Original Article
International Journal of Fuzzy Logic and Intelligent Systems
Vol. 13, No. 1, March 2013, pp. 73-82
http://dx.doi.org/10.5391/IJFIS.2013.13.1.73

ISSN(Print) 1598-2645
ISSN(Online) 2093-744X

Hybrid Feature Selection Using Genetic Algorithm and Information Theory

Jae Hoon Cho[1], Dae-Jong Lee[2], Jin-Il Park[1], and Myung-Geun Chun[2]
[1]Smart Logistics Technology Institute, Hankyong National University, Anseong, Korea
[2]Department of Electrical & Computer Engineering, Chungbuk National University, Cheongju, Korea

Abstract

In pattern classification, feature selection is an important factor in the performance of classifiers. In particular, when classifying a large number of features or variables, the accuracy and computational time of the classifier can be improved by using the relevant feature subset to remove the irrelevant, redundant, or noisy data. The proposed method consists of two parts: a wrapper part with an improved genetic algorithm(GA) using a new reproduction method and a filter part using mutual information. We also considered feature selection methods based on mutual information(MI) to improve computational complexity. Experimental results show that this method can achieve better performance in pattern recognition problems than other conventional solutions.

Keywords: Pattern classification, Feature selection, Mutual information, Computational complexity

1. Introduction

Feature selection algorithms can be categorized based on *subset generation* and *subset evaluation* [1]. *Subset generation* is a search procedure that selects candidate feature subsets, based on certain search strategies, such as complete search, sequential search, and random search. *Subset evaluation* is a set of evaluation criteria used to evaluate a selected feature subset. The criteria can be categorized into two groups based on their dependency on inductive algorithms: independent criteria and dependent criteria. Some independent criteria include distance measures, information measures, dependency measures, and consistency measures [2-5]. A dependent criterion requires a predetermined inductive algorithm in feature selection. Based on the performance of the inductive algorithm applied on the selected subset, it determines which features are selected. Under evaluation criteria, algorithms are categorized into filter, wrapper, and hybrid. Filter methods are independent of the inductive algorithm and evaluate the performance of the feature subset by using the intrinsic characteristic of the data. In the filter methods, the optimal features subset is selected in one pass by evaluating some predefined criteria. Therefore, filter methods have the ability to quickly compute very high-dimensional datasets; however, they also have the worst classification performance, because they ignore the effect of the selected feature subset on the performance of the inductive algorithm. The wrapper methods utilize the error rate of the inductive algorithm as the evaluation function. They search for the best subset of features in all available feature subsets. Wrapper methods

Received: Feb. 18. 2013
Revised : Mar. 13. 2013
Accepted: Mar. 15. 2013

Correspondence to: Myung-Geun Chun (mgchun@chungbuk.ac.)

are generally known to perform better than filter methods.

Information theory has been applied to feature selection problems in recent years. Battiti [6] proposed a feature selection method called mutual information feature selection (MIFS). Kwak and Choi [7] investigated the limitation of MIFS using a simple example and proposed an algorithm that can overcome the limitation and improve performance. The main advantage of mutual information methods is the robustness of the noise and data transformation. Despite these advantages, the drawback of feature selection methods based on mutual information is the slow computational speed due to the computation of a high-dimensional covariance matrix. In pattern recognition, feature selection methods have been applied to various classifiers. Mao [8] proposed a feature selection method based on pruning and support vector machine (SVM), and Hsu et al. [9] proposed a method called artificial neural net input gain measurement approximation (ANNIGMA) based on weights of neural networks. Pal and Chintalapudi [10] proposed an advanced online feature selection method to select the relevant features during the learning time of neural networks.

On the other hand, the techniques of evolutionary computation, such as genetic algorithm and genetic programming, have been applied to feature selection to find the optimal features subset. Siedlecki and Sklansky [11] used GA-based branch-and-bound technique. Pal et al. [12] proposed a new genetic operator called self-crossover for feature selection. In the genetic algorithm (GA)-based feature selection techniques, each chromosomal gene represents a feature and each individual represents a feature subset. If the ith gene of the chromosome equals 1, then the ith feature is selected as one of the features used to evaluate a fitness function; if the chromosome is 0, then the corresponding feature is not selected. Kudo and Sklansky [13] compared a GA-based feature selection with many conventional feature selection methods, and they showed that GA-based feature selection performs better than others for high-dimensional datasets.

