Memetic Algorithms For Feature Selection On Microarray Data

Zexuan Zhu[1,2] and Yew-Soon Ong[1]

[1] Division of Information Systems, School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798
[2] Bioinformatics Research Centre, Nanyang Technological University, Research TechnoPlaza, 50 Nanyang Drive, Singapore 637553

Abstract. In this paper, we present two novel memetic algorithms (MAs) for gene selection. Both are synergies of Genetic Algorithm (wrapper methods) and local search methods (filter methods) under a memetic framework. In particular, the first MA is a Wrapper-Filter Feature Selection Algorithm (WFFSA) fine-tunes the population of genetic algorithm (GA) solutions by adding or deleting features based on univariate feature filter ranking method. The second MA approach, Markov Blanket-Embedded Genetic Algorithm (MBEGA), fine-tunes the population of solutions by adding relevant features, removing redundant and/or irrelevant features using Markov blanket. Our empirical studies on synthetic and real world microarray dataset suggest that both memetic approaches select more suitable gene subset than the basic GA and at the same time outperforms GA in terms of classification predictions. While the classification accuracies between WFFSA and MBEGA are not significantly statistically different on most of the datasets considered, MBEGA is observed to converge to more compact gene subsets than WFFSA.

1 Introduction

Over the recent years, microarray technology has attracted increasing interest in many academic communities and industries. One major application of microarray technology lies in cancer classification. Thus far, a significant amount of new discoveries have been made and new bio-markers for various cancer have been detected from microarray data. However, due to the nature of the microarray gene expression data, cancer classification has remained a great challenge to computer scientists.

Microarray data is characterized with thousands of genes but with only a small number of samples available for analysis. This makes learning from microarray data an arduous task under the effect of curse of dimensionality. Furthermore, microarray data often contains many irrelevant and redundant features, which affect the speed and accuracy of most learning algorithms. Therefore, feature selection (also commonly known as gene selection in the context of microarray) is widely used to address these problems.

In general, feature selection methods can be categorized into filter and wrapper methods [1]. Filter methods assess the merits of features according to their

individual relevance or discriminative power with respect to the target classes. Since these methods do not involve the induction algorithm, they are relatively inexpensive to compute. Wrapper methods, on the contrary, use the induction algorithm itself to evaluate the candidate feature subsets. They select feature subsets more suitable for the induction algorithm, generally at the expense of a higher computational time when compared to filter methods. One key issue of wrapper method is how to search the space of feature subsets. On microarray data, as the number of genes (features) are typically very large, most of existing search methods (e.g., complete search, heuristic search, and random search) face the problems of intractable computational time.

Genetic Algorithm (GA) has well known ability to produce high quality solution within tractable time even on complex problems [2–6]. It has been naturally used for gene selection and has shown promising performance in dealing with microarray data [7]. Unfortunately, due to the inherent nature of GA, it often takes a long time to locate the local optimum in a region of convergence and may sometimes not find the optimum with sufficient precision. One way to solve this problem is to hybridize GA with some memetic operators (also known as local search operators) [8–10] which are capable of fine-tuning and improving the solutions generated by the GA more precise and efficient. This form of evolutionary algorithms are often referred to as Memetic algorithms (MAs) [8, 9].

In this paper, we present a comparison study on two MAs we have recently proposed for gene selection [11, 12] on synthetic and real microarray data. Both are synergies of the Genetic Algorithm (wrapper methods) and local search methods (filter methods) under a memetic framework. In particular, the Wrapper-Filter Feature Selection Algorithm (WFFSA) [11] fine-tunes the population of GA solutions by adding or deleting features based on univariate feature filter ranking method. The second MA approach, Markov Blanket-Embedded Genetic Algorithm (MBEGA) [12], fine-tunes the GA population by adding the relevant features, removing the redundant and/or irrelevant features using Markov blanket technique.

The rest of this paper is organized as follows. Section 2 presents the two MAs proposed for gene selection. Section 3 presents the numerical results obtained from our empirical study on synthetic and real microarray datasets. Analysis of the numerical results and some discussions are also presented in the section. Finally, Section 4 concludes this paper.

2 System and Methodology

In this section, we describe the memetic framework of WFFSA and MBEGA. The local search method or filter method or otherwise known also as meme used in the WFFSA and MBEGA are then described in subsection 2.1.

