ELSEVIER

Pattern Recognition Letters 23 (2002) 1495–1503

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm

Shinn-Ying Ho [*], Chia-Cheng Liu, Soundy Liu

Department of Information Engineering, Feng Chia University, 100 Wenhwa Road, Seatwen, Taichung 407, Taiwan

Received 13 August 2001; received in revised form 22 October 2001

Abstract

The goal of designing an optimal nearest neighbor classifier is to maximize the classification accuracy while minimizing the sizes of both the reference and feature sets. A novel intelligent genetic algorithm (IGA) superior to conventional GAs in solving large parameter optimization problems is used to effectively achieve this goal. It is shown empirically that the IGA-designed classifier outperforms existing GA-based and non-GA-based classifiers in terms of classification accuracy and total number of parameters of the reduced sets. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Feature selection; Intelligent genetic algorithm; Minimum reference set; Nearest neighbor classifier

1. Introduction

The nearest neighbor (1-nn) classifier is commonly used due to its simplicity and effectiveness (e.g., Kuncheva and Bezdek, 1998; Kuncheva and Jain, 1999). According to 1-nn rule, an input is assigned to the class of its nearest neighbor from a stored labeled reference set. The goal of designing an optimal 1-nn classifier is to maximize the classification accuracy while minimizing the sizes of both the reference and feature sets. It has been recognized that the editing of the reference set and feature selection must be simultaneously determined when designing the compact 1-nn classifier with high classification power. Genetic algorithms (GAs) have been shown to be effective for exploring NP-hard or complex non-linear search spaces as efficient optimizers relative to computer-intensive exhaustive search (e.g., Goldberg, 1989). Kuncheva and Jain (1999) proposed a genetic algorithm (KGA) for simultaneous editing and feature selection to design 1-nn classifiers. KGA was found to be an expedient solution compared to editing followed by feature selection, feature selection followed by editing, and the individual results from feature selection and editing. The investigated problem of designing an optimal 1-nn classifier is described as follows (e.g., Kuncheva and Jain, 1999):

Let $X = \{X_1, \ldots, X_n\}$ be the set of features describing objects as n-dimensional vectors $x = [x_1, \ldots, x_n]^T$ in R^n and let $Z = \{z_1, \ldots, z_N\}$, $z_j \in R^n$, be the data

[*] Corresponding author. Tel.: +886-4-24517250; fax: +886-4-24516101.
E-mail address: syho@fcu.edu.tw (S.-Y. Ho).

set. Associated with each z_j, $j = 1,\ldots,N$, is a class label from the set $C = \{1,\ldots,c\}$. The criteria of *editing* and *feature selection* are to find subsets $S_1 \subseteq Z$ and $S_2 \subseteq X$ such that the classification accuracy is maximal and the number N_{p} of parameters of the reduced set is minimal, where $N_{\mathrm{p}} = \mathrm{card}(S_1)\mathrm{card}(S_2)$ and $\mathrm{card}(\cdot)$ denotes cardinality. Define $P_{\text{1-nn}}(S_1,S_2)$ as the classification accuracy of the 1-nn classifier using S_1 and S_2 as a real-valued function

$$P_{\text{1-nn}} : P(Z) \times P(X) : \to [0,1],$$

where $P(Z)$ is the power set of Z and $P(X)$ is the power set of X. The optimization problem is how to search for S_1 and S_2 in the combined space such that $P_{\text{1-nn}}$ is maximal and N_{p} is minimal. Essentially, this is a bi-criteria combinatorial optimization problem having an NP-hard search space of $C(N+n, \mathrm{card}(S_1) + \mathrm{card}(S_2))$ instances (e.g., Horowitz et al., 1997), i.e., the number of ways of choosing $\mathrm{card}(S_1) + \mathrm{card}(S_2)$ out of $N+n$ parameters (0/1 decision variables), and two incommensurable and often competing objectives: maximum of $P_{\text{1-nn}}$ and minimum of N_{p}. Generally, the parameter number $N+n$ is large. Despite having been successfully used to solve many optimization problems, conventional GAs cannot efficiently solve large parameter optimization problems (LPOPs). In this paper, a novel intelligent genetic algorithm (IGA) (e.g., Ho et al., 1999) superior to conventional GAs in solving LPOPs is used to solve the problem of designing an optimal 1-nn classifier. It will be shown empirically that the IGA-designed classifier outperforms existing GA-based and non-GA-based classifiers in terms of both $P_{\text{1-nn}}$ and N_{p}.

IGA uses an intelligent crossover (IC) based on orthogonal experimental design (OED). Sections 2 and 3 briefly introduce OED and IC. Section 4 presents the design of optimal 1-nn classifiers using IGA. Section 5 reports the experimental results and Section 6 brings the conclusion.

2. Orthogonal experimental design

Experiments are carried out by researchers or engineers in all fields to compare the effects of several conditions or to discover something new. If an experiment is to be performed efficiently, a scientific approach to planning it must be considered. The statistic design of experiments is the process of planning experiments so that appropriate data will be collected, the minimum number of experiments will be performed to acquire the necessary technical information, and suitable statistic factor analysis methods will be used to analyze the collected data.

