Feature Selection using Memetic Algorithms

Cheng-San Yang
Institute of Biomedical Engineering
National Cheng-Kung University
Tainan, Taiwan
p8896117@mail.ncku.edu.tw

Li-Yeh Chuang
Dep. of Chemical Engineering
I-Shou University
Kaohsiung, Taiwan
chuang@isu.edu.tw

Yu-Jung Chen, Cheng-Hong Yang
Dep. of Computer Science and Information Engineering
National Kaohsiung University of Applied Sciences
Kaohsiung, Taiwan
1096308116, chyang@cc.kuas.edu.tw

Abstract

The feature selection process can be considered a problem of global combinatorial optimization in machine learning, which reduces the number of features, removes irrelevant, noisy and redundant data, and results in acceptable classification accuracy. In this study, we propose a combined filter method (ReliefF) and a wrapper method (memetic algorithm, MA) for classification. The goal of our method is to filter the irrelevant features and select the most important feature subsets. We used the ReliefF algorithm to calculate and update the scores of every feature for each data set, and then applied a MA for feature selection. The K-nearest neighbor (K-NN) method with leave-one-out cross-validation (LOOCV) serves as a classifier for evaluating classification accuracies. The experimental results show that the proposed method is superior to existing methods in terms of classification accuracy.

1. Introduction

Feature selection algorithms are usually categorized into two different classes based on the information to be extracted from the training data and the type of induction algorithm. They can be implemented independently from the performance of a specific learning algorithm. In order to optimize feature selection, a criterion function has to be either maximized or minimized. The effectiveness, i.e. the predictive accuracy, of the feature selection model is directly dependent on the performance of the learning algorithms.

Several methods have previously been used to perform feature selection of training and testing data, for example genetic algorithms [1], branch and bound algorithms [2], sequential search algorithms [3], mutual information [4], neural networks [5], tabu search [6], and hybrid genetic algorithms [7]. A comparative study on different feature selection methods can be found in Oh et al. [7].

As mentioned above, feature selection algorithms may be categorized into filter methods and wrapper methods [8]. The goal of the filter approach is to use the unique characteristics of each data, and then separately calculate weight values for the data. Some common filter methods are information gain [9], ReliefF [10], gain ratio [9] and chi-square [11]. Wrapper methods directly use the induction algorithms to evaluate the feature subsets. The classification algorithm subsequently calculates the feature subsets to be selected. Popular wrapper methods are genetic algorithms (GA) [1], and particle swarm optimization (PSO) [12].

In order to improve classification performance and accelerate the search to identify important feature subsets. In Zhu *et al.* [13] stated that combination of two structures will remove the irrelevant features and provide good accuracy. Since, filter methods can search through the feature space efficiently and wrapper methods can directly use the induction algorithm to evaluate the feature subsets. In summary, we propose an efficient evolutionary method (ReliefF-MA) to select relevant genes from gene expression data sets and improve classification accuracy.

The framework of ReliefF-MA is divided into two sections. The first stage uses ReliefF to calculate each gene and update the quality estimation from all features.

In the second stage, all the selected features must conform to a threshold. Subsequently, feature selection

978-0-7695-3407-7/08 $25.00 © 2008 IEEE
DOI 10.1109/ICCIT.2008.81

is once again performed, this time capitalizing on the memetic algorithms unique attributes to select the features. Afterwards, an assessment of the classification accuracy is made by K-NN classification and LOOCV (leave-one-out cross validation) [9]. After performing these two calculations, we obtained higher accuracy of classification, at the same time; the number of necessarily selected features could be lowered.

2. Related Methods

In this article, we used experimental data sets of common human diseases. ReliefF was used to select important feature subsets (genes) from all features in the gene expression data, and a memetic algorithm framework (MA) was employed for actual feature selection. Finally, the K-nearest neighbor (KNN) method with leave-one-out cross-validation (LOOCV) was used to calculate the classification accuracies.

2.1. ReliefF Algorithms

Kira and Rendell proposed a classification algorithm called Relief in 1992. It estimates the quality of features based on how well their values distinguish between instances that are near to each other [10]. Relief can deal with nominal and numerical attributes. However, it can not deal with imperfect data, and is limited to two-class problems. A modified algorithm called ReliefF, however, addresses these and other problems.