In this paper, we propose a feature selection method using both information theory and genetic algorithm. We also considered the performance of each mutual information (MI)-based feature selection method to choose the best MI-based method to combine with genetic algorithm. The proposed method consists of two parts: the filter part and the wrapper part. In the filter part, we evaluated the significance of each feature using mutual information and then removed features with low significance. In the wrapper part, we used genetic algorithm to select the optimal feature subsets with smaller sizes and higher classification performance, which is the goal of the proposed method. In order to estimate the performance of the proposed method, we applied our method on University of California-Irvine (UCI) machine-learning data sets [14]. Experimental results showed that our method is effective and efficient in finding small subsets of the significant features for reliable classification.

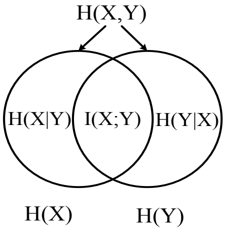


Figure 1. Relation between entropy and mutual information.

2. Mutual Information-Based Feature Selection

2.1 Entropy and Mutual Information

Entropy and mutual information are introduced in Shannon's information theory to measure the information of random variables [15]. Basically, mutual information is a special case of a more general quantity called relative entropy, which is a measure of the distance between two probability distributions. The entropy is a measure of uncertainty of random variables. More specifically, if a discrete random variable$X$has λ alphabets with its probability density function denoted as $p(x) = \Pr\{X = x\}$, $x \in \lambda$, then the entropy of X can be defined as

$$H(X) = -\sum_{x\in\lambda} p(x)\log p(x). \tag{1}$$

The joint entropy of two discrete random variables X and Y is defined as follows:

$$H(X,Y) = -\sum_{x\in\lambda}\sum_{y\in\delta} p(x,y)\log p(x,y) \tag{2}$$

where $p(x,y)$denotes the joint probability density function of X and Y. When some variables are known and the others are not, the remaining uncertainty can be described by the conditional entropy, which is defined as

$$H(Y|X) = -\sum_{x\in\lambda}\sum_{y\in\delta} p(x,y)\log p(y|x) \tag{3}$$

The common information of two random variables X and Y is defined as the mutual information between them:

$$I(X;Y) = \sum_{x \in \lambda} \sum_{y \in \delta} p(x,y) \log \frac{p(x,y)}{p(x) \cdot p(y)} \tag{4}$$

A large amount of mutual information between two random variables means that the two variables are closely related; otherwise, if the mutual information is zero, then the two variables are totally unrelated or independent of each other. The relation between the mutual information and the entropy can be described in (5), which is also illustrated in Figure 1.

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) = H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y), \\ I(X;Y) &= I(Y;X), \\ I(X;X) &= H(X) \end{aligned} \tag{5}$$

In feature selection problems, the mutual information between two variables feature *F* and class *C* is defined in terms of their probabilistic density functions $p(f)$,$p(c)$, and$p(f,c)$:

$$I(F;C) = \sum_{f \in \lambda} \sum_{c \in \delta} p(f,c) \log \frac{p(f,c)}{p(f) \cdot p(c)} \tag{6}$$

If the mutual information $I(F;C)$ between feature *F* and class *C* is large, it means that feature *F* contains much information about class *C*. If $I(F;C)$ is small, then feature *F* has little effect on output class *C*. Therefore, in feature selection problems, the optimal feature subset can be determined by selecting the features with higher mutual information.

2.2 Mutual Information-Based Feature Selection

2.21 Feature Selection Problem with Mutual Information

Feature selection is a process that selects a subset from the complete set of original features. It selects the feature subset that can best improve the performance of a classifier or an inductive algorithm. In the process of selecting features, the number of features is reduced by excluding irrelevant or redundant features from the ones extracted from the raw data. This concept is formalized as selecting the most relevant *k* features from a set of *n* features. Battiti [6] named this concept a "feature reduction" problem.

Let the FRn-k problem be defined as follows:

Given an initial set of *n* features, find the subset with *k*<*n* features, such that the subset is "maximally informative" about the class.

In information theory, the mutual information between two random variables measures the amount of information commonly found in these variables. The problem of selecting input features that contain the relevant information about the output can be solved by computing the mutual information between input features and output classes. If the mutual information between input features and output classes could be obtained accurately, the FRn-k problem could be reformulated as follows:

Given an initial set *F* with *n* features and a *C* set of all output classes, find the subset $S \subset F$with k features such that the subset minimizes$H(C|S)$ and maximizes the mutual information $I(C;S)$.

To solve this FRn-k problem, we can use three strategies: complete search, random search, and sequential search. Complete search guarantees to find the optimal feature subset according to an evaluation criterion. This strategy evaluates all possible subsets to guarantee completeness; however, it is almost impossible due to the large number of combinations. Random search starts with a randomly selected subset and proceeds in two different ways. One is to follow a sequential search, which injects randomness into the sequential approaches. The other is to generate the next subset in a completely random manner. The use of randomness helps to escape local optima in the search space. Sequential search gives up completeness and therefore risks losing optimal subsets. Many variations of the greedy algorithm to the sequential search are available, including sequential forward selection and sequential backward elimination. All these approaches add or remove features one at a time. Algorithms with sequential search are simple to implement and quickly produce results, because the order of the search space is usually $O(N^2)$ or less.

