Adithya Sagar
asg242@cornell.edu

A hybrid of discrete particle swarm optimization and support vector machine for gene selection and molecular classification of cancer

Adithya Sagar
Cornell University, New York

1.0 Introduction:

Cancer therapies require classification of cancers to target specific cancers with specific treatments. Thus the improvements in cancer therapies have been linked to improvements in cancer classification. To enhance the efficiency of the treatment it is important to identify the specific markers that are to be targeted in a treatment. Apart from enhancing the efficiency of treatment, targeting of specific markers allows for the minimization of toxicity resulting from the treatment. The advent of micro array technologies has greatly aided in the identification of specific genes through the measurement of gene expression data. Current micro array technologies allow for the measurement of thousands of gene expression levels from a single sample simultaneously [9]. However in almost all the cases the number of samples considered is far less than the number of genes measured. This often causes the problem of overfitting during classification. Thus it is important to reduce the dimensionality of the sample before using it for classification [3]. Moreover from a diagnostics perspective it is important to isolate the specific genes so that a specific diagnostic setup and treatment setup may be developed to predict, classify and treat such a cancer. This also helps in reducing the cost of treatment [1].

To achieve the above objectives various feature selection methods in combination with various classification tools have been used [1, 3, 4, 15, 16, 17, 18]. Some of the prominent methods that have been used to classify data from micro-arrays have been k-nearest neighbors (KNN), nearest centroid, linear discriminant analysis (LDA), neural networks (NN) and support vector machines (SVM). For selection of subsets of genes, feature selection methods such as t-test, Principal component analysis (PCA), individual gene selection, pair-wise gene selection, non parametric scoring and now recently evolutionary computing algorithms such as genetic algorithms (GA's) and particle swarm optimization (PSO) are being applied [9].

Currently the gene selection methods may be classified into two categories. One, a filter approach wherein each gene is considered independently evaluated according to a specific criteria and then ranked accordingly based on its score. The top ranked genes are considered for while evaluating the classification accuracy of the classifier. Prominent approaches in this category include t-test filtering, SNR filtering, PCA etc.[9]. Second is the wrapper approach wherein, selection of a subset of genes and classification is performed in the same process. A subset of genes are considered and evaluated based on the classifier's performance. This process is carried out recursively till the desired classification accuracy is obtained. Evolutionary algorithms like PSO [15, 16, 18], GA [16] have been used in conjunction with the SVMs. Guyon et. al demonstrated a recursive feature elimination method (RFE) –SVM [3] based wrapper. Though the wrapper approach results in the improvement of classification accuracy, a major problem is the computational cost associated while using the wrapper. It is important to use algorithms that traverse the search space efficiently with reduced computational costs. Thus PSO is used here which when compared to GAs or RFE, is simpler, faster and converges to an optimum quickly. However PSO has certain drawbacks like converging to a local optimum, reduction in convergence rate while approaching optimum etc. To this end a novel discrete PSOSVM is proposed that not only avoids local optima but also converges to a global optimum quickly and demonstrates enhanced classification accuracy.

2.0 Methods

2.1 Particle Swarm Optimization

Particle swarm optimization (PSO) is a stochastic global optimization technique developed by Eberhart and Kennedy in 1995 based on social behavior of birds [2]. In PSO a set of particles or solutions traverse the search space with a velocity based on their own experience and the experience of their neighbors. During each round of traversal, the velocity, thereby the position of the particle are updated based on the above two parameters. This process is repeated till an optimal solution is obtained. According to the original PSO the particle velocity and position are updated according to the following equations.

$$v_k^{in+1} = v_k^{in} + c_1 r_1 \left(pbest_k^{in} - p_k^{in}\right) + c_2 r_2 \left(gbest^{in} - p_k^{in}\right) \qquad (1)$$

$$x_k^{in+1} = x_k^{in} + v_k^{in+1} \qquad (2)$$

where $v_k^{i\ n}$ and $p_k^{i\ n}$ are the velocity and position of kth particle in ith dimension during nth iteration, pbest is the best position experience by the particle upto that iteration and gbest is the best position experience by all particles upto that iteration. The best positions of a particle are evaluated according to a fitness function.

c_1, c_2 are called acceleration constants usually equal to 2 and r_1 and r_2 are random numbers uniformly distributed in (0, 1). Thus these constants are a measure of inertia experienced by the particle.

