

# DE-SVM<sub>Rank</sub>: a Differential Evolution algorithm with a rank-based feature selection process for microarray data classification

Javier Apolloni<sup>1</sup> and Guillermo Leguizamón<sup>1</sup> and Enrique Alba<sup>2</sup>

<sup>1</sup> LIDIC - Departamento de Informática - Universidad Nacional de San Luis - Ejercito de los Andes 950, 5700 - Argentina

{javierma, legui}@unsl.edu.ar,

<sup>2</sup> Lenguajes y Ciencias de la Computación - University of Malaga - Campus de Teatinos s/n, 29071, Spain

eat@lcc.uma.es

**Abstract.** DNA Microarrays are powerful tools to analyze and identify certain disease from the expression level of the genes in tissues samples. Many machine learning techniques are suitable for building predictive models to classify microarray samples into different biological categories. The accuracy of the predictive model may benefit from a relevant feature selection method and even more, if the features are ordered in terms of its relevance. In this paper, we propose a rank-based method to create the initial population in a Binary DE-SVM based algorithm used to build a predictive model. The new algorithm (DE-SVM<sub>Rank</sub>) is evaluated in terms of the achieved accuracy by the predictive model and also, the execution time required to complete the maximum number of iterations. Experimental results on public-domain microarrays show that our proposal reduces the computational time in comparison with a similar approach while providing highly competitive results.

**Keywords:** Feature Selection, Support Vector Machines, Binary Differential Evolution, Ranking of Features

## 1 Introduction

Microarrays allow biologists to register and analyze simultaneously thousands of samples of DNA from a particular tissue or cell type. We focus on the measurement of expression levels of thousands of genes in a single experiment under several conditions [19]. A typical microarray experiment collects a large amount of information. An important aim of analyzing the information is to identify functionally related genes and to classify samples into different classes (healthy or diseased).

The classification of samples is an area for the application of machine learning techniques [12]. A classifier or predictive model can be built from either a supervised or unsupervised learning technique or a mixture of both. In particular, supervised learning infers a model from labeled samples for each class. Support Vector Machine (SVM) is an algorithm capable of generating a robust and efficient and highly effective predictive model from labeled samples [9, 7].

A typical microarray data set contains a number of samples much smaller than the number of genes. Thus, an important challenge for machine learning techniques in high-dimensional data is the so-called “curse of dimensionality”. This phenomenon affects the reliability and generalization capacity of the model. Dimension reduction of the samples can enhance the performance of the model.

Feature selection is area of fundamental interest for machine learning. Feature selection techniques choose a subset of input variables (features) by eliminating redundant and irrelevant features. The machine learning and feature selection techniques can interact in different ways. In particular, the wrapper method [15] uses the learning technique to determine which features are selected. We use this method because it allows to determine interactions among features although it has a high computational cost.

The wrapper methods search for the optimal feature subset on the space of all possible feature subsets. This is an optimization problem that is known to be NP-hard [13] and the bio-inspired algorithms can be used to solve it successfully [14, 3, 20]. Differential Evolution (DE) is one of those novel algorithms designed to find a global optimum solution. DE utilizes a stochastic parallel direct search method to explore the solution space. Moreover, DE has a simple structure and high performance to tackle large-dimensional problems such as feature selection for DNA microarray data.

DE starts to explore from randomly spreading solutions along the search space. But it is known that DNA microarrays contains irrelevant genes. Since the relevance of the features can be measured in terms of a relationship between features and classes. A preprocessing method can be used to detect irrelevant genes in advance and reduce the computational time of the search process [16].

In this paper we propose a preprocessing method to extend DE-SVM approach presented by Garcia-Nieto *et al.* in [6]. Our proposal, DE-SVM<sub>Rank</sub> uses the most relevant genes to provide a new population initialization method for DE which searches for the best gene subset by employing SVM to build and evaluate a predictive model. DE-SVM<sub>Rank</sub> have been tested upon four public domain microarray data sets and the experimental results show that our proposal is highly competitive.