Typically, the mutual information-based feature selection is performed by sequential forward selection. This method starts with an empty set of selected features, and then we add the best available input feature to the selected feature set one by one until the size of the set reaches k. This ideal sequential forward selection algorithm using mutual information is implemented as follows:

1. (Initialization) Set $F \leftarrow$ "initial set of *n* features," $S \leftarrow$ "empty set."

2. (Computation of the MI with the output class) If $\forall f_i \in F$, compute $I(C;F)$.

3. (Selection of the first feature) Select a feature that maximizes $I(C;F)$, and set $F \leftarrow F \backslash \{f_i\},\ S \leftarrow \{f_i\}$

4. (Greedy sequential forward selection) Repeat until the desired number of selected features is reached.

5. (Computation of the joint MI between variables) If $\forall f_i \in F$, compute$I(C; f_{i,}S)$.

6. (Selection of the next feature) Choose the feature $f_i \in F$ that maximizes $I(C; f_i, S)$ and set $F \leftarrow F\backslash\{f_i\},\ S \leftarrow \{f_i\}$.Output is the set S containing the selected features.

where $C$is a class, f_i is the ith feature, and $S$is a feature subset.

2.22 Battiti's Mutual Information-Based Feature Selection

In the ideal sequential forward selection, we must estimate the joint mutual information between variables $I(C; f_i, S)$ and know the probabilistic density functions of variables to compute $I(C; f_i, S)$. However, it is difficult to compute probabilistic density functions of high-dimension data; therefore, we use a histogram of data.

In selecting k features, if the output classes are composed of K_c classes and we divide the jth input feature space into P_j partitions to get the histogram, we must have $K_c \times \prod_{j=1}^{k} P_j$ cells to compute $I(C; f_i, S)$.

Because of this requirement, implementing the ideal sequential forward selection algorithm is practically impossible. To overcome this practical problem, Battiti [6] used only $I(C; f_i)$ and $I(f_i, f_s)$, instead of calculating$I(C; f_i, S)$. The mutual information $I(C; f_i)$indicates the relative importance of the input feature f_i, which was estimated based on the mutual information between the input feature f_i and the output class$C$. $I(f_i, f_s)$ indicates the redundancy between the input feature f_i and the already-selected featuresf_s. Battiti's algorithm, also known as MIFS, is essentially the same as the ideal greedy sequential forward selection algorithm, except for Step 4, which was replaced in MIFS as follows [6]:

4) (Greedy sequential forward selection) Repeat until the desired number of selected features is reached.

a) (Computation of the MI between variables) For all couples of variables (f_i, f_s) with $f_i \in F,\ f_s \in S$, compute $I(f_i, f_s)$, if it is not yet available.

b) (Selection of the next feature) choose the feature $f_i \in F$ that maximizes $I(C; f_i) - \beta \sum_{f_s \in S} I(f_i, f_s)$, and set $F \leftarrow F\backslash\{f_i\},\ S \leftarrow \{f_i\}$.

$$I(C; f_i|S) = I(C; f_i) - \beta \sum_{f_s \in S} I(f_i; f_s) \qquad (7)$$

where β regulates the influence of the redundancy of input featuref_i. If β= 0, the redundancy among features is not taken into consideration, and the algorithm selects features in the order of relative importance estimated by the mutual information between input features and output classes.

3. Genetic Algorithm-Based Feature Selection

Genetic algorithm is one of the best-known techniques for solving optimization problems. It is also a search method based on a random population. First, genetic algorithm randomly encodes the initial population, which is a set of created individuals. Each individual can be represented as bit strings that can be constructed using all possible permutations in a potential solution space. At each step, the new population is determined by processing the chromosome of the old population in order to obtain the best fitness in a given situation. This sequence continues until a termination criterion is reached. The chromosome manipulation is performed using one of three genetic operators: crossover, mutation, and reproduction. The selection step determines which individuals will participate in the reproduction phase. Reproduction itself allows the exchange of already existing genes, whereas mutation introduces new genetic material, where the substitution defines the individuals for the next population. This process efficiently provides optimal or near-optimal solutions.

In the genetic algorithm-based feature selection, the size of chromosome *n* represents the total number of features *N*, and a gene represents a feature with values "1" and "0" meaning selected and removed, respectively. Therefore, we can define genetic algorithm-based feature selection as finding the optimal feature subset with the smallest number of genes set to "1" and with a higher classification performance.