In 1994, Kononenko proposed the ReliefF algorithm. The main idea behind ReliefF is to rate features according to how well their values distinguish among instances of different classes, and how well they cluster instances of the same class. ReliefF repeatedly chooses a single instance from the data at random, and then locates the nearest instances of the same class and the nearest instances pertaining to different classes. The feature values of these instances are used to update the scores for each feature.

The pseudo-code of the ReliefF algorithm can be written as follows:

```
Algorithm ReliefF ($T, N, M$)
/*  $T$- training set, $N$-number of features   */
/*  $m$-iterate times                          */
Initialize all weights $W[A] = 0$;
For i = 1 to $m$ do begin
    Randomly select an instance R in $T$;
    Find nearest hit H and nearest miss M;
    For A = 1 to $N$
        $W[A] = W[A] - \mathrm{diff}(A, R, H)/m$
                $+ \mathrm{diff}(A, R, M)/m$;
End;
```

The ReliefF algorithm randomly selects an instance *R*, and then searches for *k* of its nearest neighbors from the same class, called nearest hits *H*. The *k* nearest neighbors from each of the different classes is called nearest misses. ReliefF updates the quality estimation *W*[*A*] for all features *A* depending on their values for *R*, nearest hits *H* and nearest misses *M*. In the update formula, we average the contribution of all the hits and all the misses.

The ReliefF algorithm is not limited to two-class problems, is very robust and can handle incomplete and noisy data.

2.2. Genetic Algorithm

Genetic Algorithms (GA) are stochastic search algorithms modeled on the process of natural selection underlying biological evolution. They can be applied to many search, optimization, and machine learning problems [14]. The basic concept behind a GA is designed to simulate evolutionary processes in natural systems, specifically those that follow the principle of survival of the fittest first laid down by Charles Darwin. As such, they represent an intelligent exploitation of a random search within a defined search space to solve a problem. A GA proceeds in an iterative manner by generating new populations of strings from old ones. Every string is the encoded binary, real, etc., version of a candidate solution. An evaluation function associates a fitness measure with every string, indicating its fitness for the problem. Standard GA apply genetic operators such selection, crossover, and mutation on an initially random population in order to compute a whole generation of new strings.

GA have been successfully applied to a variety of problems, such as scheduling problems[15], machine learning problems [19], multiple objective problems [16], feature selection problems, data mining problems [17], and traveling salesman problems [18]. Further details on the mechanisms of GAs can be found in John Holland.

Memetic Algorithm

MAs are inspired by Dawkins' notion of a meme [21]. MAs are similar to GAs, however the elements that form a chromosome are called memes, not genes. The unique aspect of the MA is that all chromosomes and offsprings are allowed to gain some experience through a local search process before being involved in the evolutionary process [22]. As such, the term MA has been used to describe a GA that heavily favors local search [23].

Similarly to GA, an initial population is randomly created by an MA. Subsequently, the local search operations move solutions towards local optima. These improvements are accumulated over all generations, resulting in a significant improvement in the overall performance [1]. Subsequently, crossover and mutation operators are applied in a fashion similar to GAs to produce offspring. These offspring are then subjected to the local search so that local optimality is always maintained.

MAs have a remarkable record of success in solving a variety of classical NP-hard optimization problems, such as graph partitioning, max independent set bin-packing, min graph coloring, quadratic assignment problems, and particularly the traveling salesman problem. In many of the above cases the authors claim that they have developed the best heuristic so far for the problem at hand [24].