The rest of this paper is organized as follows, Section 2 describes some concepts of Feature Selection and Section 3 outlines the ranking of features. In Section 4 we briefly describe Differential Evolution whereas our proposal is explained in Section 5. The experimental results are presented in Section 6 and finally, Section 7 shows the conclusions and future works.

## 2 Feature Selection

Microarray data classification involves a set of  $m$  samples labeled with a class. A sample is a vector  $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]$  of  $n$  features (genes), where  $x_{ij} \in [f_{jmin}, f_{jmax}]$  is the gene expression value for the  $j$ -th feature of the  $i$ -th sample ( $1 \leq i \leq m; 1 \leq j \leq n$ ). The label associates a class value,  $c_i \in C$  with the sample  $\mathbf{x}_i$ , where  $C = \{c_1, c_2, \dots, c_k\}$  is a set of class values for the samples. In this work, we consider samples with only two possible class values ( $k = 2$ ).

A microarray data set consists of a few hundred samples with several thousands of features. A lot of these features do not contribute with information for the predictive

model. Thus, and following the definition given in [6], the feature selection is an optimization problem for which a feature selection process must find an optimal subset  $F^* \subseteq F$  where  $F = \{f_1, \dots, f_n\}$  is all the feature set.

Additionally, John *et al.* [10] define the relevance of a feature  $f_i \in F$  through a probability distribution over the feature values and the different labels. Therefore, a feature has a strong relevance when the probability distribution is affected whenever it is eliminated from the set. However, a feature may become weaker (or less relevant) under certain combination of features. Finally, a feature is irrelevant when its elimination does not affect the probability distribution.

We can conclude that a optimum feature subset only consists of relevant features. In the next section, we present a ranking method that could help in determining which features have enough relevance to be included into the optimal subset  $F^*$ .

### 3 Ranking of Features

We have seen in Section 2 that an optimal feature subset contains relevant features and the respective relevance is defined in terms of feature correlation. Thus, the mutual information provides a measurement to detect and evaluate the degree of dependence between variables.

Formally, let  $p(x)$  be the prior probability for all  $x$  in  $X$  and let  $p(x|y)$  be the posterior probability for all  $x$  in  $X$  given the value  $y$  of  $Y$  (for all  $y$  in  $Y$ ); the mutual information between two discrete random variables  $X$  and  $Y$  is given by following Equation 1 (for more details see [17]):

$$MI(X; Y) = H(X) - H(X|Y). \quad (1)$$

where  $H(X) = -\sum_i p(x_i) \log_2(p(x_i))$  is the entropy or measure of uncertainty contained in the random variable  $X$ . The conditional entropy  $H(X|Y)$  defined by  $H(X|Y) = -\sum_j p(y_j) \sum_i p(x_i|y_j) \log_2(p(x_i|y_j))$  measures the uncertainty remaining about the random variable  $X$  when it is known the value of the variable  $Y$ .

The correlation of any feature,  $f_i \in F$  and the class values,  $C$  is derived from the mutual information,  $MI(C; f_i)$ . Besides, we consider that a feature with a larger mutual information value contains more information about class values. Thus, the mutual information establishes an order on set of features  $F$ , by placing relevant features first over irrelevant ones.

Entropy involves to handle discrete random variables, then gene expression continuous values need to be discretized prior to calculate the mutual information. There exist in the literature several discretization methods, which may be categorized as either supervised or unsupervised [4]. We choose an unsupervised discretization method due to its low computational cost and its generality for application.

The simplest method is *Equal Interval Width Method* which involves a parameter  $k$  fixed in advance. This method sorts the values of  $f_i$  and divides the range of values into,  $k$  equal sized parts (bins). The bounds of each bins is defined by  $b_{ij} = f_{i_{min}} + j\Delta$  where  $\Delta = (f_{i_{max}} - f_{i_{min}})/k$  is the width of the bins and  $j = 0, \dots, k - 1$ . The method may lead to losing information because it might combine samples of different classes into one single bin. However, it has a low computational complexity.

## 4 Binary Differential Evolution

Discrete Differential Evolution defined by Gong *et al.* in [8] is an extension to the original DE [18]. The modifications involve vectors in  $D$ -dimensional discrete space and a re-definition of the differential operators. In our case, each dimension has two possible values and thus, we develop the necessary modifications for a Binary Differential Evolution (BDE).