A generational procedure GA is shown below:

```
steady_state_GA()
{
    initialize population P;
    repeat {
        for($i$=1 to $|P|$)          {
            select two parents p1 and p2 from P;
            offspring =crossover(p1, p2);
            mutation(offspring);
        }
        replace P with offspring$_1$ ,. . . ,offspring$_{|P|}$ ;
    } until (stopping condition);
}
```

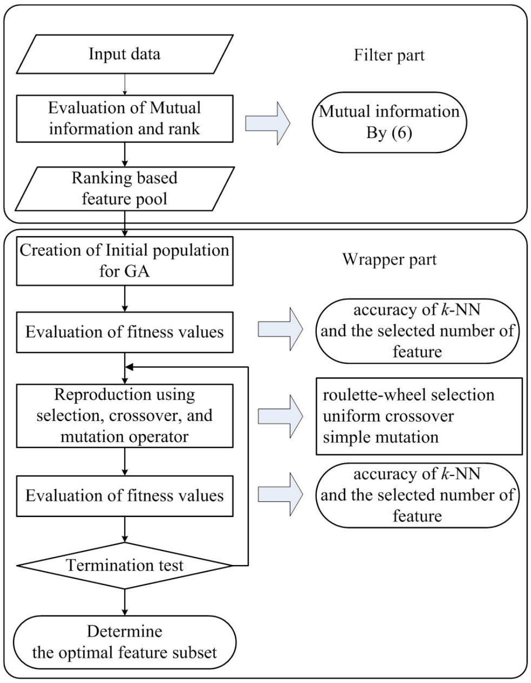

Input data
Filter part
Evaluation of Mutual information and rank
Mutual information By (6)
Ranking based feature pool
Creation of Initial population for GA
Wrapper part
Evaluation of fitness values
accuracy of k-NN and the selected number of feature
Reproduction using selection, crossover, and mutation operator
roulette-wheel selection
uniform crossover
simple mutation
Evaluation of fitness values
accuracy of k-NN and the selected number of feature
Termination test
Determine the optimal feature subset

Figure 2. Flowchart of the proposed method. GA, genetic algorithm; k-NN, k-nearest neighbor algorithm.

4. Proposed Method by Mutual Information and Genetic Algorithm

The proposed method can be divided into two parts. The filter part is a preprocessing step to remove irrelevant, redundant, and noise features according to ranking by mutual information. In the wrapper part, the optimal feature subset is selected from the preprocessed features using genetic algorithm with the fitness function based on the number of features and on the accuracy of classification. Figure 2 shows the structure of the proposed method, described as follows.

4.1 Filter Part by Mutual Information

[Step 1] *Evaluate mutual information.* We determine the mutual information between each feature *F* and each class *C* by (6).

[Step 2] *Select the top-ranked features.* Each feature is ranked using the evaluated mutual information. Then, we select the top-ranked features with higher mutual information to use as candidate individuals for the genetic algorithm. To determine the number of top-ranked features, we categorized the features into three types: full-top-ranked, half-top-ranked, and *U*-top-ranked. Full and half-top-ranked can be used on data with a small feature size, and *U*-top-ranked can be used on data with a high feature size, where *U* is the selection rate determined by the user.

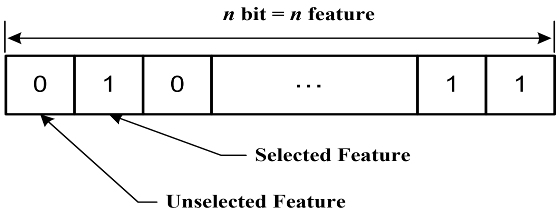


Figure 3. n-dimension binary chromosome.

4.2 Wrapper Part by Genetic Algorithm

[Step 3] *Set initial parameters of GA and generate initial population.* Set the initial parameters of GA, such as population size, probability of crossover and mutation, the number of generation, weight of fitness, and initial population rate. We also generate the initial population from the features selected in Step 2. In genetic algorithm-based feature selection methods, each chromosome is represented by an n-bit binary for an n–dimensional feature space $\{b_1, b_2, \cdots, b_n\}$. If the ith feature is present in the feature subset represented by the chromosome, then $b_i = 1$; otherwise, $b_i = 0$.

An example of a simple procedure is as follows:

```
Initial population:
    for ($i$=1 to $|P|$)
        for (each gene $j$ in $i$th chromosome)
            if (random_number ( )<$\alpha$)
                $j$th gene of $i$th chromosome = 1;
            else
                $j$th gene of $i$th chromosome = 0;
        end
    end
end
```

where $|P|$ is the population size, *random_number* is a function that generates a random floating number within [0,1], and $\alpha$is the expected number of selected features. Figure 3 shows the structure of an n-dimension binary chromosome.