Pseudo-code for a MA procedure

```
Begin;
  Generate initial population randomly;
  Do local search for each individual in the
  population;
  For $i$ = 1 to number of generations;
     Randomly selection crossover or mutation
     If crossover;
         Select two parents $i_a$ and $i_b$ at random;
         Generate offspring $i_c$ = crossover ($i_a$ and $i_b$);
         Do local search for each offspring ($i_c$);
     Else if mutation;
         Select a chromosome $i$ at random;
         Generate an offspring $i_m$ = mutation ($i$);
         Do local search for  each offspring ($i_m$);
     End if ;
         If $i_c$ or $i_m$ is better than the worst
            Chromosome
         then replace the worst chromosome
             by $i_c$ or $i_m$;
   Next $i$;
End;
```

2.4 K-Nearest Neighbor

The K-nearest neighbor (K-NN) method was first introduced by Fix and Hodges in 1951, and is one of the most popular nonparametric methods [3][10]. The purpose of the algorithm is to classify a new object based on attributes and training samples. The K-nearest neighbor method consists of a supervised learning algorithm where the result of a new query instance is classified based on the majority of the K-nearest neighbor category. The classifiers do not use any model for fitting and are only based on memory, which works based on a minimum distance from the query instance to the training samples to determine the K-nearest neighbors. Any tied results are solved by a random procedure.

The K-NN method has been successfully applied in various areas, e.g. statistical estimation, pattern recognition, artificial intelligence, categorical problems, and feature selection. The advantage of the K-NN method is that it is simple and therefore easy to implement. K-NN is not negatively affected by large training data, and is indifferent to noisy training data. In this study, the feature subset was measured by the leave-one-out cross-validation of one nearest neighbor (1-NN). Neighbors are calculated using their Euclidean distance. The 1-NN classifier does not require any user-specified parameters, and the classification results are implementation independent.

For leave-one-out cross validation, a classifier is designed using (n - 1) samples and evaluated on the one remaining sample; this is repeated n times, with different training sets of size (n - 1).

3. The Hybrid Feature Selection Algorithm

Recent studies on MAs have revealed their successes on a wide variety of real-world problems. However, the algorithm have solves binary problems.

In this study, we calculated ReliefF quality updates for six gene expression data sets by Weka [20]. Weka can calculates the ReliefF quality updates of each feature and sorts the features in accordance with their ReliefF quality updates. Higher values indicate a higher discrimination of this feature from other categories, meaning that the feature can be used to calculate classification result effectively.

After calculating the ReliefF quality updates of all features, we institute a threshold for the results. Since the results for most of the ReliefF quality updates is zero after the calculation, there are not many features which have an influence on the category. If the ReliefF quality updates of the features are higher than the threshold, we select the feature, if there are lower, the feature is not selected.

For example, gene expression data sets may contain nine genes (features) which can be represented by F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9. After the application of ReliefF, the 9 ReliefF scores are: F_1=0, F_2=0.4, F_3=0, F_4=0.9, F_5=0, F_6=1.2, F_7=0.6, F_8=0.5, F_9=0. Since most of the scores were 0, we used 0 as the threshold value. The five values that are above this threshold value (F_2, F_4, F_6, F_7, F_8) were used in the second-stage for the feature selection process.

3.1 ReliefF-MA

In this section we introduce ReliefF-MA for classification problems. Previous studies point out that wrapper methods generally outperform filter methods in terms of prediction accuracy but they are generally computationally more intensive. However, filter and wrapper methods can complement each other [13]. Hence, we combined filter and wrapper methods to not only reduce the computing time, but also improve classification accuracy. Huang *et al*. [26] indicate that by reducing the gene expression data sets to a smaller selection of relevant features classification accuracy can effectively be improved. ReliefF-MA constitutes a two-stage process.

In the first stage, relevant features are selected by ReliefF. In the second stage, the MA population is initialized randomly, with each chromosome encoding a candidate feature subset. Subsequently, a local search is performed on all or parts of the chromosomes. After that, crossover or mutation is randomly applied, and the local search is repeated. Finally, the classification accuracy is calculated by 1-NN and LOOCV.

To elucidate how to obtain an optimal accuracy, we describe the separate processes in detail below. They consist of the encoding of the chromosome and its initialization, the fitness evaluation, the local search, the crossover and mutation operation, and a replacement process.

3.1.1. Encoding chromosome and initialization. A binary string of a length equal to the total number of features is used to encode the chromosome in such a way that each bit encodes a single feature: S=F_1 F_2 F_3… F_i, i=1, 2… n. In Figure 1, the bit "1" implies that the feature is selected, while "0" implies that the feature is excluded.