BDE evolves a set ( $P$ ) of  $N$  binary vectors,  $\mathbf{x}_g^i \in \{0, 1\}^D$ , where  $g$  ( $g = 0 \dots \max_G$ ) is the current generation in the process of evolution, and  $i = 1, \dots, N$ . The population of vectors is initialized within the solution space. The process searches for the best solution by applying the differential operators to the vectors until a stop condition ( $\max_G$ ) is reached.

The differential mutation operator generates a mutant vector,  $\mathbf{v}_g^i$ , by randomly selecting one of the possible binary vectors that are located at a given Hamming distance (Equation 2) from the base vector,  $\mathbf{x}_g^{r1}$ :

$$d_H(\mathbf{v}_g^i, \mathbf{x}_g^{r1}) = \begin{cases} \lceil d' \rceil & \text{if } U(0, 1) < d' - \lfloor d' \rfloor \\ \lfloor d' \rfloor & \text{otherwise.} \end{cases} \quad (2)$$

where  $d' = F \cdot d_H(\mathbf{x}_g^{r2}, \mathbf{x}_g^{r3})$  and  $U(0, 1) \in [0, 1]$  is a random uniformly distributed value and  $i, r1, r2, r3$  are mutually exclusive integers randomly selected from  $\{1, \dots, N\}$ . The scale factor  $F \in [0, +\infty)$  is used to avoid the stagnation of the search process. There are different manners to interpreting  $d_H(\mathbf{v}_g^i, \mathbf{x}_g^{r1}) \in \{0, \dots, D\}$ . We consider *any-change mutation* where  $d_H(\mathbf{v}_g^i, \mathbf{x}_g^{r1})$  components from the base vector are randomly selected and flipped.

The *binomial crossover* operator generates the trial vector  $\mathbf{u}_g^i$  by following Equation 3:

$$u_g^i(j) = \begin{cases} v_g^i(j) & \text{if } U(0, 1) \leq Cr \text{ or } j = j_r, \\ x_g^i(j) & \text{otherwise.} \end{cases} \quad (3)$$

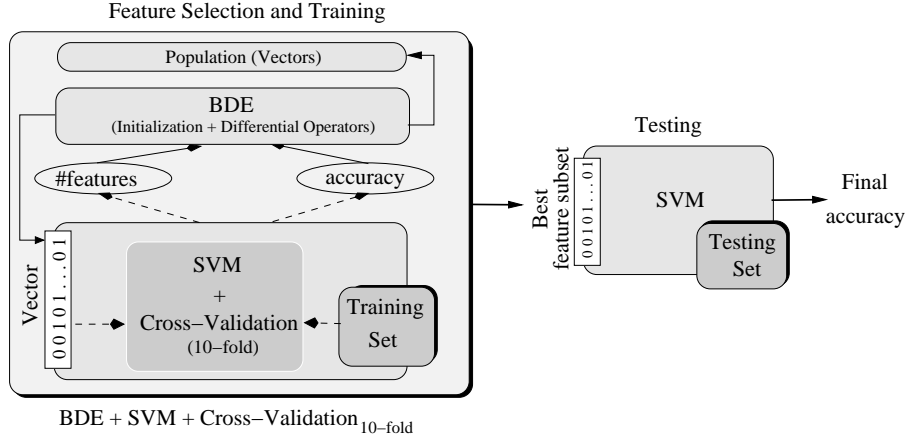
where  $Cr \in [0, 1]$  is the crossover probability and  $U(0, 1) \in [0, 1]$  is a uniform randomly distributed value and  $j_r$  ( $j_r \in \{1, \dots, D\}$ ) guarantees at least one component of the mutant vector for avoiding to duplicate  $\mathbf{x}_g^i$ .

Finally, the selection operator accepts the trial vector,  $\mathbf{u}_g^i$  for the next generation, if and only if, it leads to an improvement in the current solution,  $\mathbf{x}_g^i$ .