[Step 4] *Evaluate fitness function of initial generation.* Evaluate the fitness values of all individuals in the population using a classifier to evaluate each chromosome (the selected feature subset) based on the classification accuracy and the dimension of the feature subset. Tan et al. [14] proposed the following fitness function in order to optimize two objectives: maximize the classification accuracy of the feature subset and minimize the size of the feature subset.

$$fitness(z) = \lambda \cdot acc(z) + (1 - \lambda) \cdot \left(\frac{1}{feats(z)} \right) \quad (8)$$

where z is a feature vector representing a selected feature subset, and λ is a weight value between 0 and 1. The function is composed of two parts. The first part is the weighted classification accuracy$acc(z)$from a classifier and the second part is the weighted size $feats(z)$of the feature subset represented by$z$. In order to get a balance between accuracy and dimensionality reduction, the following fitness function is proposed:

$$fitness(z) = \lambda \cdot acc(z) - (1 - \lambda) \cdot \frac{feats(z)}{total\_feat} \quad (9)$$

where *total_feat* is the maximum number of features for the problem. In (8) and (9), in order to obtain the classifier with the best accuracy and the smallest size, λ is set to the rate 0.5.

[Step 5] *Perform selection, crossover, and mutation step.* To produce the new feature subsets, these operators are carried out by the GA: selection operator, crossover operator, and mutation operator. The selection operator selects new feature subsets based on the fitness value of each feature, and then the crossover and mutation operators create the next generation feature subsets.

[Step 6] *Evaluate the fitness function of the next generation.*

[Step 7] *Perform termination test.* If a predefined generation is satisfied, then stop the algorithm. Otherwise, go to Step 5. In this study, the termination condition required only one generation.

5. Experiments and Discussions

Ten-fold cross-validation procedure is commonly used to evaluate the performance of k-nearest neighbor algorithm(k-NN) with 1-nearest neighbor. In the 10-fold cross-validation, the selected feature subsets are partitioned into ten. To test the MLP, one feature subset is retained as the validation data, and the remaining nine feature subsets are used as training data. The cross-validation process is repeated 10 times, and the 10 sets of results can be averaged to produce a single estimate. In the filter part, we used *0.7*-top-ranked feature. Table 1 shows the parameters for the genetic algorithm and Table 2 shows the UCI datasets used in this experiment. The first two rows are artificial datasets and the rest are real-world datasets.

Table 1. Parameters for the genetic algorithm

Parameter	Value
Population size	20
Probability of crossover	0.7
Probability of mutation	0.1
Generation	50
Weight of fitness (λ)	0.5
Initial population rate(α)	0.5
Selection method	Roulette wheel
Crossover	Two point crossover

In this experiment, we used the fitness values evaluated by using (9) and the three genetic operators: roulette-wheel selection, uniform crossover, and simple mutation.

Figures 4 and 5 show the fitness values of GA in each generation and the number of selected features in each generation. We can see that the optimal feature subset was effectively found by the proposed method.

In Table 3, all the results of different methods are obtained from [9]. The NN column lists the results with no feature subset selection. The second column shows the results of the standard wrapper-backward elimination method. The third column shows the results of the ANNIGMA-wrapper method proposed by Hsu et al. The fourth column represents the results of GA, and the final column shows the results of the proposed method. For each error and each number of selected features, we include the average and the standard deviation. As shown in Table 3, the proposed method shows better performance than the other methods for most datasets with small features. More specifically, the error rate is 4.2% when using the eight dominant features chosen by the proposed method, whereas the error rate is 11.4% for NN without feature selection. From this result, one can see that the proposed method makes it possible to dramatically decrease the error.

6. Conclusion

The feature selection methods can be divided into two groups, filter method and wrapper method, based on their dependence and independence on the inductive algorithm. Filter methods