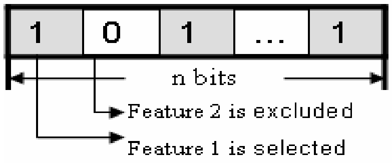


Figure 1. Encoding of the chromosome

3.1.2. Fitness evaluation. The fitness of a chromosome or selected feature subset is evaluated using the 1-NN classifier with LOOCV. For LOOCV, neighbors are calculated using their Euclidean distance. We used the classification accuracy estimated from LOOCV and the number of selected features as performance measures.

3.1.3. Local search. Local search is a method of searching a small area around a solution and adopting a better solution if one can be found. The search begins with randomly selecting a dimension of feature. The process is repeated for the new feature and the algorithm continues until a local optimum is found. In this research, a simple random local search method is employed.

Given the generations of local search $d, d = D/10$, where *D* is all dataset's features.

The local search algorithm summarized as follows:

(1) Calculating the accuracy of original chromosome (*pre_accuracy*)

(2) Randomly generate the features (f_i) from *d*, where f_i (i = 1, 2, …, d).

(3) Judged f_i this feature:

(3.1) Case 1: If this feature is excluded, then change feature from excluded to select (this chromosome is called new_ chromosome) and calculated the new_ chromosome's accuracy (*new_ accuracy*). At this time, comparing between two accuracies. If *new_ accuracy* > *pre_ accuracy* then the original chromosome will be replaced. As shown in fig2_ (a) Case 1.

(3.2) Case 2: Comparing result if *new_ accuracy* < *pre_ accuracy* then retain the original chromosome. As shown in fig2_ (b) Case 2.

(3.3) Case 3: If this feature is selected, then randomly generate next feature to implement local search. As shown in fig2_ (c) Case 3.

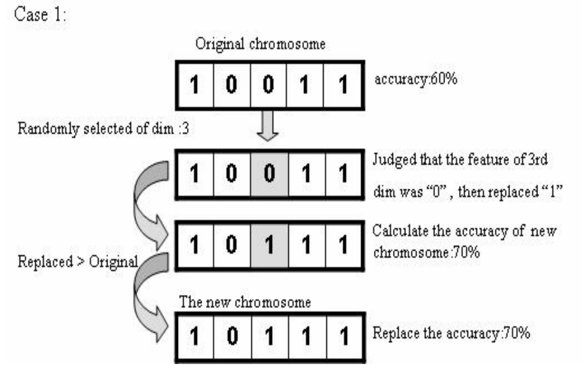


Figure 2._ (a).Case 1.

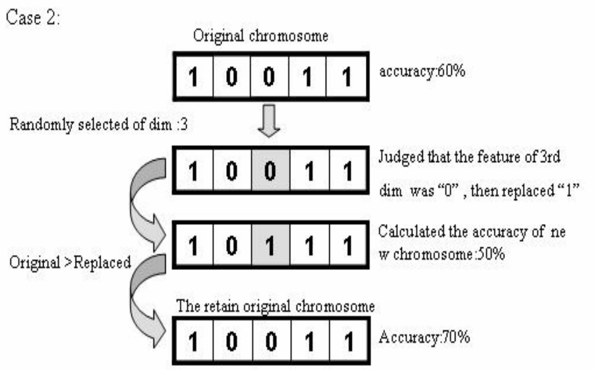

Case 2:
Original chromosome
1 0 0 1 1
accuracy:60%
Randomly selected of dim :3
1 0 0 1 1
Judged that the feature of 3rd dim was "0" , then replaced "1"
1 0 1 1 1
Calculated the accuracy of ne w chromosome:50%
Original >Replaced
The retain original chromosome
1 0 0 1 1
Accuracy:70%

Figure 2._ (b).Case 2.

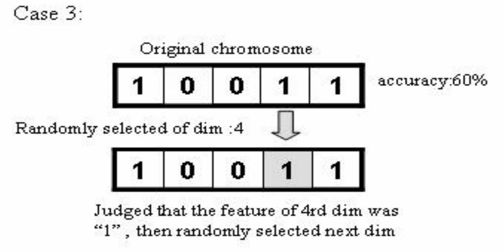

Case 3:
Original chromosome
1 0 0 1 1
accuracy:60%
Randomly selected of dim :4
1 0 0 1 1
Judged that the feature of 4rd dim was "1" , then randomly selected next dim

Figure 2._ (c).Case 3.