## 5 DE-SVM<sub>Rank</sub> Algorithm for Feature Selection

We propose to extend DE-SVM with a technique to improve the initialization of the population of DE. The general outline of our proposal DE-SVM<sub>Rank</sub>, shown in Figure 1, presents two big phases of processing. In the first phase, BDE initializes the population from a ranking of features and then, performs a search for an optimal feature subset. In the second phase, the best subset found is tested to determine the final accuracy of the feature subset.

A vector of the population represents a feature subsets and has as many components as genes in the microarray. The  $i$ -th component of the vector is set to 1 whenever  $f_i$



**Fig. 1.** Outline of DE-SVM<sub>Rank</sub> for gene selection and classification of DNA Microarrays

( $i$ -th gene) is in the subset and 0 in other case. In this paper, we propose to initialize the population as outlined in Algorithm 1.

The algorithm performs a ranking of features (line 1 to 3) followed by the initialization of the vectors (line 4 to 18). The parameter  $pPOP_{RANK} \in [0, 1]$  establishes that only a proportion of the vectors of the population are initialized from the ranking of features. The remaining vectors are initialized in a random (uniform) manner on the space. The parameter  $pGEN_{RANK} \in [0, 1]$  sets the average number of features to include initially in each vector. That is due to the size of the feature subsets affects the computational cost of the whole process. Besides, the experiments show that the optimal subset involves only a few features. Finally, the parameter  $pRank \in [0, 1]$  limits the initialization of the vector from only a proportion of the best features.

Equation 4 shows the design of the objective function that guides the evolutionary process. Based on both the size and accuracy of the feature subset, the function  $h$  assigns a fitness value to each vector of the population.

$$h(\mathbf{x}_i) = \alpha \cdot accuracy + (1 - \alpha) \cdot \#features \quad (4)$$

where  $\#features$  is the size of the feature subset and  $accuracy$  is the percentage of correctly classified training samples by the model and  $\alpha \in [0, 1]$  regulates the strength given to each objectives. After some attempts, we decide to use  $\alpha = 0.3$ .

We use a quite popular method known as *Cross-validation method* to estimate the error of the model. The method divides the training set into  $k$ -folds of equal size, each fold is left out of the SVM classifier design and used as testing set. The error estimation is computed as the overall error measured on all folds. It is desirable although difficult to achieve an unbiased model with a low variance. We have chosen  $k$  equal to 10 because the method shows both low bias and variance to a reasonable computational cost [5].

---

**Algorithm 1** Initialization of the DE population

---

• Let  $F$  be the feature set and let  $B$  be the set of bins where the values of  $f_i$  are into  $[-1, 1]$ ;

*//Ranking Process*

- 1: Calculate the frequency of  $f_j$  into each bin  $b_{jl}$  ( $f_j \in F, b_j \in B, 1 \leq l \leq k$ );
- 2: Calculate  $MI(C; f_j)$  (see Equation 1) ( $f_j \in F, 1 \leq j \leq D$ );
- 3: Sort  $F$  in decreasing  $MI$  values;

*//Initialization of the population*

- 4:  $Y \leftarrow$  Select  $N * pPOP_{RANK}$  unique random values from  $\{1, \dots, N\}$
- 5: **for** each vector  $\mathbf{x}_i$  of the population  $P$  **do**
- 6:   **if**  $i \in Y$  **then**
- 7:      $Z \leftarrow$  Select  $D * pGEN_{RANK}$  unique random values from  $\{1, \dots, D * pRank\}$
- 8:   **else**
- 9:      $Z \leftarrow$  Select  $D * pGEN_{RANK}$  unique random values from  $\{1, \dots, D\}$
- 10:   **end if**
- 11:   **for** each position  $j$  of  $\mathbf{x}_i$  **do**
- 12:     **if**  $j \in Z$  **then**  $x_i(j) \leftarrow 1$  **else**  $x_i(j) \leftarrow 0$
- 13:   **end for**
- 14: **end for**
- 15: **Output:** Population  $P$ .