An efficient way to study the effect of several factors simultaneously is to use OED based on orthogonal arrays (OAs) and factor analysis. OAs are used to provide the treatment settings at which one conducts the "all-factors-at-one" statistical experiments (e.g., Mori, 1995). Many design experiments use OAs for determining which combinations of factor levels or treatments to use for each experimental run and for analyzing the data. The OA-based experiments can provide near-optimal quality characteristics for a specific objective. Furthermore, there is a large saving in the experimental effort.

OA is a matrix of numbers arranged in rows and columns where each row represents the levels of factors in each run and each column represents a specific factor. In the context of experimental matrices, orthogonal means statistically independent. The properties of OA are: (1) For the factor in any column, every level occurs the same number of times. (2) For the two factors in any two columns, every combination of two levels occurs the same number of times. (3) If any two columns of an OA are swapped or some columns are ignored, the resulting array is still an OA. (4) The selected combinations used in OA experiments are uniformly distributed over the whole space of all possible combinations.

The major reason for using OAs rather than other possible arrangements in robust designs is that OAs allow the individual factor (also known as main) effects to be rapidly estimated, without the fear of distortion of results by the effects of other factors. Factor analysis can evaluate the effects of factors on the evaluation function, rank the most effective factors, and determine the best level for each factor such that the evaluation is optimized.

OED is certainly not mere observation of an uncontrolled and random process. Rather, they are well-planned and controlled experiments in which certain factors are systematically set and modified. OED specifies the procedure of drawing a representative sample of experiments with the intention of reaching a sound decision. Therefore, OED, which makes uses of the efficient experimental design and factor analysis, is regarded as a systematic reasoning method.

3. Intelligent crossover

In the conventional crossover operations of GA, two parents generate two children with a combination of their chromosomes using a *randomly* selected cut point. The merit of IC is that the systematic reasoning ability of OED is incorporated in the crossover operator to economically estimate the contribution of individual genes to a fitness function, and consequently intelligently pick up the better genes to form the chromosomes of children. The high performance of IC arises from that IC replaces the generate-and-test search for children using a random combination of chromosomes with a systematic reasoning search method using an intelligent combination of selecting better individual genes. Theoretically analysis and experimental studies for illustrating the superiority of IC with the use of OA and factor analysis can be found in (e.g., Ho et al., 1999). A concise example of illustrating the use of OA and factor analysis can be found in (e.g., Ho et al., 1999; Ho and Chen, 2001).

3.1. OA and factor analysis

A two-level OA used in IC is described as follows. Let there be γ factors with two levels for each factor. The total number of experiments is 2^γ for the popular "one-factor-at-a-time" study. The columns of two factors are orthogonal when the four pairs, $(1,1)$, $(1,2)$, $(2,1)$, and $(2,2)$, occur equally frequently over all experiments. Generally, levels 1 and 2 of a factor represent selected genes from parents 1 and 2, respectively. To establish an OA of γ factors with two levels, we obtain an integer $\omega = 2^{\lceil \log_2(\gamma+1) \rceil}$, build an orthogonal array $L_\omega(2^{\omega-1})$ with ω rows and $(\omega-1)$ columns, use the first γ columns, and ignore the other $(\omega-\gamma-1)$ columns. Table 1 illustrates an example of OA $L_8(2^7)$. The algorithm of constructing OAs can be found in (e.g., Leung and Wang, 2001). OED can reduce the number of experiments for factor analysis. The number of OA experiments required to analyze all individual factors is only ω or $O(\gamma)$.

After proper tabulation of experimental results, the summarized data are analyzed to determine the relative effects of various factors. Let y_t denote the positive function evaluation value of experiment t, $t = 1, 2, \ldots, \omega$. Let $Y_t = y_t(1/y_t)$ if the objective function is to be maximized (minimized). Define the main effect of factor j with level k as S_{jk}:

$$S_{jk} = \sum_{t=1}^{\omega} Y_t^2 F_t, \tag{1}$$

where $F_t = 1$ if the level of factor j of experiment t is k; otherwise, $F_t = 0$. Notably, the main effect

Table 1
Orthogonal array $L_8(2^7)$

Experiment number	Factor							Function evaluation value
	1	2	3	4	5	6	7	
1	1	1	1	1	1	1	1	y_1
2	1	1	1	2	2	2	2	y_2
3	1	2	2	1	1	2	2	y_3
4	1	2	2	2	2	1	1	y_4
5	2	1	2	1	2	1	2	y_5
6	2	1	2	2	1	2	1	y_6
7	2	2	1	1	2	2	1	y_7
8	2	2	1	2	1	1	2	y_8

reveals the individual effect of a factor. The most effective factor j has the largest main effect difference (MED) $|S_{j1} - S_{j2}|$. If $S_{j1} > S_{j2}$, the level 1 of factor j makes a better contribution to the optimization function than level 2 does. Otherwise, level 2 is better.