The proposed Memetic framework for gene selection is briefly outlined in Figure 1. At the start of the MA search, a population of potential gene subset solution is generated randomly with each chromosome encoding a candidate gene subset. In the present work, each chromosome is composed of a bit string

of length equal to the total number of features or genes in the problem of interest. Using binary encoding, a bit of '1' ('0') implies the corresponding gene is selected (excluded). The fitness of each chromosome is then obtained using an objective function based on the induction algorithm:

$$Fitness(c) = J(S_c) \tag{1}$$

where S_c denotes the selected gene subset encoded in a chromosome c, and the gene selection objective function, $J(S_c)$, provides a measure on the classification error for the given gene subset S_c. In this study, we use support vector machine (SVM) as the classifier since it has shown superior performance over other methods on microarray data. Further, to reduce the computational cost incurred, the leave-one-out error of SVM is estimated using the radius margin bound [13]. When two chromosomes are found having similar fitness (i.e. for a misclassification error of less than one data instance, the difference between their fitness is less than a small value of $\varepsilon = 1/n$, where n is the number of instances), the chromosome with a smaller number of selected genes is given greater chance of surviving to the next generation.

Memetic Algorithm for Gene Selection
BEGIN

1. **Initialize:** Randomly generate an initial population of feature subsets encoded with binary string.
2. **While**(*not converged or computational budget is not exhausted*)
3. Evaluate fitness of all feature subsets in the population based on $J(S_c)$.
4. Select the elite chromosome c_b to undergo local search.
5. Replace c_b with improved chromosome c_b'' using Lamarckian learning.
6. Perform evolutionary operators based on restrictive selection, crossover, and mutation.
7. **End While**

END

Fig. 1. Memetic algorithm for Gene Selection

In each MA generation, the elite chromosome, i.e., the one with the best fitness value, then undergoes a local search procedure in the spirit of Lamarckian learning [8]. Subsequently, the population of chromosome then undergoes evolutionary operators that includes linear ranking selection, and our proposed restrictive crossover and restrictive mutation [11] operators with elitism. To accelerate the evolutionary search, a constraint on the maximum number of '1' bits,m, in each chromosome is imposed.

2.1 Local Search

The local search procedure proposed is a recipe of two heuristic operators, namely *Add* and *Del*. For a given selected gene subset encoded in chromosome c, we define $\mathbf{X}$ and $\mathbf{Y}$ as the subsets of selected and excluded genes encoded in c, respectively. An *Add* operator inserts genes from $\mathbf{Y}$ into $\mathbf{X}$, while a *Del* operator removes existing genes from $\mathbf{X}$ to $\mathbf{Y}$. The important question is which gene to add or delete from a given chromosome that encodes potential gene subset. Here, we consider two possible scheme for adding or deleting genes in WFFSA and MBEGA.

1. **Filter Ranking (WFFSA)**: All features are ranked using a filter method. In this study the ReliefF [14] is considered. *Add* operator selects a feature from $\mathbf{Y}$ using the linear ranking selection method described in [15], and moves it to $\mathbf{X}$. *Del* selects a feature from $\mathbf{X}$ also using linear ranking selection and moves it to $\mathbf{Y}$. The outline for *Add* and *Del* operators are provided in Figures 2 and 3, respectively.
2. **Markov Blanket [16] (MBEGA)**: Here, both the *Add* and *Del* operators selects a feature from $\mathbf{Y}$ using also the linear ranking selection approach. However, MBEGA differs in the use of the *C-correlation* measure [17] instead of ReliefF in WFFSA for ranking of features (see Figure 2 for the details). Further for a given X_i, MBEGA proceeds to remove all other features in $\mathbf{X}$ that have been covered by X_i using the approximate Markov blanket[3] [17]. If a feature X_j has a Markov blanket given by X_i, this suggests that X_j gives no additional useful information beyond X_i on class C. Hence, X_j may be considered as redundant and could be safely removed. If there is no feature in the approximate Markov blanket of X_i, the operator then tries to delete X_i itself. The detailed procedure for *Del* operator is outlined in Figure 4.

***Add* Operator:**
BEGIN

1. Rank the features in $\mathbf{Y}$ in descending order based on ReliefF in WFFSA while the *C-correlation* measure in MBEGA.
2. Select a feature Y_i in $\mathbf{Y}$ using linear ranking selection [15] such that the higher the quality of a feature in $\mathbf{Y}$, the more likely it will be selected to move to $\mathbf{X}$.
3. Add Y_i to $\mathbf{X}$.

END

Fig. 2. *Add* operator

[3] Here we use the approximate Markov blanket [17] instead of a complete Markov blanket [16] to reduce the computational expense involved.

