Vmax and Vmin are user-specified parameters (in our case Vmax = 6, Vmin = -6). The position of particles after updating is calculated by the function $S(v_{id}^{new})$ (Eq. 3) [9]. If $S(v_{id}^{new})$ is larger than r3, then its position value is represented by {1} (meaning this position is selected for the next update). If $S(v_{id}^{new})$ is smaller than r3, then its position value is represented by {0} (meaning this position is not selected for the next update) [10].

2.2. Complementary Binary Particle Swarm Optimization (CBPSO)

The principal task of complementary strategy is assistance in the particles, which are trapped in a local optimal, on to a new region of the search space, and thus to facilitates global exploration. In CBPSO, a particle swarm is randomly initialized in a first step, and the particles are distributed over the D- dimensional search space. The position of each particle is represented in binary string form; the bit value {0} and {1} represent a non-selected and selected feature, respectively. Each particle is updated by following two values. The first one, pbest, is the best solution (fitness) it has achieved so far. The other value tracked by the BPSO is the global best value gbest obtained so far by all particles in the population. The position and velocity of each particle are updated by Eqs.1-4. However, according to Eqs.1-4, each particle will cluster around gbest and will only move a very small distance in the next generation if the distance between gbest and the surrounding particles is small. In order to avoid this premature convergence, complementary particles are introduced and replace the 50% of original particles with selected randomly from the swarm. These complementary particles are essential for the success of a given optimization task. The introduction of complementary particles in CBPSO is simple and can be done without increasing the computational complexity of the process. Complementary particles overcome the inherent defects (premature convergence) of BPSO by initializing a new global search over the entire search space from the complementary position of the 50% particles with selected randomly from the swarm.

2.3. K-Nearest Neighbor (K-NN)

The K-nearest neighbor (K-NN) method is one of the most popular nonparametric methods [11, 8] used for classification of new objects based on attributes and training samples. K-NN consists of a supervised learning algorithm which instantly classifies the result of a query instance based on the majority of the K-nearest neighbor category. The advantage of the K-NN method is that it is simple and easy to implement. K-NN is not negatively affected when training data are large, and indifferent to noisy training data [11]. In this study, the feature subset was measured by the leave-one-out cross-validation (LOOCV) of one nearest neighbor (1-NN), and calculated using their Euclidean distance. The pseudo-codes for CBPSO and 1-NN are given below.

CBPSO Pseudo-code

```
Begin
  Randomly initialize particles swarm
  while (number of iterations, or the stopping criterion is not met)
    Evaluate fitness of particle swarm by 1-NN( )
    for n = 1 to number of particles
      Find pbest_i and gbest
      for d = 1 to number of dimension of particle
        update the position of particles by Eq. 1-4
      next d
    next n
    if fitness of gbest is the same three times then
      for n = 1 to half number of particles
        for d = 1 to number of dimension of particle
          if the position of particles = 1 then
            position of particles = 0
          end if
          if the position of particles = 0 then
            position of particles = 1
          end if
        next d
      next n
    end if
  next generation until stopping criterion
end
```

1-NN pseudo-code

```
Begin
  for i = 1 to sample number of classification problem
    for j = 1 to sample number of classification problem
      for k = 1 to dimension number of classification problem
        dist_i = dist_i + (data_ik – data_jk)^2
      next k
      if dist_i < nearest then
        class_i = class_j
        nearest = dist_i
      end if
    next j
  next i
  for i = 1 to sample number of classification problem
    if class_i = real class of testing data then correct = correct + 1
    end if
  next i
  Fitness value = correct / number of testing data
end
```

3. Experimental results and Discussion

In order to investigate the effectiveness and performance of the CBPSO algorithm for classification problems, we chose six classification problems from the literature in our experiment. These data sets were obtained from the UCI Repository [9]. The data format is shown in Table 1. Three types of classification problems, small-, medium-, and large-sized groups, were tested. If the number of features is between 10 and 19, the sample group can be considered small; the Wine (WI), Vehicle (VE), and Segmentation (SE) problems constitute such small sample groups. If the number of features is between 20 and 49, the sample group test problems are of medium size. Groups in this category included the WDBC and Ionosphere (IO) problems. If the number of features is greater than 50, the test problems are large sample group problems; this group included the Sonar (SO) problem. The 1-NN method with leave- one-out cross-validation