In this study, the chromosomes are encoded in binary form so that the features can be divided into selected and excluded ones.

Pseudo code for local search

```
/*  D- The all dataset's features */
Begin;
Randomly generate generations for local search
$d = D/10$
For a given chromosome i calculate
    fitness_KNN(i);
    Using LOOCV to calculate classification
    accuracy of original_chromosome
    (pre_ accuracy);
    For d=1 to number of variables in
    chromosome i;
        If chromosome's feature is excluded then
            change feature from excluded to select;
            Calculate fitness_KNN of the
            new_chromosome (new_ accuracy);
            If new_ accuracy > pre_ accuracy then
                replace the original chromosome;
            Else if new_ accuracy < pre_ accuracy
                then retain the original chromosome;
            End If;
        Else randomly generate next feature to
                implement local search
        End If;
    Next d;
End;
```

3.1.4. Crossover and mutation operations. The crossover operation is similar to the one in traditional GAs, where two parents $P_1$and P_2 produce an offspring by exchanging information. This can be done by randomly selecting one chromosome from the population and then arbitrarily changing some of its information. The benefit of mutation is that it randomly introduces new genetic material into the evolutionary process, and thereby perhaps avoids stagnation around local minima. A rather small mutation rate, less than 0.1, is usually used.

3.1.5. Replacement. The crossover or mutation offspring of two parents is compared to them. If the offspring is superior to both parents, it replaces the similar parent; if it is in between the two parents, it replaces the inferior parent; otherwise, the most inferior chromosome in the population is replaced.