***Del* Operator in WFFSA:**
BEGIN

1. Rank the features in $\mathbf{X}$ in ascending order using ReliefF.
2. Select a feature X_i in $\mathbf{X}$ using linear ranking selection [15] such that the lower the quality of a feature in $\mathbf{X}$, the more likely it will be selected to move to $\mathbf{Y}$.
3. Remove X_i to $\mathbf{Y}$.

END

Fig. 3. *Del* operator in WFFSA

***Del* Operator in MBEGA:**
BEGIN

1. Rank the features in $\mathbf{X}$ in descending order based on *C-correlation* measure.
2. Select a feature X_i in $\mathbf{X}$ using linear ranking selection [15] such that the higher the *C-correlation* value of a feature in $\mathbf{X}$, the more likely it will be selected.
3. Eliminate all features in $\mathbf{X} - \{X_i\}$ which are in the approximate Markov blanket of X_i. If no feature is eliminated, remove X_i itself.

END

Fig. 4. *Del* operator in MBEGA

It is possible to quantify the computational complexity of the two local operators based on the search range L, which defines the maximum numbers of *Add* and *Del* operations. Therefore, with L possible *Add* and L possible *Del* operations, there are a total of L^2 possible combinations of *Add* and *Del* operations that may be applied on a chromosome during local learning. Since our previous study in [11] suggests $L = 4$ and the improvement first strategy gives better search performances than several others scheme considered in WFFSA, we use such a configurations in the present comparison study between WFFSA and MBEGA for gene selection. The details of the local search learning procedure used to improve only the elite chromosome of each GA search generation is outlined in Figure 5.

3 Empirical Study

In this section, we investigate the performances of WFFSA and MBEGA on both synthetic and real world microarray data. Results of basic GA (i.e., without local search) are also presented for comparison. In our empirical study, WFFSA, MBEGA and GA use the same parameter configurations with population size = 50, crossover probability = 0.6, and mutation rate = 0.5. The stopping criteria for all three algorithms are defined by a convergence to the global optimal or a

Local Search
BEGIN

1. Select the elite chromosome c_b to undergo memetic operations.
2. **For** $l = 1$ **to** L^2
3. Generate a unique random pair $\{a, d\}$ where $0 \leq a, d < L$.
4. Apply a times *Add* on c_b to generate a new chromosome c_b'.
5. Apply d times *Del* on c_b' to generate a new chromosome c_b''.
6. Calculate fitness of modified chromosome c_b'' based on $J(S_c)$.
7. **If** c_b'' is better than c_b either on fitness or number of features
8. Replace the genotype c_b with c_b'' and stop memetic operation.
9. **End If**
10. **End For**

END

Fig. 5. Local Search Procedure

maximum computational budget of 2000 fitness functional calls is reached. It is worth noting that the fitness function calls made to $J(S_c)$ in the local search are also included as part of the total fitness function calls for fair comparison to the GA. The maximum number of bit '1' in the chromosome m is set to 50.

To prevent overfitting, we consider a balanced .632+ external bootstrap [12] for estimating the prediction accuracy of a given gene subset. At each bootstrap, a training set is sampled with replacement from the original dataset, and the test data is formed by unsampled instances. Note that $J(S_c)$ uses only the training data while the prediction accuracy of a feature subset is evaluated based on the unseen test data. The external bootstrap is repeated 30 times for each dataset and the average results are reported.

3.1 Synthetic Data

We begin our study of the proposed MAs on synthetic microarray data since the true optimal gene subset is known beforehand. Here, the two-class synthetic data used is composed of 4030 genes and 80 samples with 40 samples for each class label. The centroid of these two classes are located at (-1,-1,-1) and (1,1,1). Three groups of relevant genes are generated from a multivariate normal distribution, with 10 genes in each group. All these relevant genes are generated using variance 1 and a mean vector $[\mu_1, \mu_2, \ldots, \mu_{80}]$, of $\mu_1 = \ldots = \mu_{40} = -1$ and $\mu_{41} = \ldots = \mu_{80} = 1$. The correlation between intra-group genes is 0.9, whereas the correlation between inter-group genes is 0. Hence, genes in the same group are redundant with each other and the optimal gene subset to separate the two classes consists of any 3 relevant genes from different groups. Another 4000 irrelevant genes are added to the data. Among these 4000 genes, 2000 are drawn from a normal distribution of N(0,1) and the other 2000 genes are sampled with a uniform distribution of U[-1,1].

