Hamming Distance based Binary PSO for Feature Selection and Classification from high dimensional Gene Expression Data

Haider Banka and Suresh Dara [*]

Department of Computer Science and Engineering
Indian School of Mines, Dhanbad 826004, India.
banka.h.cse@ismdhanbad.ac.in, darasuresh@live.in

Abstract. In this article, a Binary Particle Swarm Optimization (BPSO) algorithm is proposed incorporating hamming distance as a distance measure between particles for feature selection problem from high dimensional microarray gene expression data. Hamming distance is used as an similarity measurement for updating the velocities of each particles or solutions. It also helps to reduce extra parameter (i.e. V_{min}) as needed in conventional BPSO during velocity updation. An initial fast pre-processing heuristic method is used for crude domain reduction from high dimension. Then the fitness function is suitably designed in multi objective framework for further reduction and soft tuning on the reduced features using BPSO. The performance of the proposed method is tested on three benchmark cancerous datasets (i.e., colon, lymphoma and leukemia cancer). The comparative study is also performed on the existing literature to show the effectiveness of the proposed method.

Keywords: Feature selection, Hamming distance, Microarray data, Evolutionary Computation, Optimization, Classification

1 Introduction

The microarray experiments produce gene expression patterns that provides dynamic information about cell function. In a single experiment, the DNA microarray technologies can simultaneous monitor and analysis of thousands of different genes in histological or cytological specimens, which is helps to find classifying diseased genes according in different levels of tissues [1]. Gene expression profiles usually contains a large number of genes but a few number of samples available . An important need to analyze and interpret the huge amount of data, involving the decoding of around $24000-30000$ human genes [2]. High dimensional feature selection (FS) techniques can help us to identify momentous features by applying certain selection criteria, and an effective feature selection reduces computation cost and increase the classification accuracy.

[*] This work was partially supported by Council of Scientific and Industrial Research (CSIR), New Delhi, India, under the Grant No. 22(0586)/12/EMR-II.

Over past few decades, the evolutionary computational techniques becomes very popular to solve many optimization problems. Nature inspired metaphors like Evolutionary algorithms (e.g, Genetic Algorithms (GA) [3], Differential Evolution (DE) [4]), and Swarm Optimization (e.g., Particle Swarm Optimization (PSO) [5], Ant Colony Optimization (ACO) [6], etc.) are playing important role in analysing and solving diverse types of problems.

The Hamming distance is the proportion of positions at which two binary vector sequences differ. Computing the Hamming distance between two vector requires two steps: (i) compute the XOR and (ii) count the number of 1's in the resulting vector. Let $S = \{0, 1\}$. The d-dimensional search space S^d consists of bit strings of length d. Each point $x \in S^d$ is a string $x = (x_0, x_1, \ldots, x_{d-1})$ of zero's and one's. Given two points $x, y \in S^d$, the Hamming distance $HD(x, y)$ between them is the number of positions at which the corresponding strings differ, i.e., $HD(x, y) = |\{i : x_i \neq y_i\}|$. It have been using in many application domains like DNA analysis [7], alignment,DNA barcode identification and classification of species [8], where DNA bar-codes provides full length barcode sequences for bar-code analytic.

The feature selection is a difficult problem in extracting and analyzing information form large database is associated with high complexity. Recently different approaches in this direction includes based on genetic algorithms (GAs) [9], genetic programming (GP)[10], and fuzzy based feature selection [11], binary particle swarm optimization [12], PSO based feature selection using multi objective functions [13], and PSO based support vector machine(SVM) [14], and BPSO based catfish effect feature selection [15].

In this article, we propose a swarm intelligent computational technique based Binary PSO algorithm for feature selection. In comparison with other heuristic techniques, PSO has less parameters, fast convergence, less computational burden, and high accuracy. We focus on feature selection from microarray gene expression profiles using BPSO with hamming distance. During velocity updating process we incorporate the hamming distance, which is used as dissimilarity measures between two solutions. At the first stage, the microarray data is preprocessed to remove irrelevant, redundant features removed and discretized to binary d-Distinction table, reduce the dimensionality of gene samples are decrease the computational burden. In the second stage, BPSO-HD is used to select the significant feature subsets. Experimental validation of selected feature subsets is done in terms of classification accuracy (%) using with k-NN classifier.

The rest of this paper is organized as follows. Section 2 describes the preliminaries of BPSO. The proposed BPSO-HD algorithm for feature selection described in Section 3. The experimental results with comparison study are reported in Section 4 followed by conclusion in Section 5.