The PSO developed by Eberhart and Kennedy is suited for continuous optimization problems. The current problem requires a discrete version of the PSO as the features here are genes which are discrete entities. To address this problem Q.Shen [15] developed a discrete version of PSO and applied it to gene selection. Each particle contains n number of features wherein each feature or position is assigned 0 or 1. An assignment of 1 corresponds to the selection of the feature and an assignment of 0 corresponds to its rejection. In Shen's approach velocity of a particle in a dimension for a given iteration is generated randomly between 0 and 1. Thereby position of each particle is updated according to the following rules

$$0 \leq v_k^{in} < 0.4;\ p_k^{in}(new) = p_k^{in}(old) \qquad (3)$$

$$0.4 \leq v_k^{in} < 0.6;\ p_k^{in}(new) = pbest_k^{in} \qquad (4)$$

$$0.6 \leq v_k^{in} < 1;\ p_k^{in}(new) = gbest^{in} \qquad (5)$$

Yu et. al [18] also followed the same update rules as suggested by Shen. However to avoid converging to a local optimum they used a variable to store continuous unchangeable values of particle best values. If a particle has the same number of particle best values consecutively for a fixed number of times, the particle best was set to zero. This was done to allow the particles to escape local optima. Alba et. al used geometric particle swarm optimization which applied a 3-parent mask based crossover to move the particle [17].

The current approach however uses update rules for particles that differ from the ones used above. It uses a linear combination of current position, particle best position and global best position to determine the next position of a particle. Each particle position is a vector whose features are binary valued. For example (1,0,1,1,1,0,0…..1) is a position vector of the particle where 1 represents selection of the corresponding gene and 0 represents rejection. The subsequent position vector is determined by a linear combination of three vectors, the particle's current position vector, best position vector of the particle and the best position vector among all particles.

$$x_k^{in+1} = w_1 x_k^{in} + w_2 pbest_k^{in} + w_3 gbest^{in} \qquad (6)$$

where w_1, w_2, w_3 are probabilities assigned to current position, particle best and global best such that $w_1+w_2+w_3=1$.

2.2 Support Vector Machines

Support vector machines are a class of linear learning machines used for classification and regression [4]. In binary classification problems SVM constructs a maximal margin separating hyperplane to separate the input data points into classes.

The separating hyper-plane is of the form

$$< w, x_i > + b = 0 \qquad (7)$$

where w is the normal vector to the hyper-plane.

Since it is a binary classification problem the two classes can be denoted with +1 and -1.
We can select two hyper-planes of the margin in a way that there are no points between them and then try to maximize their distance.

Thus the parallel hyper-planes are of the form

$$< w, x_i > +b = 1 \qquad (8)$$

$$< w, x_i > +b = -1 \qquad (9)$$

The distance between the hyper-planes i.e. 2/||w||(separating and the ones parallel to it) is to be maximized. To exclude the data points we have

$$< w, x_i > +b \geq 1 \qquad ; \qquad < w, x_i > +b \geq -1$$

These conditions can be rewritten as

$$y_i(< w, x_i > +b) \geq 1 \qquad (10)$$

Thus constructing the Lagrangian the whole optimization problem can be reconstructed as

Minimize:

$$\frac{1}{2} < w, w > - \sum c_i y_i [(< w, x_i > +b) - 1] \qquad (11)$$

$$c_i \geq 0 \qquad (12)$$

The corresponding dual representation is

Maximize

$$\sum c_i - \sum \frac{1}{2} c_i c_j y_i y_j < x_i, x_j > \qquad (13)$$

$$c_i \geq 0 \qquad (14)$$

$$\sum c_i y_i = 0 \qquad (15)$$

wherein the inner product may be replaced with an appropriate kernel to map the input vectors to a higher dimension to construct a maximal hyperplane. For further theory refer to Vapnik et al[3].

SVM is widely used in bioinformatics for cancer classification, protein fold class prediction, in predicting protein-protein interactions etc. However in most of these problems the number of features per sample is much greater than the number of samples available. This usually creates a problem of overfitting with the SVMs and leads to a reduction in classification accuracy. It has been observed that reduction in number of features leads to an increase in accuracy and thus feature selection is preceded before classification. [1,3,4]

2. 3 PSOSVM for cancer classification

Here a novel particle swarm optimization support vector machine hybrid is proposed for a molecular-level based classification of cancer and a subsequent selection of gene markers important in recognizing cancerous tissues.