---

## 6 Experimental Results

Next, we compare the performance of our proposal, DE-SVM<sub>Rank</sub> and DE-SVM approach. We study the accuracy of the optimal feature subset and the computational cost of both proposals. BDE and feature selection method are implemented using MALLBA library [1] (C++). Our proposal make use of the LIBSVM library [2] to try and validate the predictive model.

### 6.1 Data Sets

We evaluate our proposal using real-world data sets. Table 1 lists public domain microarray data sets collected in the Kent Ridge Biomedical Dataset public repository [11]. In all cases, the gene expression level values are normalized and scaled into  $[-1, 1]$ . This will ease the comparison among data sets.

### 6.2 Parameter Setting

DE-SVM<sub>Rank</sub> was executed on a heterogeneous PCs cluster with Intel processors and Linux O.S. (Debian distribution). SVM was configured with a Linear Kernel and the best parameter value was found in a pre-processing stage. Table 2(a) summarizes the kernel parameter values for each data set. A trial-and-error process was employed to adjust the parameters of BDE. The best options are outlined in Tables 2(b) and 2(c).

### 6.3 Analysis of Results

First, we display in Table 3 the number of genes in terms of the mean of the best feature subset obtained in 25 independent runs. The total number of genes of each microarray

**Table 1.** Data sets, amount of genes, and number of samples

Data set	No. of genes	Class	Training Set		Testing Set	
			No. of Samples	No. of Samples	No. of Samples	No. of Samples
Prostate Cancer	12600	Tumor	52		25	
		Normal	50		9	
Ovarian Cancer	15154	Tumor	108		54	
		Normal	65		26	
Lung Cancer	12533	MPM	16		15	
		ADCA	16		134	
Leukemia	7129	ALL	27		20	
		AML	11		14	

**Table 2.** DE-SVM<sub>Rank</sub> parameter settings

(a) SVM settings		(b) BDE settings		(c) Ranking settings	
Data sets	$C$	BDE Parameters		Ranking Parameters	
Prostate Cancer	0.03125	$F$	0.05	$pPOP_{Rank}$	0.5
Ovarian Cancer	0.03125	$Cr$	0.08	$pGEN_{Rank}$	0.07
Lung Cancer	0.03125	$N$	25	$pRank$	0.9
Leukemia	0.5	$max_G$	4000		

is summarized in the second column. The mean size of the initial subset and final subset for DE-SVM<sub>Rank</sub> is shown in the third and fourth columns respectively. We must notice that in the last column, all results confirm that the size of the original set (column two) has been reduced by an amount close to 99, 95%.

**Table 3.** Number of genes (original set and the obtained from DE-SVM<sub>Rank</sub>) and the average reduction percentage for each data set

Data Sets	No. Genes (all set)	DE-SVM <sub>Rank</sub>		
		No. genes (initial) Mean (s.d)	No. genes (final) Mean (s.d)	Reduction(%)
Prostate-Cancer	12600	428.6(±9.4)	4.2(±1.2)	99.97
Ovarian-Cancer	15154	526.2(±12.7)	6.4(±1.6)	99.96
Lung-Cancer	12533	424.8(±10.6)	3.7(±1.5)	99.97
Leukemia	7129	232.4(±8.9)	2.9(±0.5)	99.96

The accuracy is computed as the percentage of correctly classified samples over the total amount of classified samples. The accuracy for the whole feature set, using the testing set with SVM, is shown in the second column of Table 4. Columns three to five of the table summarize the values of accuracy in terms of the best feature subset obtained in 25 independent runs. The third column shows the evaluation on the training set, and the fourth and fifth columns does it on the testing set. The average accuracy

(column five) is slightly smaller than the one of the original feature set. However, note that the best execution of our proposal (column four) improves the accuracy of the original feature set in three of the four case.