2 Preliminaries of Binary PSO

This section formally explain the basics of binary particle swarm optimization as continuation and understanding of the present work. Particle Swarm Optimization (PSO) is a intelligence based multi-agent heuristic optimization technique [16]. The PSO start with the random initialisation of population(swarm) of individuals(particles) in the n-dimensional search space. Each particle keeps two values (*personal best* and *global best*) in its memory, its own best experience, one is best fitness values whose position P_i and it value P_{best}, and second one is best experience of whole swarm which position denotes P_g and its value $gbest$. The i^{th} particle denote as $X_i = (X_{i1}, X_{i2}, X_{i3}, \ldots, X_{id}, \ldots, X_{in})$, also denotes the velocity $V_i = (V_{i1}, V_{i2}, \ldots, V_{id}, \ldots, V_{in})$. The position and velocity of each particles update in each iteration according to the equ.(1) and equ.(2)

$$\begin{aligned} V_{id}(t+1) = wV_{id}(t) + c_1\rho_1(P_{id}(t) - X_{id}(t)) \\ + c_2\rho_2(P_g(t) - X_{id}(t)) \end{aligned} \tag{1}$$

$$X_{id}(t+1) = X_{id} + V_{id}(t+1) \tag{2}$$

where $d = 1, 2, 3, \ldots, n$ (i.e., dimension of each particle), w is inertia weight, it provides a balance between global and local exploration, and results in fewer iterations on average to find a sufficiently optimal solution, c_1 and c_2 are the same positive constants called the *cognitive* and *social acceleration* coefficient, represent the weighting of the stochastic acceleration terms that pull each particle toward *pbest* and *gbest* positions. ρ_1 and ρ_2 are two random numbers in the range of [0, 1].

PSO has been called as BPSO where each particle contains a combination of 0's and 1's [17]. Here, velocity V_{id} mapped into interval [0,1] via sigmoid function as $S(V_{id}) = \frac{1}{1+e^{-V_{id}}}$ and the velocity updates based on equ.(1), and position updates based on equ.(3), where ρ is random number in [0,1].

$$X_{id} = \begin{cases} 1, & \text{if } \rho < S(V_{id}) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The above update process implemented for all dimensions, and for all particles.

3 The Proposed Approach

In this section, we discuss the preprocessing of gene expression data, d-Distinction table, fitness function, and finally our proposed algorithm.

3.1 Preprocessing of Gene expression data

Gene expression data typically consists of huge number of features, but small number of samples available. The majority of features were not relevant to the description of the problem, could potentially degrade the classification performance by masking the contribution of the relevant features. Preprocessing directs to eliminating of ambiguously expressed genes as well as the constantly expressed genes across the tissue classes. Attribute-wise normalization is done by

$$a'_j(x_i) = \frac{a_j(x_i) - min_j}{max_j - min_j}, \forall i \tag{4}$$

where max_j and min_j correspond to the maximum and minimum gene expression values for attribute a_j over all samples. This constitutes the normalized gene data set, i.e., (continuous) attribute value table wether the range (0,1). Then we choose thresholds Th_i and Th_f, based on the idea of quartiles [9]. Let the N patterns be sorted in the ascending order of their values along the jth axis. In order to determine the partitions, we divide the measurements into a number of small class intervals of equal width δ and count the corresponding class frequencies fr_c. The position of the kth partition value ($k = 1, 2, 3$ for four partitions) is calculated as $Th_k = l_c + \frac{R_k - cfr_{c-1}}{fr_c} * \delta$, where l_c is the lower limit of the cth class interval, $R_k = \frac{N*k}{4}$ is the rank of the kth partition value, and cfr_{c-1} is the cumulative frequency of the immediately preceding class interval, such that $cfr_{c-1} \leq R_k \leq cfr_c$. Here we use $Th_i = Th_1$ and $Th_f = Th_3$. As a result, Th_1 is statistically chosen (for four partition i.e., $k = 3$) such that 1/3 of the sample values lies below Th_1. Similarly, Th_3 is statistically chosen in such a way so that 2/3 of the sample values lies below Th_3 under that particular feature. Remove from the table those attributes for which the number of '$*$'s are $\geq Th_a$. This is the *modified* (reduced) attribute value table $\mathcal{F}_r$. Features normalization using equ.(4), convert features values to binary (0 or 1) and if $a' \leq Th_i$ then put '0', if $a' \geq Th_f$ then put '1' otherwise put '*' don't care. Find the average of '*' occurrences over the feature table. Choose this as threshold Th_d, Remove from the table those attributes for which the number of '$*$'s are $\geq Th_d$. Prepare d-Distinction table based on equ.(5)

d-Distinction Table: The distinction table is a binary matrix where rows are labeled as the objects pairs and columns are features in F. An entry $b((k, j), i)$ of the matrix corresponds to the attribute a_i and pair of objects (x_k, x_j).

$$b((k,j),i) = \begin{cases} 1, & \text{if } a_i(x_k) \neq a_i(x_j). \\ 0, & \text{if } a_i(x_k) = a_i(x_j). \end{cases} \tag{5}$$

The presence of a '1' signifies the ability of the attribute a_i to discern (or distinguish) between the pair of objects (x_k, x_j).