Examples of particles in discrete space

Particle 1	1 0 0 0…………………………………………………………………7129 times
Particle 2	0 0 1 0……………………………………………………………………….1 0 1 1 1
Particle 3	1 1 0 1……………………………………………………………………….0 1 0 1 1

Example of training data set

Sample 1	0.6 0.9 1.0 2.0……………………………………………………………7129 digits
Sample 2	0.1 0.3 0.6 0.4…………………………………………………………………….
Sample 3	0.5 0.4 0.3 0.6…………………………………………………………………….

Corresponding training subset for particle 1 (since first four digits are 1 0 0 0)

Sample 1	0.6 ………………………………………………………………
Sample 2	0.1 …………………………………………………………………….
Sample 3	0.5 …………………………………………………………………….

Corresponding training subset for particle 2(since first four digits are 0 0 1 0)

Sample 1	1.0 ………………………………………………………………
Sample 2	0.6 …………………………………………………………………….
Sample 3	0.3 …………………………………………………………………….

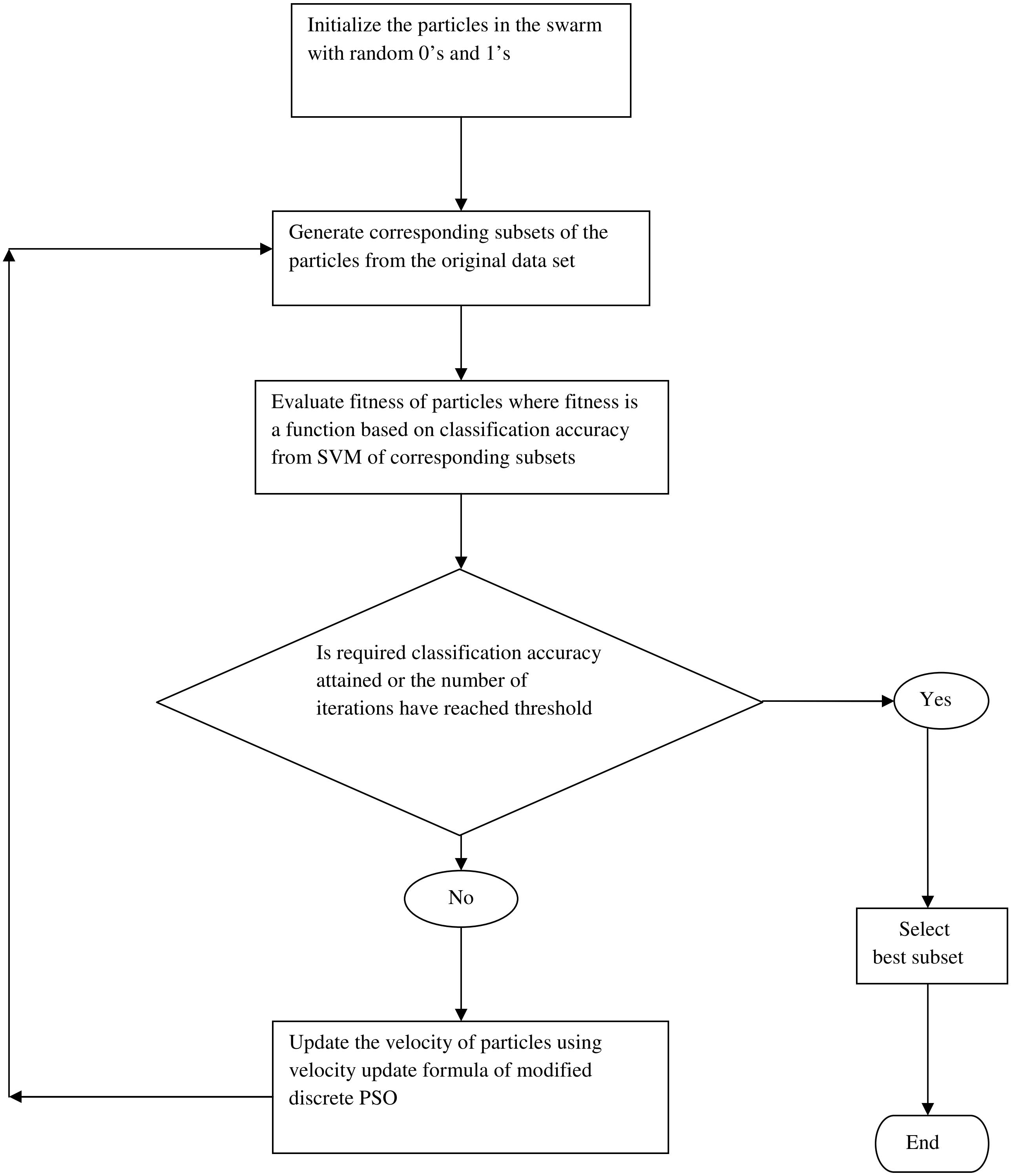
Initialize the particles in the swarm with random 0's and 1's
Generate corresponding subsets of the particles from the original data set
Evaluate fitness of particles where fitness is a function based on classification accuracy from SVM of corresponding subsets
Is required classification accuracy attained or the number of iterations have reached threshold
Yes
No
Select best subset
Update the velocity of particles using velocity update formula of modified discrete PSO
End