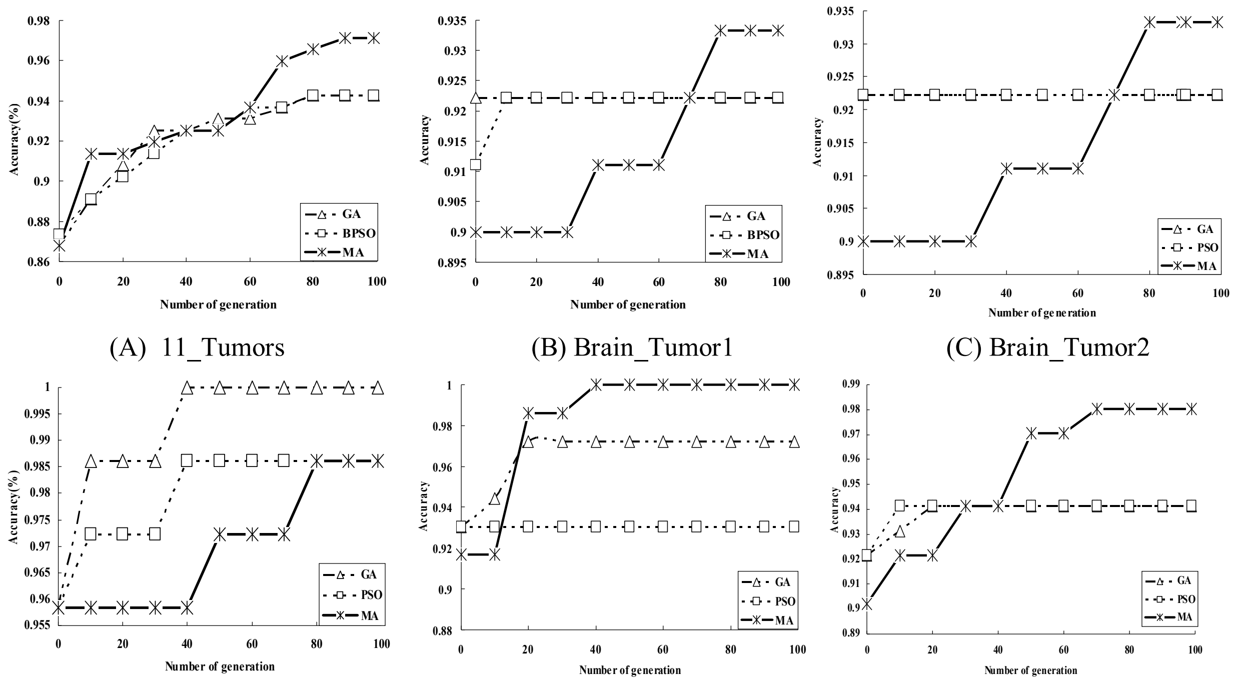
4. Experimentation

4.1 Environment

In order to evaluate the performance of the system, experiments were performed using six data sets from http://www.gems-system.org. These gene expression data were obtained by the oligonucleotide technique. The six human cancer-related data sets used to test the algorithms are summarized in Table 1, which includes the data set name, the number of categories, and samples.

4.2 The parameter setting

The threshold of ReliefF is 0 for most of the data sets, the population size is 30, the number of generations is 100, and the crossover rate is set to 1.0. And the mutation rate is set to 0.1.



(A) 11_Tumors

(B) Brain_Tumor1

(C) Brain_Tumor2

(D) Leukemia1

(E) Leukemia2

(F) Prostate_Tumor

Figure 3. Comparison of the classification accuracies by GA, BPSO and MA

4.3 Experimental results

For the six gene expression data sets, the number of genes was selected by using ReliefF as shown in Table 2. The Leukemia1 data set is only remained 15.92% features of the original data after satisfying the threshold value of ReliefF. However The Brain_Tumor2 data set is remained 43.07% features of the original data after satisfying the threshold value of ReliefF.

The second stage use an MA aimed at the feature subsets previously selected by ReliefF to perform another feature selection and calculate the classification accuracy.

In Fig. 3, classification results for ReliefF-GA, ReliefF-BPSO (Binary PSO) and ReliefF-MA are compared for the six data sets used.

In Fig. 3(D). The effect on Leukemia1 dataset is inferior to ReliefF-GA, because the Leukemia1 data set only has 848 features after ReliefF selected. This data set is not only the lowest in all datasets from Table1 but also we aimed at the 1 / 10 of the features after ReliefF for local search. Hence the accuracy was not better. Maybe relation with the generations of local search only few features therefore caused the accuracy was inferior to ReliefF-GA.

Table 3 show that the average classification accuracy of the proposing method improved 1.88% in ReliefF-GA and 3.14% in ReliefF-BPSO.

We also compared some Non-SVM and MC-SVM methods from Statnikov *et al.*[25] to our method. In amongst the Non-SVM methods, the best was K-NN, and the best amongst the MC-SVM methods was OVR; however ReliefF-MA outperforms both K-NN and MC-SVM.

4.4 Discussion

Feature selection improves calculation efficiency and classification accuracy in classification problems with multiple features, since not all features necessarily influence classification accuracy [6]. Selecting appropriate features improves the predictive accuracy; however selecting inappropriate features compromises the predictive accuracy. Hence, employing appropriate feature selection to select optimal features for a category results in higher classification accuracy.

According to the experimental results presented, ReliefF-MA outperformed the other evolutionary algorithms, probably because neither ReliefF-GA nor

ReliefF-PSO uses a local search. Contrary to GA-based methods, the MA employs a local search.

The performance of this local search operation allows the MA to reach a local optimum or improve the solution. A local search was performed on each individual of the population to improve its experience and thus achieve a population of local optimum solutions. Then, crossover and mutation operators applied to the offsprings. The offsprings were yet again subjected to the local search so that local optimality was always maintained [27].

5. Conclusion

In this study, we propose a novel hybrid filter and wrapper feature selection method called ReliefF-MA. The experimental results presented show that this method conducts more efficient searches than other methods, and is capable of producing high classification accuracy with a small number of features. Most importantly, the proposed method was superior to the other algorithms in terms of accuracy, particularly for large-sized problems. ReliefF-MA effectively reduced computing time and improved operational efficiency, as well as classification accuracy.