**Table 4.** Accuracy Values (on training and testing set) for SVM and DE-SVM<sub>Rank</sub>

Data Sets	SVM		DE-SVM <sub>Rank</sub>	
	Ac. (all the set)	Ac. Train (%)	Ac. Final (%)	
	Testing set (%)		Mean (s.d.)	Best
Prostate-Cancer	91.1	90.1(±3.4)	100	80.8(±10.1)
Ovarian-Cancer	99.4	97.2(±2.2)	100	97.0(±2.4)
Lung-Cancer	100.0	99.0(±1.7)	99.32	90.4(±8.1)
Leukemia	85.3	99.7(±0.9)	91.17	73.5(±8.4)

In Table 5, we display the execution time required by DE-SVM<sub>Rank</sub> to perform the whole process. Note that Ovarian Cancer microarray consumes a larger amount of computer time. But it is due to it has a larger number of features and also a larger number of samples in the training set. However, the execution time is relatively low. Moreover, the initialization procedure of DE-SVM<sub>Rank</sub> (column 4) requires a negligible amount of time.

**Table 5.** Execution time (in sec.) required by DE-SVM<sub>Rank</sub> and also average time (in percentage) spent in the initialization procedure of DE-SVM<sub>Rank</sub> (25 independent runs)

Data Sets	DE-SVM <sub>Rank</sub>		
	Total Time (sec.)		Ini. Time (%)
	Best	Mean (s.d)	
Prostate-Cancer	$9,59E+2$	$1.02E+3 (\pm 4.03E+1)$	0.004
Ovarian-Cancer	$1,92E+3$	$2.00E+3 (\pm 1.10E+2)$	0.003
Lung-Cancer	$2,83E+2$	$2.86E+2 (\pm 6.54E+0)$	0.009
Leukemia	$1,85E+2$	$1.84E+2 (\pm 4.91E+0)$	0.009

Finally, we evaluate and compare DE-SVM<sub>Rank</sub> and DE-SVM with regard to the execution time. In the case of DE-SVM, it was used the same parameter settings as for DE-SVM<sub>Rank</sub>. Although 5 independent runs were done because each one of them requires too much time to complete all the process. The results are summarized in Table 6. Note that our proposal notably reduces (up to 90%) the time consumed by DE-SVM to process the same microarrays.

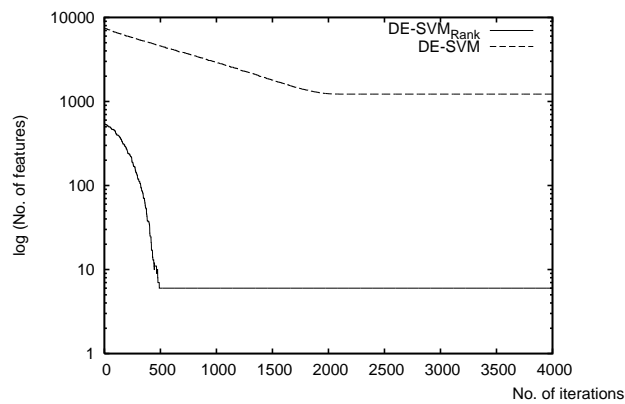
Moreover, we compare both approaches with regards to the number of genes of the best feature subset along the process. For that, we select the median of the runs on both approaches, for the Ovarian Cancer microarray. Note in Figure 2, that the predictive model may benefit from the ranking technique as it selects a smaller number of features than DE-SVM.



**Table 6.** Execution time required by DE-SVM (5 independent runs) and percentage of reduction time by using DE-SVM<sub>Rank</sub> (25 independent runs)

Data Sets	DE-SVM	DE-SVM vs. DE-SVM <sub>Rank</sub>
	Time (sec.) mean (s.d.)	Reduction (%)
Prostate-Cancer	$5.10E^4 (\pm 6.96E^3)$	98.00
Ovarian-Cancer	$6.34E^4 (\pm 1.03E^4)$	96.85
Lung-Cancer	$4.58E^3 (\pm 4.27E^1)$	93.77
Leukemia	$2.06E^3 (\pm 2.10E^1)$	91.09

**Fig. 2.** Number of genes along the evolutionary process in a typical execution (median) of DE-SVM and DE-SVM<sub>Rank</sub> on Ovarian Cancer microarray experiment.