Table 1. Feature selection by each algorithm on synthetic data

Algorithm	GA	WFFSA	MBEGA
Test Error	0.0701	0.0250	**0.0202**
#Selected Groups	2.6	**3**	**3**
#Selected Genes	34.1	35.5	**9.7**
#Selected Relevant Genes	3.2	23.2	8.2
#Selected Redundant Genes	0.6	20.2	5.2
#Selected Irrelevant Genes	30.9	12.3	1.5

#Selected Genes = #Selected Relevant Genes + #Selected Irrelevant Genes;
#Selected Relevant Genes = #Selected Groups + #Selected Redundant Genes.

The results of the feature selection by GA, WFFSA, and MBEGA are tabulated in Table 1. Both the WFFSA and MBEGA outperform GA in terms of classification accuracy, showing lower test error rates in Table 1 than the latter. MBEGA obtains the lowest test error rates among all three methods. Both WFFSA and MBEGA also select more compact feature subset than GA. They consistently select relevant genes found in all the 3 groups, while GA fails to do so and converges to subsets that contain irrelevant genes at the end of the computational budget of 2000 fitness function calls. Among all, MBEGA selects the smallest subset of relevant genes with the use of Markov blanket for local learning. Since the filter ranking procedure in WFFSA considers individual gene contributions during local learning, it cannot identifying the interactions between the genes. Therefore, unlike MBEGA, WFFSA is incapable of removing the redundant genes.

3.2 Microarray Data

In this section, we consider some real world microarray datasets having significantly large number of features (genes). In particular, 6 publicly available datasets (i.e., Colon Cancer, Central Nervous System, Leukemia, Breast Cancer, Lung Cancer, and Ovarian Cancer) in [18] are considered in the present study. In Table 2, the average test error and average number of selected genes of each feature selection algorithm (across 30 runs using the .632+ bootstraps) on the eleven datasets are reported. In gene selection, evaluation of the objective function based on SVM and local learning takes up the overwhelming bulk of the computation. To speedup the gene selection process, we considered the Grid problem solving environment developed in [19, 20]. By doing so, a sub-linear improvement in the gene selection efficiency can be achieved via parallelism since gene subsets and local search in an EA generation can be conducted independently and simultaneously across multiple compute nodes.

It is consistent with the earlier results in section 3.1 on synthetic data that both WFFSA and MBEGA obtain lower test error rates than GA (see Table 2). This suggests the local searches in both MAs have successfully helped to fine-tune

Table 2. Performance of feature selection algorithms on microarray datasets

		GA	WFFSA	MBEGA
Colon	*err*	0.1899	0.1523	**0.1434**
(2000 × 60)	$\overline{\lvert S_c \rvert}$	23.3	**23.1**	24.5
Central Nervous System	*err*	0.317	**0.277**	0.2779
(7129 × 60)	$\overline{\lvert S_c \rvert}$	24.1	22.1	**20.5**
Leukemia	*err*	0.0769	**0.0313**	0.0411
(7129 × 72)	$\overline{\lvert S_c \rvert}$	25.2	29.6	**12.8**
Breast	*err*	0.3119	0.2582	**0.1926**
(24481 × 97)	$\overline{\lvert S_c \rvert}$	22.1	27.8	**14.5**
Lung	*err*	0.0193	**0.0088**	0.0104
(12533 × 181)	$\overline{\lvert S_c \rvert}$	24.4	28.6	**14.1**
Ovarian	*err*	0.0057	0.0050	**0.0048**
(15154 × 253)	$\overline{\lvert S_c \rvert}$	23.3	18.7	**9.0**

err: test error; $\overline{\lvert S_c \rvert}$: average number of selected genes.

the GA more accurately and efficiently. As a result, smaller subset of important genes that generates improved classification accuracies are found for the datasets. Further, the results in Table 2 also suggests that WFFSA and MBEGA are competitive to each other in terms of classification accuracy. Both outperforms each other on three out of the eleven datasets considered. Nevertheless, it is worth highlighting that once again MBEGA converges to more compact gene subsets than WFFSA by eliminating the redundant genes that exists.

4 Conclusions

In this paper, two MAs feature selection algorithms WFFSA and MBEGA are investigated and compared using synthetic and real world microarray data. With the inclusion of local learning based on filter and/or Markov blanket guided *Add/Del* operators in the basic GA, both MAs have been observed to converges to solutions with more compact feature/gene subsets at improved classification performance and efficiency. The prediction accuracy of both MAs described are not significantly different when experimented on synthetic and microarray data. Nevertheless, MBEGA generally leads to a lower redundant features in the final solution than WFFSA. For diagnostic purpose in clinical practice, a smallest subset of genes would generally be preferred if good predictive performance is maintained. Therefore, MBEGA would be more suitable for gene selection.