(LOOCV) was used to evaluate all data sets. Employing the 1-NN method with LOOCV as a fitness function for CBPSO has two distinct advantages; first, the calculation time can be decreased, and secondly, higher classification accuracy can be obtained. In our experiment, the same conditions were used to compare the performance of GA, HGAs, BPSO and CBPSO algorithms, i.e. population size = 20, and the K = 1 in part of K-Nearest Neighbor. The other parameters of BPSO and CBPSO are set as follows: generations = 100 (maximum number of iterations), w = 1.0, and c1 = c2 = 2 [13]. The number of generations for the SGA, HGAs, BPSO and CBPSO is listed in Table 2. In Table 3, the classification accuracy and selected number of features is shown for the data sets tested with CBPSO and compared to BPSO and other feature selection methods from the literature [14]. In this paper, each data set was executed for 5 runs by the proposed method and this number is the same as Oh et al. used [14]. From Table 3, it can be seen that the classification accuracies of the Wine, Segmentation and WDBC classification problems obtained with CBPSO are 99.44%, 97.92% and 98.24%, respectively. This constitutes an increase of 4% classification accuracy compared to the literature methods shown in Table 3. Fig. 1-a to 1-f plot the number of generations versus the accuracy of BPSO and CBPSO for the six data sets, After several generations, the classification accuracy curve of BPSO usually remains unchanged, this characteristic indicating that BPSO is stuck in a local optimal. However, the classification accuracy achieved by BPSO can further be improved when complementary distribution strategy particles are introduced (CBPSO). Furthermore, from Fig. 1, we can observe the curve of CBPSO is displayed as step shape, and this phenomenon representative the CBPSO has capability to make a break-through in a local optimal. In Oh et al. [14], the classification accuracies were measured by the four values

Table 1. The number of generations setting

Data sets	Number of samples	Number of classes	Number of features	Evaluation method
WI	846	4	18	1-NN
VE	2310	7	19	1-NN
SE	569	2	30	1-NN
WDBC	351	2	34	1-NN
IO	6435	6	36	1-NN
SO	208	2	60	1-NN

Table 2. Format of classification test problems

Data sets	SGA	HGA				BPSO	CBPSO
		(1)	(2)	(3)	(4)		
WI	10000	600	200	120	80	100	100
VE	10000	600	200	120	80	100	100
SE	10000	600	200	120	80	100	100
WDBC	20000	600	200	120	80	100	100
IO	25000	600	200	120	80	100	100
SO	40000	600	200	120	80	100	100

Legend: SGA: simple genetic algorithm, HGA: hybrid genetic algorithm (Oh *et al.* [14]), BPSO: binary particle swarm optimization, CBPSO: the proposed method

Table 3. Classification accuracy of features for the tested data sets

Data sets	SFS	PTA	SFFS	SGA	HGA				BPSO	CBPSO
					(1)	(2)	(3)	(4)		
WI	95.51	95.51	95.51	95.51	95.51	95.51	95.51	95.51	98.31	**99.44**
VE	69.50	71.75	73.52	72.97	73.52	73.52	73.52	73.52	**75.06**	**75.06**
SE	92.95	92.95	92.95	92.95	92.95	92.95	92.95	92.95	**97.92**	**97.92**
WDBC	94.02	94.20	94.20	93.95	94.06	94.38	94.13	94.27	97.54	**98.24**
IO	93.45	93.45	93.45	94.70	95.38	**95.56**	**95.56**	95.50	94.02	94.87
SO	91.82	92.31	93.75	95.49	95.86	95.96	96.34	96.15	95.67	**97.60**
Average	89.54	90.03	90.56	90.93	91.21	91.31	91.34	91.32	93.09	**93.86**

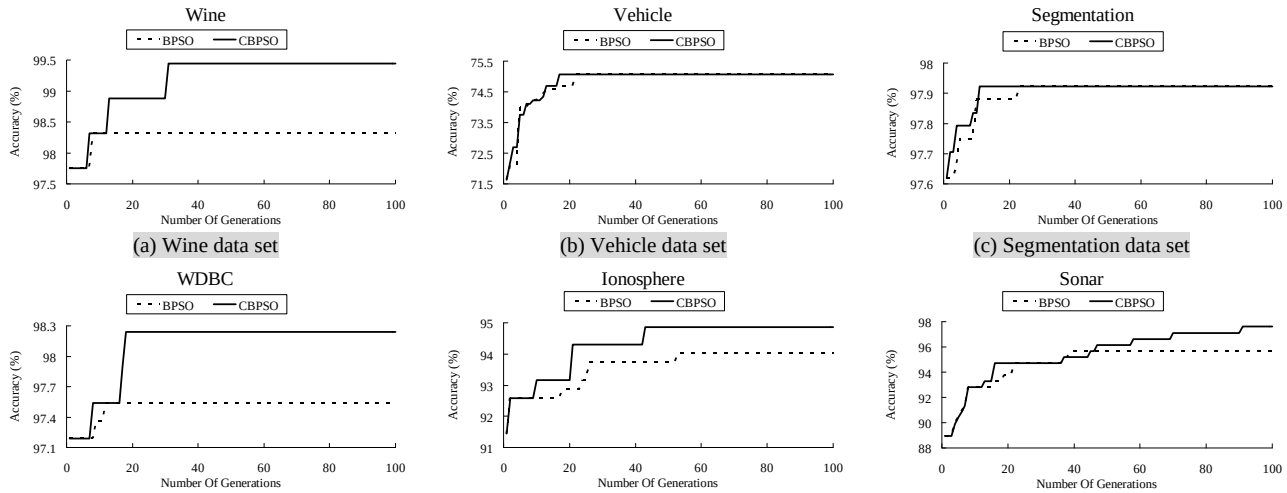
Legend: SFS: sequential forward search, PTA: plus and take away, SFFS: sequential forward floating search, SGA: simple genetic algorithm, HGA: hybrid genetic algorithm (Oh *et al.* [14]), BPSO: binary particle swarm optimization, CBPSO: the proposed method. Highest values are in bold-type. For GAs, BPSO and CBPSO.