Table 2. UCI datasets and classifiers used in this experiment

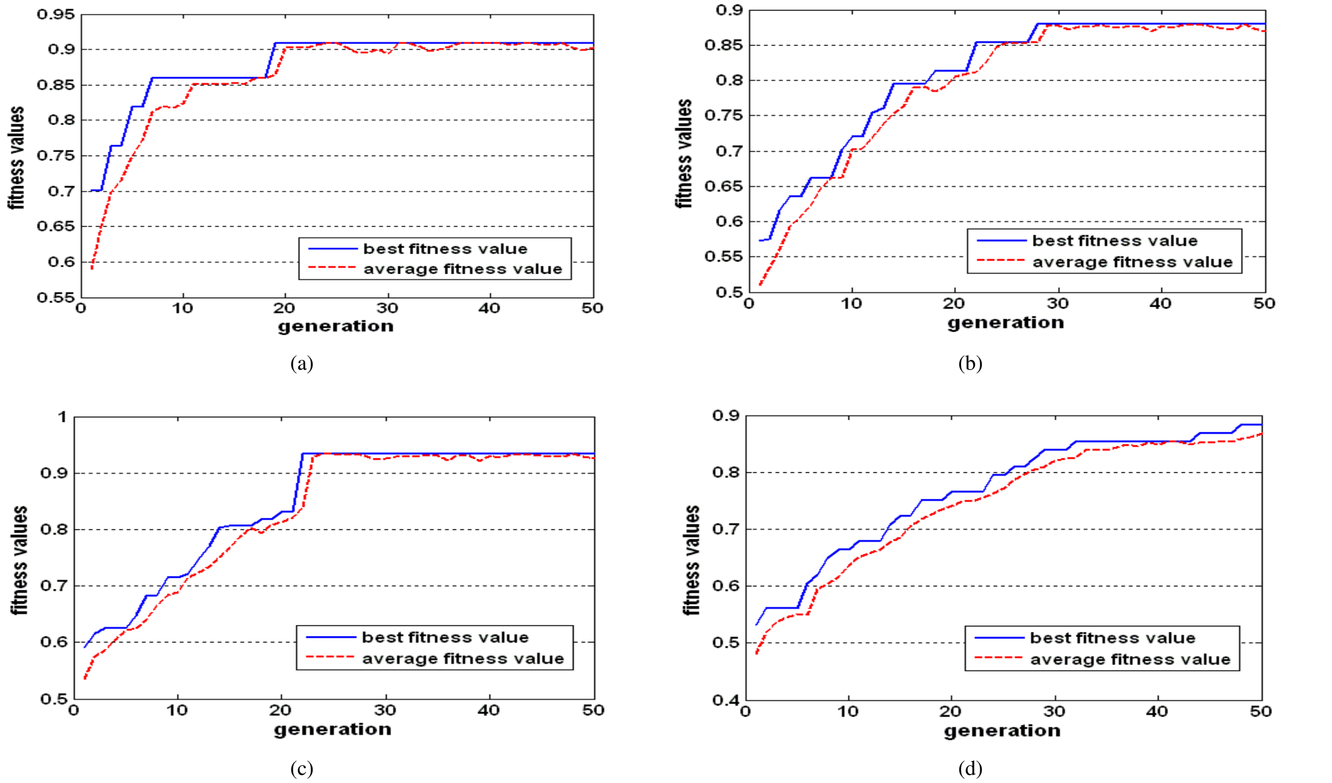
Data set	No. of feature	No. of classes	No. of samples	Classifier	Cross-validation test
Monk3b	15	2	554	1-NN	10-fold
Breast cancer	9	2	699	1-NN	10-fold
Credit	9	2	690	1-NN	10-fold
Ionosphere	34	2	351	1-NN	10-fold

NN, neural networks; UCI, University of California-Irvine.

Table 3. Comparison results between the proposed and other methods

Dataset	NN[9]		Wrapper method[9]		ANNIGMA-Wrapper[9]		SGA		Proposed method	
	Feature	Error (%)	Feature	Error (%)	Feature	Error (%)	Feature	Error (%)	Feature	Error (%)
Monk3b	15	2.8±0.0	4.4±1.1	2.8±0.0	**2.2±0.4**	2.8±0.0	3.5±0.8	3.2±1.0	3.2±1.0	**2.6±0.1**
Cancer	9	4.1±4.7	7.2±1.2	3.6±1.1	5.8±1.3	3.5±1.2	5.2±0.7	2.8±0.7	**2.1±0.4**	**1.4±0.3**
Credit	15	14.1±1.7	13.4±1.0	14.4±0.8	6.7±2.5	12.0±0.8	7.3±1.6	15.4±1.5	**5.2±1.5**	**11.0±0.9**
Ionosphe	34	11.4±3.9	9.0±2.5	9.8±1.3	9.0±2.5	9.8±1.3	12.4±2.7	8.2±3.1	**8.2±2.3**	**7.2±0.7**

ANNIGMA, artificial neural net input gain measurement approximation; NN, neural networks; SGA, simple genetic algorithm.