In the future, it would be worthwhile developing other schemed, such as gene rearrangement for chromosome encoding and applying our proposed method to solve other problems.

6. References

[1] M.L. Raymer, W.F. Punch, E.D. Goodman, L.A. Kuhn, and A.K. Jain, “Dimensionality Reduction Using Genetic Algorithms”, IEEE Trans. Evolutionary Computation, vol. 4, no. 2, 2000, pp 164-171.

[2] B. Yu, and B. Yuan, “A More Efficient Branch and Bound Algorithm for Feature Selection”, Pattern Recognition, vol. 26, no.6, 1993, pp. 883-889.

[3] P. Pudil, J. Novovicova, and J. Kittler, “Floating Search Methods in Feature Selection”, Pattern Recognition Letters, vol.15, 1994, pp. 1119-1125.

[4] B. Roberto, “Using mutual information for selecting features in supervised neural net learning”, IEEE Transactions on Neural Networks, vol. 5, no. 4, 1994, pp.537-550.

[5] F. Brill, D. Brown, and W. Martin, “Fast Genetic Selection of Features for Neural Network Classifiers”, IEEE Transactions on Neural Networks, vol. 3, no. 2, 1992, pp.324-328.

[6] H. Zhang, and G. Sun, “Feature selection using tabu search method”, Pattern Recognition, no.35, 2002, pp.701-711.

[7] I.-S. Oh, J.-S. Lee, and B.-R. Moon, “Hybrid Genetic Algorithms for Feature Selection”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol.26, no.11, 2004, pp.1424-1436.

[8] M. Dash, and H. Liu, “Feature selection for classification”, Int. J. Intell. Data Anal., vol. 1, no. 3, 1997, pp. 131–156.

[9] J.R. Quinlan, “Induction of decision trees”, Machine Learning, no. 1, 1986, pp.81-106.

[10] M. Robnic- ˇ Sikonja, and I. Kononenko, “Theoretical and empirical analysis of ReliefF and RReliefF”, Mach. Learn, vol.53, no. 1/2, 2003, pp. 23–69.

[11] H. Liu, and R. Setiono, “Chi2: Feature selection and discretization of numeric attributes”, in Proc. 7th IEEE Int. Conf. Tools With Artif. Intell, 1995, pp.388-391.

[12] J. Kennedy, and R. Eberhart, “Particle swarm optimization”, Procee dings of the IEEE international conference on neural networks, 1995, pp.1942–1948.

[13] 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: CYBERNETICS, vol. 37, no.1, 2007, pp.70-76.

[14] D.E. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-Wesley, Reading, MA, 1989.

[15] E.S. Hou, N. Ansari, and H. Ren, “A Genetic Algorithm for Multiprocessor Scheduling”, IEEE Transactions on Parallel and Distributed Systems, vol. 5, no. 2, 1994, pp. 113-120.

[16] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II”, IEEE Trans. Evol. Comput., vol. 6, 2002, pp. 182-197.

[17] S. Kim, and B.-T. Zhang, “Evolutionary learning of web-document structure for information retrieval”, in Proc. of the 2001 Congress on Evolutionary Computation, vol. 2, 2001, pp.1253-1260.

[18] W. Pullan, *Adapting the Genetic Algorithm to the Traveling Salesman Problem,* IEEE Congress on Evolutionary Computation, 2003.

[19] H. Vafaie, and K. D. Jong, Genetic Algorithms as a Tool for Feature Selection in Machine Learning, Proceeding of the 4th International Conference on Tools with Artificial Intelligence, 1992, pp.200-204.

[20] E. Frank, M. Hall, L. Trigg, G. Holmes, and I.H. Witten, “Data mining in bioinformatics using Weka”, Bioinformatics, vol. 20, 2004, pp.2479-81.

[21] R. Dawkins, “The selfish gene”, Oxford: Oxford University Press, 1976.

[22] P. Merz, and B. Freisleben, “A genetic local search approach to the quadratic assignment problem”, In: Ba¨ck CT, editor. Proceedings of the 7th international conference on genetic algorithms. San Diego, CA: Morgan Kaufmann, 1997, pp. 465–472.