For a decision table $\mathcal{F}$ with N condition attributes and a single decision attribute d, the problem of finding a d-reduct is equivalent to finding a minimal subset of columns $R(\subseteq \{1, 2, \cdots, N\})$ in the distinction table equ.(5), satisfying $\forall (k, j) \exists i \in R : b((k, j), i) = 1,$ whenever $d(x_k) \neq d(x_j)$. So, in effect, we may consider the distinction table to consist of N columns, and rows corresponding to only those object pairs (x_k, x_j) such that $d(x_k) \neq d(x_j)$. Let us call this shortened distinction table, *d-distinction table*. Note that, as $\mathcal{F}$ is taken to be consistent, there is no row with all 0 entries in a d-distinction table.

As object pairs corresponding to the same class do not constitute a row of the d-distinction table, there is a considerable reduction in its size thereby leading to a decrease in computational cost. Additionally, If either of the objects in a pair, has '$*$' as an entry under an attribute in table $\mathcal{F}_r$. Then in the distinction table, put '0' at the entry for that attribute and pair. The entries '1' in the matrix correspond to the attributes of interest for arriving at a classification decision.

3.2 Fitness function

The feature selection can be done by BPSO-HD Algorithm(1) using the following objective function. We proposed a fitness function, which includes two sub functions (F_1, F_2). Where F_1 finds number of features(i.e number of 1's), F_2 decides the extent to which the feature can recognise among the objects pairs. The proposed fitness functions is as follow:

$$Fit = \alpha_1 F_1(\boldsymbol{v}) + \alpha_2 F_2(\boldsymbol{v}) \tag{6}$$

where the two sub functions $F_1(\boldsymbol{v}) = \frac{N - O_{\boldsymbol{v}}}{N}$, and $F_2(\boldsymbol{v}) = \frac{R_{\boldsymbol{v}}}{C_1 * C_2}$ under the condition $0 < \alpha_1, \alpha_2 < 1$ such that $\alpha_1 + \alpha_2 = 1$. Here, $\boldsymbol{v}$ is the chosen feature subsets, $O_{\boldsymbol{v}}$ represents the number of 1's in $\boldsymbol{v}$, C_1 and C_2 are the number of objects in the two classes, and $R_{\boldsymbol{v}}$ is the number of object pairs (i.e., rows in the d-Distinction table) $\boldsymbol{v}$ can discern between. The fitness function F_1 gives the candidate credit for containing less number of features or attributes in $\boldsymbol{v}$, and F_2 determines the extent to which the candidates can discern among objects pairs in the d-Distinction table.

3.3 BPSO using Hamming Distance(BPSO-HD) Algorithm

The Algorithm (1) is proposed for feature selection, the chromosomes of population represents as particles in swarm. BPSO-HD algorithm starts with initialized random populations. We update the velocity using equ.(7). In each iteration, find the *Pbest* and *gbest* and update. For each dimension of particle, update the positions based on the its corresponding velocities using equ.(3) and equ.(7) with respectively.

In the Algorithm (1), we are using equ.(7) to update the velocity of each particle position. Equation (1) may generate negative values. However, using equ.(7), it doesn't generate any negative value, hence no use of V_{min} to set velocity boundary. Therefore, half of the comparisons are reduced, hence it leads to reduce the computational burden. Here, X and P are binary stings, the difference after the operation between X and P is also binary. Hence the equation (7) is more logical and justified.

$$\begin{aligned} V_{id}(t+1) = {} & wV_{id}(t) + c_1\rho_1(P_{id}(t) \oplus X_{id}(t)) \\ & + c_2\rho_2(P_{gd}(t) \oplus X_{id}(t)) \end{aligned} \tag{7}$$

4 Results and Comparisons

We have implemented the BPSO-HD Algorithm to find minimal feature subsets on high dimensional microarray data. We taken three different cancer datasets; colon[1], lymphoma[2], and leukemia[3]. We used $c_1, c_2 = 2$ and $w = 0.4 - 0.9$ as