Figure2. Flow chart of PSOSVM

In this method a fixed population of particles is considered, wherein each particle has the same number of dimensions as the number of genes in the data set considered. For example a position vector of an n-dimension particle would be of the following form. $p_k=\{p_k^1, p_k^2,p_k^3,\ldots\ldots\ldots , p_k^n\}$ where p_k^i is either 0 or 1. Thus the total number of dimensions of a particle is equal to the total number of genes of a sample. Thereafter each particle is initialized randomly with 0's and 1's. The bit 1 when assigned causes the selection of corresponding gene and bit 0 causes the gene to be discarded. This generates a new feature subset corresponding to the particle under consideration. The subset corresponding to the particle has only those genes to which bit 1 has been assigned. Hence for a fixed population of N particles N corresponding subsets are generated. The fitness of each particle here is a function of classification accuracy. Unlike the previous works of (Shen et. al, Yu et. al, Alba et. al) wherein Leave One Out Cross Validation accuracy (LOOCV) is considered as a measure of fitness, here classification accuracy is the fitness value of the particle as ultimately it is the classification accuracy resulting on a test which is required to be optimized*. For each training subset generated the SVM is trained on the training subset and then tested on the corresponding test subset to obtain the classification accuracy. This approach considers also considers all the genes and significantly differs from past works where a fixed number of genes are selected initially and then remaining genes are input into the wrapper [1,3,15,16,18].

Particle Swarm Optimization generally faces the problem of converging to a local optimum. To avoid this problem a two pronged approach was considered. (1) For 60% of the particles considered all the features are initialized randomly with 0s and 1s. For 10% of the particles only 20 genes are selected randomly and the rest discarded, for another 10% 30 genes are selected and for the last 10%, 150 genes are selected. This results in an initialization diversity thus generating different position vectors with varying fitness values. (2) The update rule given by equation (6) is applied only to 80% of the particles, for the rest of the particles genes are selected randomly. This step allows for the rest of the 80% of particles to diverge out of local optima.

The probability weights considered can be grouped into two sets for current position, particle best position and global best position. A) (w1,w2, w3)=(0.33,0.34,0.33) and B)(w4,w5,w6)=(0.25,0.45,0.3). The first set was used for 80% of iterations and the next set was used for the rest of the 20% of iterations. Increase in probability weight corresponding to particle best value was to allow for particles to move to their best positions at a faster rate.

SVM^{perf} was considered for generating classification accuracies http://svmlight.joachims.org/svm_perf.html and for generating cross validation accuracy for the data sets LIBSVM www.csie.ntu.edu.tw/~cjlin/libsvm/ was considered. The PSOSVM code was written in C and run on a MACOSX platform with 4GB RAM and 2.66 GHz processor.

3.0 Datasets

The primary data set considered here was the (acute lymphoblastic leukemia) ALL-(acute myeloid leukemia) AML data set. The data set consisted of a training data set containing 27 ALL samples and 11 AML samples and a testing data set containing 20 ALL samples and 14 AML samples. Each sample contained expression levels from 7129 genes. The data was then normalized and converted to a format suitable for input to the SVM. The raw unprocessed data is available http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi. The corresponding reference paper is ‘Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring’ by Golub et al.