[23] P. Moscato, MG. Norman, “A memetic approach for the traveling salesman problem—implementation of a computational ecology for combinatorial optimization on

message-passing systems", International conference on parallel computing and transporter application, 1992, pp. 177–186.

[24] P.Zou, Z. Zhou, G. Chen, and X. Yao, "A Novel Memetic Algorithm with Random Multi-local-search: A case study of TSP", IEEE. vol.2, 2004 p.2335– 2340.

[25] A. Statnikov, C.F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy, "A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis", Bioinformatics, vol. 21, 2005, pp.631–643.

[26] H.L. Huang, C.C. Lee, and S.Y. Ho, "Selecting a minimal number of relevant genes from microarray data to design accurate tissue classifiers" BioSystems, 2006.

[27] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," Advanced Engineering Informatics, vol. 19, 2005, pp. 43-53.

Table 1.Cancer-related human gene expression data sets

Data set	Diagnostic task	Number of		
		Samples	Genes	Classes
11_Tumors	Eleven various human Tumor types	174	12533	11
Brain_Tumor1	Five human brain tumor types	90	5920	5
Brain_Tumor2	Four malignant glioma types	50	10367	4
Leukemia1	Acute myelogenous leukemia (AML), acute lympboblastic leukemia (ALL) B-cell, and ALL T-cell	72	5327	3
Leukemia2	AML, ALL, and mixed-lineage leukemia (MLL)	72	11225	3
Prostate_Tumor	Prostate tumor and normal tissue	102	10509	2

Table 2.Selected Gene Numbers by Filter Method

Data set	Original Genes(A)	Genes after ReliefF (B)	Rate(B/A)
11_Tumors	12533	3181	25.38 %
Brain_Tumor1	5920	1612	27.23 %
Brain_Tumor2	10367	4465	43.07 %
Leukemia1	5327	848	15.92 %
Leukemia2	11225	4596	40.94 %
Prostate_Tumor	10509	2016	19.18 %

Table 3.Classification Accuracy of Non-SVM, MC-SVM, ReliefF-GA, ReliefF-PSO and ReliefF-MA

Methods Data sets	Non-SVM			MC-SVM					ReliefF	ReliefF-GA	ReliefF-PSO	ReliefF-MA
	KNN	NN	PNN	OVR	OVO	DAG SVM	WW	CS	KNN	KNN	KNN	KNN
11_Tumors	78.51	54.14	77.21	94.68	90.36	90.36	94.68	95.30	85.06	94.25	94.25	**97.13**
Brain_Tumor1	87.94	84.72	79.61	91.67	90.56	90.56	90.56	90.56	88.89	92.22	92.22	**93.33**
Brain_Tumor2	68.67	60.33	62.83	77.00	77.83	77.83	73.33	72.83	82.00	90.00	88.00	**92.00**
Leukemia1	83.57	76.61	85.00	97.50	97.32	96.07	97.50	97.50	93.06	**100.00**	98.61	98.61
Leukemia2	87.14	91.03	83.21	97.32	95.89	95.89	95.89	95.89	88.89	97.22	93.06	**100.00**
Prostate_Tumor	51.09	33.25	39.22	71.14	71.14	71.14	71.14	71.14	91.18	94.12	94.12	**98.04**
Average	76.15	66.68	71.18	88.22	87.18	86.98	87.18	87.20	88.18	94.64	93.38	**96.52**

Legends: (1) Non-SVM: Traditional classification method. (2) MC-SVM: Multi-class support vector machines. (3) KNN: K-Nearest Neighbors.(4) NN: Back propagation Neural Networks. (5) PNN: Probabilistic Neural Networks. (6) OVR: One-Versus-Rest. (7) OVO: One-Versus-One. (8) DAG: DAGSVM. (9) WW: Method by Weston and Watkins. (10) CS: Method by Crammer and Singer. (11)ReliefF: Correlation-based Feature Selection. (12) ReliefF-GA: Correlation-based Feature Selection-Genetic Algorithm (13) ReliefF-BPSO: Correlation-based Feature Selection-Particle Swarm Optimization (14) ReliefF-MA: Correlation-based Feature Selection-Memetic Algorithm.