5 Acknowledgement

This work has been funded in part under the A*STAR SERC Grant No. 052 015 0024 administered through the National Grid Office.

References

1. R. Kohavi and G. H. John, Wrapper for Feature Subset Selection, Artificial Intelligence, 97(1-2)273-324, 1997.
2. M. H. Lim, Y. Yu and S. Omatu, Efficient Genetic Algorithms using Simple Genes Exchange Local Search Policy for the Quadratic Assignment Problem, Computational Optimization and Applications, 15(3)249-268, March 2000.
3. Y. S. Ong and A.J. Keane, A domain knowledge based search advisor for design problem solving environments, Engineering Applications of Artificial Intelligence, 15(1)105-116, 2002.
4. M. H. Lim, Y. Yu and S. Omatu, Extensive Testing of A Hybrid Genetic Algorithm for Solving Quadratic Assignment Problems, Computational Optimization and Applications, 23(1)47-64, 2002.
5. J. H. Li, M. H. Lim and Q. Cao, "A QoS-Tunable Scheme for ATM Cell Scheduling Using Evolutionary Fuzzy Systems," Applied Intelligence, 23(3)207-218, 2005.
6. Y. S. Ong, P. B. Nair and K. Y. Lum, Max-Min Surrogate-Assisted Evolutionary Algorithm for Robust Aerodynamic Design, IEEE Trans. on Evolutionary Computation, 10(4), 392-404, 2006.
7. M. Wahde and Z. Szallasi, A survey of methods for classification of gene expression data using evolutionary algorithms. Expert Review of Molecular Diagnostic,6(1)101-110, 2006.
8. Y. S. Ong and A. J. Keane, Meta-Lamarckian in Memetic Algorithm, IEEE Trans. Evolutionary Computation, 8(2)99-110, 2004.
9. Y. S. Ong, M. H. Lim, N. Zhu and K. W. Wong, Classification of Adaptive Memetic Algorithms: A Comparative Study, IEEE Transactions On Systems, Man and Cybernetics - Part B, 36(1)141-52, 2006.
10. Z. Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane and K. Y. Lum, Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization, IEEE Transactions On Systems, Man and Cybernetics - Part C, 37(1)66-76, 2007.
11. Z. Zhu, Y. S. Ong and M. Dash, Wrapper-Filter Feature Selection Algorithm Using A Memetic Framework, IEEE Transactions On Systems, Man and Cybernetics - Part B, vol. 37, no. 1, Feb 2007.
12. Z. Zhu, Y. S. Ong and M. Dash, Markov Blanket-Embedded Genetic Algorithm for Gene Selection, Pattern Recognition, in communication, 2006.
13. V. Vapnik. Statistical Learning Theory, Wiley, 1998
14. M.Robnic-Sikonja and I. Kononenko, Theoretical and Empirical Analysis of ReliefF and RReliefF, Machine Learning, vol. 53, no. 1-2, pp. 23-69, 2003.
15. J. E. Baker, Adaptive Selection Methods for Genetic Algorithms, In Proc. Int'l Conf. Genetic Algorithm and Their Applications, pp. 101-111, 1985.
16. D. Koller and M. Sahami, Toward optimal feature selection, In 13th International Conference on Machine Learning, Morgan Kaufmann, Bari, Italy, 1996.
17. L. Yu and H. Liu, Efficient feature selection via analysis of relevance and redundancy, Journal of Machine Learning Research, 5,1205-1224,2004.
18. J. Li and H. Liu, Kent Ridge Biomedical Data Set Repository, Http://sdmc-lit.org.sg/GEDatasets, 2002.
19. M. Salahuddin, T. Hung, H. Soh, E. Sulaiman, Y. S. Ong, B. S. Lee, Y. Ren, Grid-based PSE for Engineering of Materials (GPEM), CCGrid, May 14-17, 2007.
20. H.K. Ng, D. Lim, Y. S. Ong, B. S. Lee, L. Freund, S. Parvez and B. Sendhoff, A Multi-Cluster Grid Enabled Evolution framework for Aerodynamic Airfoil Design Optimization, International Conference on Natural Computing, 27-29 August 2005, LNCS 3611, pp. 1112-1121.