[1] http://microarray.princeton.edu/oncology

[2] http://llmpp.nih.gov/lymphoma/data/figure1/figure1.cdt

[3] http://www.genome.wi.mit.edu/MPR

Algorithm 1 : The BPSO-HD Algorithm for Feature Selection

```
    Input: $c_1, c_2, w, V_{max}$, d-Distinction table
    Output: Feature subsets
Step: 1 Initialize population randomly    ▷ population of particles with binary random
        positions and velocities
Step: 2 Run maximum iterations
Step: 3 Evaluate fitness value of particle using equ.(6) for each Particle in swarm
        (a) if (the fitness value of $X_i > P_i$) then $P_i = X_i$
        (b) if (the fitness value of $X_i > P_g$) then $P_g = X_i$
Step: 4 for each dimension of particle
        (a) update each dimension velocity using equ.(7)
        (b) update velocity: if $(V_{id}(t+1) > V_{max})$ then $V_{id}(t+1) = V_{max}$
        (c) update position: if $(S(V_{id}(t+1)) > rand(0,1))$ then $X_{id}(t+1) = 1$
        (d) else $X_{id}(t+1) = 0$
Step: 5 Stop if maximum no.of iterations reached, else goto Step:2
```

Table 1. Details of the two-class microarray data

Datasets	Total Features	Reduced Features#	Classes	Samples		
				Total	Train	Test
Colon	2000	1102	Colon cancer	40	20	20
			Normal	22	11	11
Lymphoma	4026	1867	Other type	54	27	27
			B-cell	42	21	21
Leukemia	7129	3783	ALL	47	27	20
			AML	25	11	14

# After Preprocessing

parameter values based on literature [18]. We are interested on two-class problem (i.e normal and diseased samples), as summarized in Table 1.

The Table 2 represents $k-NN$ classification results with single objective function of GA. Here, it is giving 100% correct classification score for all three data sets when $k = 1$. For colon data 93.55% score when $k = 3$, 90.33% for $k = 5$ and for $k = 7$ it is 83.88%, on 10 feature subset. For lymphoma data, it is 93.75%, 93.75% and 89.59% where $k = 3, k = 5$ and $k = 7$ respectively. Similarly, for leukemia data, where k =3,5 and 7 the correct classification is 94.74%.

The Table 3 depicts comparative performance between proposed algorithm, GA with single objective function, and multi objective genetic algorithm (NSGA-II) [9] using with k-NN classifier. For colon and leukemia datasets, our proposed algorithm giving better results at all k values. For lymphoma dataset, it is 100% correct classification score where $k = 1$, remaining k values results all are also near to NSGA-II results.

Table 2. Comparative Performance on three datasets using k-NN Classifier

Dataset	Popu-lation size	Selected feature subset	k-nearest neighbors classification (%) on test set							
			$k = 1$		$k = 3$		$k = 5$		$k = 7$	
			Cr	ICr	Cr	ICr	Cr	ICr	Cr	ICr
Colon	10	10	100	0	93.55	6.45	90.33	9.67	83.88	16.12
	20	8	100	0	93.55	6.45	77.42	22.58	74.20	25.80
	30	7	100	0	83.88	16.12	74.20	25.80	67.75	32.25
	50	6	100	0	80.65	19.35	80.65	19.35	74.20	25.80
Lymphoma	10	15	100	0	93.75	6.25	89.59	10.41	89.59	10.41
	20	17	100	0	93.75	6.25	91.67	8.33	87.50	12.50
	30	14	100	0	93.75	6.25	89.59	10.41	87.50	12.50
	50	14	100	0	93.75	6.25	93.75	6.25	87.50	12.5
Leukemia	10	13	100	0	89.48	10.52	89.48	10.52	89.48	10.52
	20	10	100	0	94.74	5.26	94.74	5.26	94.74	5.26
	30	11	100	0	92.11	7.89	84.22	15.78	81.58	18.52
	50	11	100	0	92.11	7.89	86.85	13.15	81.58	18.42

Cr and **ICr** Average Correct and Incorrect classification score of two classes

Table 3. Comparative Performance Between BPSO-HD, NSGA-II and GA on three datasets using k-NN Classifier