Other data sets that were tested only for cross validation accuracies but not for a biological analysis are as follows

http://levis.tongji.edu.cn/gzli/data/mirror-kentridge.html

1. Colon cancer dataset 62 samples (40 tumors, 22 normal), 2000 genes
2. Lung cancer dataset 181 samples(150 ADCA, 31 MPM), 12533 genes
3. Ovarian cancer data set 253 samples(162 cancerous, 91 normal), 15154 genes
4. Prostate cancer data set 136 samples(77 tumor, 59 non tumor), 12600 genes

4. Results and Discussion

4.1 Results

All the 7129 genes were considered for each of the 38 samples in the training data set. SVM was applied to the training data sets choosing different kernels and different tradeoff coefficients (C value). SVM performed best for linear kernel at C=100. The results are summarized below. For a linear kernel the resulting testing accuracy was 71.43% at C=0.01. For a linear kernel the testing accuracy at C=100 was 85.29%

Table1. LOOCV at C=0.01 (default) for different kernels

Kernel	**Leave One Out Cross Validation Accuracy**		
	Error	**Recall**	**Precision**
Linear	13.16	100	84.38
Polynomial	28.95	100	71.05
Radial Basis	28.95	100	71.05
Sigmoid	26.92	100	72.97

Table2. LOOCV at C =100 for different kernels

Kernel	Leave One Out Cross Validation Accuracy		
	Error	Recall	Precision
Linear	5.26	100	93.10
Polynomial	28.95	100	71.05
Radial Basis	28.95	100	71.05
Sigmoid	42.11	77.78	67.74

The parameters for PSOSVM were set as follows:

1. Number of particles = 100
2. Number of iterations = 50
3. Probability weights (w1,w2,w3,w4,w5,w6) =(0.33,.34,.36,0.25,0.45,0.3)

Using the above parameters PSOSVM was applied to the data sets. The best test set classification accuracy obtained was **94.12%.** The experiment was repeated in using the same parameters making a 5% modification in the weights and the best classification accuracy obtained in all the cases was 94.12%. The number of genes obtained in each of these cases was 150 with a variation of 2%. This is the first time that classification accuracy from the test set is being considered for evaluating the fitness of a particle instead of depending on cross validation accuracies

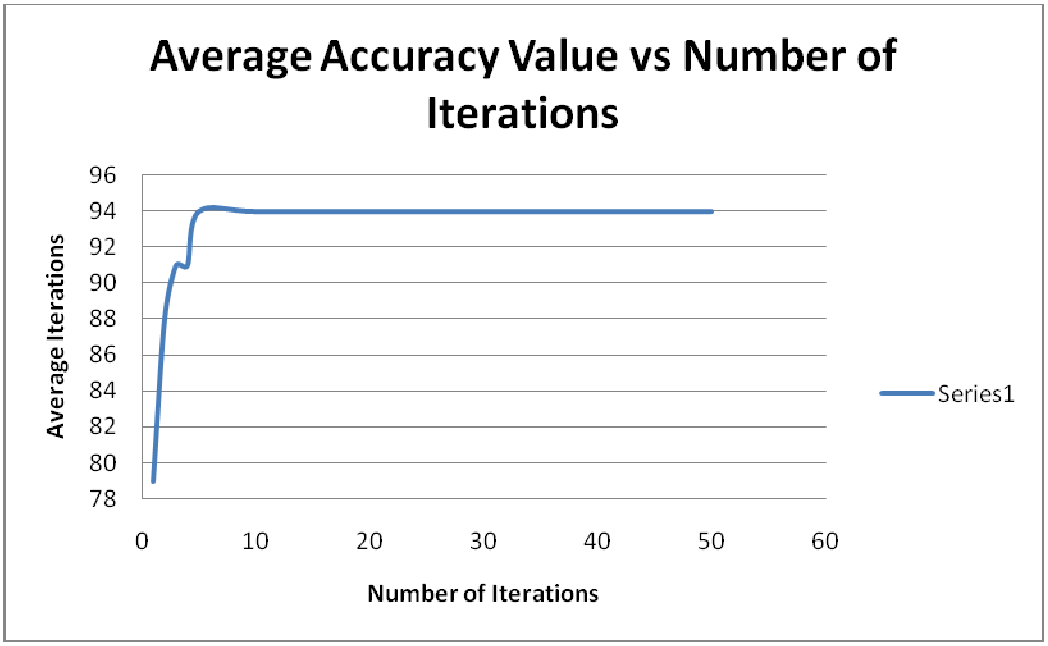

Figure3. Accuracy Value vs Number of Iterations

In order to make a comparison with results from other studies a 5-fold cross validation accuracy was used a fitness function value and the PSOSVM was applied to the AML-ALL training data set and other datasets. The results are displayed in the table below