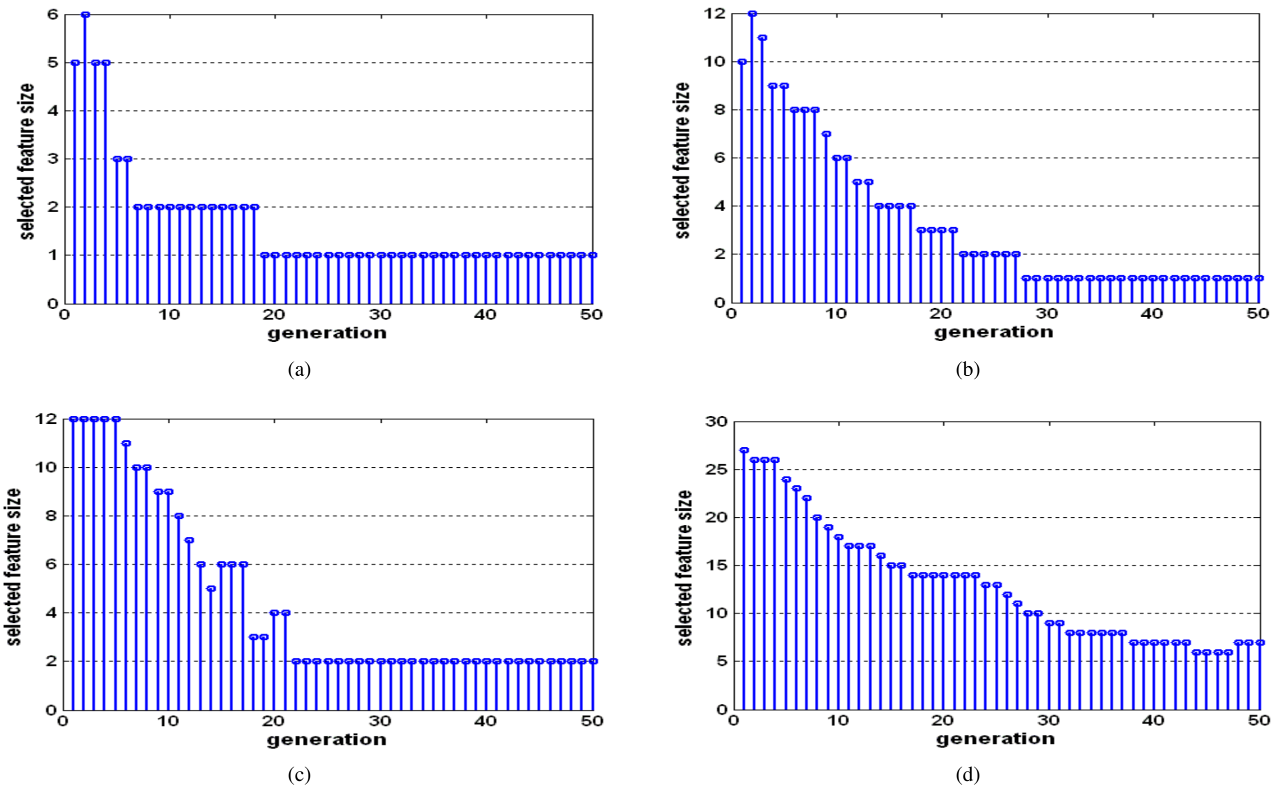
(a)

(b)

(c)

(d)

Figure 4. Fitness values of the genetic algorithm. (a) Breast cancer Wisconsin data, (b) credit data, (c) monk3b data, (d) ionosphere data.



(a)

(b)

(c)

(d)

Figure 5. The number of selected feature in each generation. (a) Breast cancer Wisconsin data, (b) credit data, (c) monk3b data, (d) ionosphere data.

have fast computational ability because the optimal feature subset is selected in one pass by evaluating some predefined criteria. However, they have the worst classification performance, because they ignore the effect of the selected feature subset on the performance of the inductive algorithm. The wrapper methods have higher performance than the filter methods, whereas they have high computational cost.

In order to overcome the drawbacks of both filter methods and wrapper methods, we propose a feature selection method using both information theory and genetic algorithm. The proposed method was applied to UCI datasets and some gene expression datasets. For the various experimental datasets, the proposed method had better generalization performance than previous ones. More specifically, the error rate is 4.2% when using the eight dominant features chosen by the proposed method, whereas the error rate is 11.4% for NN without feature selection. From these results, one can see that the proposed method makes it possible to dramatically decrease the error without increasing the computational time.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0024037).

References

[1] M. Dash and H. Liu, “Feature selection for classification,” *Intelligent Data Analysis*, vol. 1, no. 3, pp. 131-156, 1997. http://dx.doi.org/10.1016/S1088-467X(97)00008-5

[2] M. Rais, J. Barrera, and D. C. Martins Jr, “U-curve: a branch-and-bound optimization algorithm for U-shaped cost functions on Boolean lattices applied to the feature selection problem,” *Pattern Recognition*, vol. 43, no. 3, pp.