D/5, 2D/5, 3D/5, and 4D/5 of the total number of features D. However, the optimal number of features for each test problem is unknown. From the experimental result, it can be concluded that the classification accuracy can not breakthrough if the number of features was constraints during the selection procedure.

4. Conclusions

CBPSO is introduced in which the complementary distribution strategy is applied to perform feature selection and improve the performance of BPSO. The K-NN method with LOOCV served as an evaluator of the CBPSO fitness functions. In CBPSO, complementary particles will initialize a new search from the complementary position of the 50% particles with selected randomly from the swarm when the gbest value stays unchanged after three numbers of iterations. Better solutions can be found via their global exploration, and guiding the entire particle swarm to more promising regions in the search space. The experimental results show that CBPSO simplifies feature selection by reducing the total number of necessarily selected features effectively, and that it obtained the highest in five of six data set test problems compared to BPSO and other feature selection methods from the literature. CBPSO can be used as a preprocessing tool to optimize the feature selection process. From computational complexity and classification accuracy, it can be

concluded that the proposed CBPSO method demonstrate higher efficiency than the BPSO and other methods. CBPSO could conceivably be applied to problems in other areas in the future.



(a) Wine data set (b) Vehicle data set (c) Segmentation data set

(d) WDBC data set (e) Ionosphere data set (f) Sonar data set

Fig. 1 The number of generations versus the accuracy of BPSO and CBPSO for the six data sets

5. References

[1] S. Gunal, and R. Edizkan. Subspace based feature selection for pattern recognition. *Information Sciences*. 2008, 178 (19): 3716-3726.

[2] M.J. Martin-Bautista, and M.-A. Vila. A Survey of Genetic Feature Selection in Mining Issues. Proc. of Congress on Evolutionary Computation. 1999, 2: 1314-1321.

[3] S. Piramuthu. Evaluating Feature Selection Methods for Learning in Data Mining Applications. *Proc. IEEE 31st Ann. Hawaii Int'l Conf. System Science.* 1998, 5: 294-301.

[4] M.S. Lew (Ed.). Principles of Visual Information Retrieval. *Springer press,* 2001.

[5] Y. Liu, and F. Dellaert. A Classification Based Similarity Metric for 3D Image Retrieval. Proc. I*EEE Int'l Conf. Computer Vision and Pattern Recognition.* 1998, pp. 800-805.

[6] H. Liu, and R. Setiono. A probabilistic approach to feature selection-A filter solution. *Proc. 13th Int'l Conf. Mach. Learn.* 1996, pp. 319-327.

[7] R. Kohavi, and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence.* 1997, 97 (1-2): 273-324.

[8] E. Fix, and J.L. Hodges, Discriminatory Analysis-Nonparametric Discrimination: Consistency Properties. Project 21-49-004, Report 4, US Air Force School of Aviation Medicine, *Randolph Field.* 1951, pp. 261-279.

[9] K. Crammer, and Y. Singer. On the learn ability and design of output codes for multiclass problems. *Machine Learning.* 2002, 47 (2-3): 201-233.

[10] J. Kennedy, and R.C. Eberhart. A discrete binary version of the particle swarm algorithm. Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, *IEEE International Conference,* 1997, 5: 4104-4108.

[11] T. Cover, and P. Hart. Nearest neighbor pattern classification. *Proc. IEEE Trans. Information Theory.* 1967, 13 (1): 21-27.

[12] P.M. Murphy, and D.W. Aha. UCI Repository of Machine Learning Databases. Technical report, Department of Information and Computer Science, *University of California, Irvine, Calif.*, 1994. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html.

[13] J. Kennedy, R.C. Eberhart, and Y. Shi. Swarm Intelligence. Morgan Kaufmann Publishers, *San Francisco,* 2001.

[14] I.-S. Oh, J.-S. Lee, and B.-R. Moon. Hybrid Genetic Algorithms for Feature Selection. *IEEE Trans. Pattern Analysis and Machine Intelligence,* 2004, 26 (11): 1424-1437.