Table3. Comparison of cross validation accuracies for different datasets across various algorithms

Dataset	PSOSVM	GPSOSVM	GA+SVM	Shen's Method	Yu's discrete PSOSVM	SNR+SVM	SVM
AML-ALL	**97.4**	97.38	97.27	98.08	NA	NA	85.2
Colon	**94.12**	100	93.55	95.2	96.77	87.1	90.3
Lung	**91.1**	99.44	NA	NA	NA	NA	**50%**
Ovarian	**97.12**	99.44	NA	NA	NA	NA	**99.205%**
Prostate	**94.12**	98.66	NA	NA	NA	NA	**50.8%**

4.2 Discussion

This approach for the first time attempts to optimize the classification accuracy of the test set rather than the cross validation accuracy of the training sets. The resultant optimized classification accuracy for the AML-ALL test set data has been found to be 94.12 %. In order to make a comparative analysis with other algorithms, the five fold cross validation accuracy for the AML-ALL training data has been found using this approach. The value is equal to 97.38 % and is comparable to the k-fold accuracy values obtained from other approaches as shown in table 3. The other values have been comparatively lower with respect to GPSOSVM. It might be possible to improve the resultant accuracies by adjusting the parameters like number of particles randomized, probability weights, number of iterations etc. specific to each data set.

The graph of average accuracy values versus the number of iterations shows that the optimum accuracy value is reached within the first 10 iterations and remains constant thereafter. Thus the algorithm is quick to find the global optimum in the search space. To ensure that the optimum is the global optimum 20% of the particles were updated randomly and did not follow the update rules as prescribed in equation 6. The number of iterations was set to be equal to 50 so as to allow the particles to emerge out of false optima. However results showed that even at the end of 50 iterations the accuracy value remains constant and was equal to the one obtained in the first 10 iterations. PSOSVM was run on the training and test data sets for about 20 times and the optimum subsets that generated the maximum classification accuracy contained about 150 genes with a variation of nearly 2% in all the cases. Previous approaches that obtained 4-6 genes [Ref] initially removed a set of genes based on a ranking from signal to noise ratio (SNR) and then applied the wrapper to the existing subset of genes. The current approach considers all the genes and then selects an optimum subset. These subsets are not unique as there may be redundancies in the gene subsets i.e. gene 6 and gene 900 may have the same biological significance but result in a different subset contribution. The elimination of genes based on SNR takes care of these redundancies, thus possibly resulting in subsets with lower genes. In the current method to identify the genes that may be important in cancer pathways all the subsets that generated optimum accuracy were considered and the most frequently occurring genes were selected. Gene numbers 804, 2896, 4823, 4849 were found to be the most frequently occurring genes.

Table4.

Gene Number	Gene Accession Number	Gene Description
804	HG1612-HT1612_at	Macmarks
2896	U20362_at	Tg737 mRNA
4823	X94232_at	Novel T-Cell Activation Protein
4849	X95735_at	Zyxin

To further investigate the variance of accuracy values with number of genes in the subset, a basic form of PSOSVM that converged to a local optima was run on AML-ALL training data sets. The numbers of genes in each subset were considered to be input training data set to the subsequent run of PSOSVM and the corresponding accuracy values were observed. This was process was continued till the classification accuracy kept increasing. The classification accuracy peaked when the number of genes in the subset was in the range of 30-200. The leave one out cross validation accuracy ranged between 97%- 100% and the maximum classification accuracy obtained was 94%. These results are in agreement with the results obtained by applying PSOSVM optimizing test set classification accuracy as shown previously.

Table5.