557-568, Mar. 2010. http://dx.doi.org/10.1016/j.patcog.2009.08.018

[3] S. Foithong, O. Pinngern, and B. Attachoo, “Feature subset selection wrapper based on mutual information and rough sets,” *Expert Systems with Applications*, vol. 39, no. 1, pp. 574-584, Jan. 2012. http://dx.doi.org/10.1016/j.eswa.2011.07.048

[4] N. R. Pal and M. Malpani, ”Redundancy-constrained feature selection with radial basis function networks,” in *Proceedings of 2012 IEEE International Joint Conference on Neural Networks (IJCNN)*, Brisbane, 2012, pp. 1-8. http://dx.doi.org/10.1109/IJCNN.2012.6252638

[5] T. Zhang, ”On the consistency of feature selection using greedy least squares regression,” *Journal of Machine Learning Research*, vol. 10, no. Mar, pp. 555-568, Mar. 2009.

[6] R. Battiti, “Using mutual information for selecting features in supervised neural net learning,” *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537-550, Jul. 1994. http://dx.doi.org/10.1109/72.298224

[7] N. Kwak and C. H. Choi, “Input feature selection for classification problems,” *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 143-159, Jan. 2002. http://dx.doi.org/10.1109/72.977291

[8] K. Z. Mao, “Feature subset selection for support vector machines through discriminative function pruning analysis,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 1, pp. 60-67, Feb. 2004. http://dx.doi.org/10.1109/TSMCB.2002.805808

[9] C. N. Hsu, H. J. Huang, and S. Dietrich, “The ANNIGMA-wrapper approach to fast feature selection for neural nets,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 32, no. 2, Apr. 2002. http://dx.doi.org/10.1109/3477.990877

[10] N. R. Pal and K. Chintalapudi, “A connectionist system for feature selection,” *Neural, Parallel and Scientific Computations*, vol. 5, no. 3, pp. 359-381, Sep. 1997.

[11] W. Siedlecki and J. Sklansky, “A note on genetic algorithms for large-scale feature selection,” *Pattern Recognition Letters*, vol. 10, no. 5, pp. 335-347, Nov. 1989. http://dx.doi.org/10.1016/0167-8655(89)90037-8

[12] N. R. Pal, S. Nandi, and M. K. Kundu, “Self-crossover: a new genetic operator and its application to feature selection,” *International Journal of Systems Science*, vol. 29, no. 2, pp. 207-212, May. 1998. http://dx.doi.org/10.1080/00207729808929513

[13] M. Kudo and J. Sklansky, “Comparison of algorithms that select features for pattern classifiers,” *Pattern Recognition*, vol. 33, no. 1, pp. 25-41, Jan. 2000. http://dx.doi.org/10.1016/S0031-3203(99)00041-2

[14] F. Tan, X. Z. Fu, Y. Q. Zhang, and A. G. Bourgeois, “Improving feature subset selection using a genetic algorithm for microarray gene expression data,” in *Proceedings of 2006 IEEE Congress on Evolutionary Computation*, Vancouver, 2006, pp. 2529-2534. http://dx.doi.org/10.1109/CEC.2006.1688623

[15] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, Urbana: University of Illinois Press, 1949.

Jae Hoon Cho received his B.S. and M.S. degrees in Control Engineering from Hanbat National University in 2002 and 2004, respectively, and his Ph.D. in Control Engineering from Chungbuk National University in 2011. He is currently Research Professor at the Smart Logistics Technology Institute, Hankyong National University, Anseong, Gyeonggi-do, Korea. His research interests include fuzzy intelligent optimization algorithms and smart grid technology.
E-mail: jhcho@hknu.ac.kr

Jin Il Park received his B.S. and M.S. degrees in Control Engineering from Hanbat National University in 2001 and 2003, respectively, and his Ph.D. in Control Engineering from Chungbuk National University in 2009. He is currently Research Professor at the Smart Logistics Technology Institute, Hankyong National University, Anseong, Gyeonggi-do, Korea. His research interests include fuzzy intelligent optimization algorithms and pattern classification.
E-mail: moralskr@gmail.com

Dae Jong Lee received his B.S., M.S., and Ph.D. degrees in Electrical Engineering from Chungbuk National University in 1995, 1997, and 2002, respectively. His current research interests include pattern classification, image processing, and biometrics.

Email: bigbell@chungbuk.ac.kr

Myung-Geun Chun received his B.S. degree in Electronics Engineering from Pusan National University in 1987 and his M.S. and Ph.D. degrees in Intelligent Systems from the Korea Advance Institute of Science and Technology (KAIST), Daejeon, Korea, in 1989 and 1993, respectively. Prior to joining Chungbuk National University, he worked at Samsung Electronics as a senior researcher. He is currently Professor at the Department of Electronic Engineering, Chungbuk National University, Cheongju, Korea. His research interests include the design and development of intelligent systems as well as the design of biometric systems with privacy protective capabilities.

E-mail: mgchun@chungbuk.ac.kr