## 7 Conclusions and Future Work

In this paper, we have presented DE-SVM<sub>Rank</sub>, an extension to DE-SVM algorithm. Our proposal involves a new population initialization scheme for DE based on a ranking of the features. We have compared the performance of our proposal with DE-SVM on public domain microarrays. Generally, DE-SVM<sub>Rank</sub> obtained a higher accuracy than DE-SVM and that the predictive model from the original feature set. We also found that the execution time is drastically reduced from the initialization based on feature ranking. The feature selection process and the accuracy of the predictive model also is benefited from the initialization procedure.

As future work, we plan to employ another techniques of ranking more specific without increasing the execution time. Also, we plan to utilize other microarray experiments and compare it with other state-of-the-art techniques.

**Acknowledgments.** J. Apolloni and G. Leguizamón acknowledge the support of the Universidad Nacional de San Luis and the LIDIC (Laboratory of Investigation and Development in Computational Intelligence). The third author acknowledges funds

from the CICE of the Junta de Andalucía, under contract P07-TIC-03044 (DIRICOM <http://diricom.lcc.uma.es>) and Spanish Ministry of Sciences and Innovation (MICINN) and FEDER under contract TIN2011-28194 (RoadMe <http://roadme.lcc.uma.es>).

## References

1. Alba, E., Luque, G., García-Nieto, J., Ordóñez, G., Leguizamón, G.: MALLBA: A software library to design efficient optimisation algorithms. *Int. J. of Innovative Computing and Applications* 1(1), 74–85 (2007)
2. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines (2002), <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
3. Chen, S., Zeng, X., Li, G., Yang, J., Yang, M.: Redundant gene selection based on particle swarm optimization. In: *Bioinformatics, Systems Biology and Intelligent Computing, 2009. IJCBS '09*. pp. 10–16 (2009)
4. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: *Prieditis, A., Russell, S. (eds.) Proceedings of the Twelfth International Conference on Machine Learning*. pp. 194–202 (1995)
5. Duval, B., Hao, J.K.: Advances in metaheuristics for gene selection and classification of microarray data. *Briefings in Bioinformatics* pp. 127–141 (2009)
6. Garcia-Nieto, J., Alba, E., Apolloni, J.: Hybrid DE-SVM approach for feature selection: Application to gene expression datasets. In: *Logistics and Industrial Informatics, 2009. LINDI 2009. 2nd International*. pp. 52–57 (2009)
7. Garcia-Nieto, J., Jourdan, L., Talbi, E.: Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms. *IEEE Con. on Ev. Comput.* pp. 284–290 (2007)
8. Gong, T., Tuson, A.L.: Differential Evolution for Binary Encoding, pp. 251–262. *Advances in Soft Computing, Springer Berlin/Heidelberg* (2007)
9. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. *Mach. Learn.* 46(1-3), 389–422 (2002)
10. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: *ICML'94*. pp. 121–129 (1994)
11. Kent Ridge Biomedical Dataset Public Repository: <http://datam.i2r.a-star.edu.sg/datasets/krbd/index.html>
12. Mitchell, T.: *Machine Learning*. McGraw-Hill, New York (1997)
13. Narendra, M., Fukunaga, K.: A branch and bound algorithm for feature subset selection. In: *IEEE Trans. on Computer.* vol. 26, pp. 917–922, (1977)
14. Price, K.V., Storn, R., Lampinen, J.: *Differential Evolution: A Practical Approach to Global Optimization*. Springer, London (2005)
15. Saeys, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. *Bioinformatics* 23(19), 2507–2517 (2007)
16. Sharma, A., Imoto, S., Miyano, S.: A top-r feature selection algorithm for microarray gene expression data. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* 9(3), 754–764 (2012)
17. Song, Q., Ni, J., Wang, G.: A fast clustering-based feature subset selection algorithm for high dimensional data. *Knowledge and Data Engineering, IEEE Transactions on* 99, 1 (2011)
18. Storn, R., Price, K.: *Differential Evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces*. Tech. rep. (1995)
19. Stoughton, R.: Applications of DNA microarrays in biology. *Annu Rev Biochem* pp. 53–82 (2005)
20. Wang, W.X.C.: A hybrid improved ant colony optimization and random forests feature selection method for microarray data. In: *INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on*. pp. 559–563 (2009)