Number of Particle	Number of iterations	Number of GENES	LOOCV	Classification Accuracy(Best subsets only)
50	100	3500-4000	94, 94, 94, 94, 94,94, 94, 94, 94, 94	77
50	100	1600-2000	94, 94, 94, 94, 97, 94, 97, 94, 94, 94	77, 79
50	100	800-1000	94, 97,94,97,94, 94, 94, 97, 97, 94	79, 82
50	100	450-600	97, 97, 97, 94, 97, 97,94, 97, 97, 97	82, 85
50	100	150-300	97, 97, 97, 100, 94, 97, 97, 94, 97, 97	82, 85, 88, 91
50	100	50-200	97, 97, 100, 97, 100, 97, 97, 94, 97, 97	82, 85, 88, 91,94
50	100	30-150	97, 97,100, 97, 100, 94, 100, 97, 97	88,91,94
50	100	15-30	97, 94, 97, 100,100, 97, 97, 97, 94, 100	82, 85, 88, 91
50	100	5-20	94, 92 ,97, 97, 94, 92, 94, 94, 97,92	79,82,88

5. Conclusion

PSOSVM hybrid considered all the genes in the sample and optimized the test set classification accuracy. The solution generated by the algorithm was successful in converging a global optimum within a few iterations. The optimal subsets generated by the algorithm contained about 150 genes in number. Possible redundancies in the number of genes are speculated to be the cause for generating high number of genes in optimal subsets. Considering all the optimal subsets the most frequently occurring genes were considered to be the most important marker genes. To validate the number of genes in the optimal subset, a subset of subset approach was performed which showed that the maximum cross validation accuracy and the resultant test set classification accuracy occurs when the numbers of genes in the subset are in between 50-200.

The algorithm for the first time optimized the test set classification accuracies rather than the cross validation accuracies for the training set. In order to do a comparative analysis the 5-fold cross validation accuracies for various data sets was generated using this PSOSVM. The resulting accuracy values are in near agreement with the ones found in literature. Further direction of work could consider applying the same to other biological data sets and carry out pertinent analysis of gene subsets. Another possible direction of work would be to reduce the number of genes obtained in each subset by integrating a filter approach within the PSOSVM wrapper.

6. References

1. T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, Science 286 (1999) 531.
2. M. Clerc, J. Kennedy, Proceedings of the IEEE Transactions on Evolutionary Computation, vol. 6, 2002, p. 58.
3. I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Mach. Learn. 46 (2002) 389.
4. V.Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
5. O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, “Choosing multiple parameters for support vector machines,” *Machine Learning*, vol. 46, pp. 131□159, 2002.
6. J. Kennedy, R.C. Eberhart, Particle Swarm optimization, in: Proc. of IEEE Int. Conf. on Neural Networks, Perth, Australia (1995)
7. J. Kennedy, R. Eberhart, A discrete binary version of the particle swarm optimisation algorithm, in: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 1997, pp. 4104–4108.
8. Q. Shen, J. Jianhui, Modified particle swarm optimisation algorithm for variable selection in MLR and PLS modeling: QSAR studies of antagonism of antagonism of angiotensin II antagonists, European Journal of Pharmaceutical Sciences 22 (2004) 145–152.

9. Microarray bioinformatics-Dov Stekel

10. B. Schölkopf, A. Smola, R. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12, 2000, 1207-1245

11. Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

12. Yang, J. and Honavar, V. (1998). Feature subset selection using genetic algorithm. *Journal of IEEE Intelligent Systems*. 13. 44 – 49

13. Rätsch, G., Onoda, T. and Müller, K., R. (2001). Soft Margins for AdaBoost. *Machine Learning*. 42(3). 287 – 320.

14. Müller, K. R., Mika, S., Rätsch, G., Tsuda, K. and Schölkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*. 12(2). 181 – 201.

15. Qi Shen, Wei-Min Shi, Wei Kong, Bao-Xian Ye. A combination of modified particle swarm optimization algorithm and support vector machine for gene selection and tumor classification. Talanta 71 (2007) 1679–1683

16. Enrique Alba, Jos´e Garc´ıa-Nieto, Laetitia Jourdan and El-Ghazali Talbi. Gene Selection in Cancer Classification using PSO/SVM and GA/SVM Hybrid Algorithms Evolutionary Computation(2007) 284-290

17. T. Joachims, *Training Linear SVMs in Linear Time,* Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), 2006.

18. Hualong Yu, Guochang Gu, Haibo Liu, Jing Shen, Changming Zhu. A Novel Discrete Particle Swarm Optimization Algorithm for Microarray Data-based Tumor Marker Gene Selection. Proceedings of the International Conference on Computer Science and Software Engineering - Volume 01 (2008) 1057-1060