Dataset	feature subset size	Used Method	k-nearest neighbors classification (%) on test set							
			$k = 1$		$k = 3$		$k = 5$		$k = 7$	
			Cr	ICr	Cr	ICr	Cr	ICr	Cr	ICr
Colon	≤ 10	Proposed	**100**	**0**	**93.55**	**6.45**	**90.33**	**9.67**	**83.88**	**16.12**
	≤ 10	NSGA-II [9]	90.3	9.07	90.3	9.07	87.1	12.9	80.6	19.4
	≤ 15	GA [9]	71.0	29.00	58.10	41.90	48.40	51.60	61.30	38.70
Lymphoma	≤ 20	Proposed	**100**	**0**	93.75	6.25	93.75	6.25	87.50	12.5
	≤ 2	NSGA-II	93.8	16.2	**95.8**	**4.2**	**95.8**	**4.2**	**95.8**	**4.2**
	≤ 18	GA	89.59	10.41	89.59	10.41	93.76	6.24	93.76	6.24
Leukemia	≤ 10	Proposed	**100**	**0**	**94.74**	**5.26**	**94.74**	**5.26**	**94.74**	**5.26**
	≤ 5	NSGA-II	94.1	5.9	91.2	8.8	91.2	8.8	88.2	11.8
	≤ 19	GA	73.50	26.50	73.53	26.47	60.77	38.23	67.65	32.35

5 Conclusion

In this article, we proposed BPSO with hamming distance to find future selection in gene expression microarray data. Hamming distance plays an important role in velocity updating of solutions. Our preprocessing aids faster convergence along the search space and successfully employed to eliminate redundant, and irrelevant features. We investigate the effectiveness of hamming distance with BPSO with combination of different parameters and population sizes.

The main goal of the feature selection is selecting minimal number of features and get higher classification accuracy. Here we have achieved the goal through the implementation of two sub fitness functions. The performance of the proposed method and existed methods are compared using k-NN classifier and reported better classification accuracy. For three cancer data sets results reported in this paper demonstrating the feasibility and effectiveness of the proposed feature selection method.

References

1. Stekel, D.: Microarray Bioinformatics. Oxford university and bius (2003)
2. : Special Issue on Bioinformatics. Volume 35. IEEE Computer Society (July 2002)
3. AlSukker, A., Khushaba, R., Al-Ani, A.: Enhancing the diversity of genetic algorithm for improved feature selection. In: IEEE International Conference on Systems Man and Cybernetics. (2010) 1325–1331
4. He, X., Zhang, Q., Sun, N., Dong, Y.: Feature selection with discrete binary differential evolution. In: International Conference on Artificial Intelligence and Computational Intelligence. Volume 4. (2009) 327–330
5. Xue, B., Zhang, M., Browne, W.: Particle swarm optimization for feature selection in classification: A multi-objective approach. IEEE Transactions on Cybernetics (99) (2013) 1–16
6. Xiong, W., Wang, C.: A hybrid improved ant colony optimization and random forests feature selection method for microarray data. Networked Computing and Advanced Information Management, International Conference on (2009) 559–563
7. Wang, D., Do, H.: Computational localization of transcription factor binding sites using extreme learning machines. Soft Computing **16**(9) (2012) 1595–1606
8. Kuksa, P., Pavlovic, V.: Efficient alignment-free dna barcode analytics. BMC Bioinformatics **10** (2009)
9. Banerjee, M., Mitra, S., Banka, H.: Evolutionary rough feature selection in gene expression data. IEEE Transactions on Systems, Man, and Cybernetics, Part C **37** (July 2007) 622–632
10. Neshatian, K., Zhang, M.: using genetic programming for context-sensitive feature scoring in classification problems. Connection Science **23** (February 2011) 183–207
11. M.Vieira, S., M.C.Sousa, J., Kaymak, U.: Fuzzy criteria for feature selection. Fuzzy sets and systems **189** (2012) 1–18
12. Cervante, L., Xue, B., Zhand, M., Shang, L.: Binary particle swarm optimisation for feature selection: A filter based approach. In: IEEE World Congress on Computational Intelligence, Australia (June 2012)
13. Bing Xue, M.Z., N.Browne, W.: Particle swarm optimization for feature selection in classification: A multi-objective approach. IEEE Transactions on Cybernetics., in press (2013)
14. Enrique Alba, Jose Garcia-Nieto, L.J., Talbi, E.G.: Gene selection in canser classification using PSO-SVM and GA/SVM hybrid algorithm. In: Congress on Evolutionary Conputation, Singapour (April 2007)
15. Chuang, L.Y., Tsai, S.W., Yang, C.H.: Improved binary particle swarm optimization using catfish effect for feature selection. Elsevier-Expert Systems with Applications (2011) 12699–12707
16. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks. Volume 4. (1995) 1942–1948
17. Kennedy, J., Eberhart, R.: A discrete binary version of the particle swarm algorithm. In: IEEE International Conference on Systems, Man, and Cybernetics. Volume 5. (1997) 4104–4108
18. Sudholt, D., Witt, C.: Runtime analysis of binary pso. In: Proc. 10th annual conference on Genetic and evolutionary computation, NewYork,DC (2008) 135–142