After the better level of each factor is determined, an intelligent combination consisting of factors with better levels can be efficiently derived. OED is effective for development design of efficient search for the intelligent combination of factor levels, which can yield a best or near-best function evaluation value among all values of 2^{γ} combinations.

3.2. IC operator

A candidate solution consisting of 0/1 decision variables to an optimization problem is encoded into a chromosome using binary codes. One gene (variable) of a chromosome is regarded as a factor of OED. If values of a specific gene in two parent chromosomes are the same, i.e., all equal to value 0/1, this gene is not necessary to participate the IC operation. Two parents breed two children using IC at a time. Let the number of participated genes in a parent chromosome be γ. How to use OA and factor analysis to achieve IC is described as the following steps:

Step 1: Select the first γ columns of OA $L_{\omega}(2^{\omega-1})$ where $\omega = 2^{\lceil \log_2(\gamma+1) \rceil}$.

Step 2: Let levels 1 and 2 of factor j represent the jth variable of a chromosome coming from the parents 1 and 2, respectively.

Step 3: Evaluate the fitness function values y_t for experiment t where $t = 1, 2, \ldots, \omega$.

Step 4: Compute the main effect S_{jk} where $j = 1, 2, \ldots, \gamma$ and $k = 1, 2$.

Step 5: Determine the better level for each variable. Select level 1 for the jth factor if $S_{j1} > S_{j2}$. Otherwise, select level 2.

Step 6: The chromosome of the first child is formed from the intelligent combination of the better genes from the derived corresponding parents.

Step 7: Rank the most effective factors from ranks 1 to γ. The factor with large MED has higher rank.

Step 8: The chromosome of the second child is formed similarly as the first child except that the variable with the lowest rank adopts the other level.

Step 9: The best and the second best individuals among the two parents and two generated children based on fitness performance are used as the final children of IC for the elitist strategy.

It takes about $\omega = 2^{\lceil \log_2(\gamma+1) \rceil}$ fitness evaluations for performing an IC operation. The value γ for each IC operation would gradually decrease when evolution proceeds with a decreasing number of non-determinate variables. This behavior can helpfully cope with the large parameter optimization problem of simultaneous editing and feature selection.

4. IGA-designed 1-nn classifier

4.1. Chromosome encoding and fitness function

The feasible solution S corresponding to the reduced reference and feature sets is encoded using a binary string consisting of $N + n$ bits and representing two sets: $S_1 \subseteq Z$ and $S_2 \subseteq X$. The first N bits are used for S_1, and the last n bits for S_2. The i-th bit has value 1 when the respective element of Z/X is included in S_1/S_2, and 0 otherwise. The search space consists of $2^{(N+n)}$ points.

The fitness function $F(S)$ using a counting estimator (e.g., Raudys and Jain, 1990) and a penalty term as a soft constraint on the total cardinality of S_1 and S_2 is defined as follows:

$$\text{maximize } F(S) = \sum_{j=1}^{m} h_s^{\text{CE}}(v_j) - \alpha(\text{card}(S_1) + \text{card}(S_2)), \qquad (2)$$

where α is a weight. The classification accuracy is measured on a validation set $V = \{v_1, \ldots, v_m\}$, different from the training set Z. If v_j is correctly classified on S by 1-nn rule, $h_s^{\text{CE}}(v_j) = 1$, and 0 otherwise.

4.2. IGA

Conventional GA (e.g., KGA) which is called simple genetic algorithm (SGA) consists of five primary operations: initialization, evaluation, selection, crossover, and mutation. IGA can be simply the same as SGA with elitist strategy in initialization, evaluation, selection, and mutation operations. IGA with IC in the proposed approach is described as follows:

Step 1: Initialization. Randomly generate an initial population with N_{pop} individuals.

Step 2: Evaluation. Evaluate the fitness function values of all individuals.

Step 3: Selection. Use the rank selection that replaces the worst $P_{\mathrm{s}}N_{\mathrm{pop}}$ individuals with the best $P_{\mathrm{s}}N_{\mathrm{pop}}$ individuals to form a new population, where P_{s} is a selection probability.

Step 4: Crossover. Randomly select $P_{\mathrm{c}}N_{\mathrm{pop}}$ individuals to perform IC, where P_{c} is a crossover probability.

Step 5: Mutation. Apply the conventional bit-inverse mutation operator to the population using a mutation probability P_{m}. The best individual is retained without being subject to the mutation operation.

Step 6: Termination test. If a prespecified termination condition is met, end the algorithm. Otherwise, go to Step 2.

Although different control parameter specifications of GA may result in different performances of the designed classifiers, IGA uses the same control parameters as KGA herein to illustrate its simplicity and efficiency of designing 1-nn classifiers. The control parameters of KGA are as follows: $N_{\mathrm{pop}} = 10$, $P_{\mathrm{c}} = 1.0$, and $P_{\mathrm{m}} = 0.1$; the number of 1's in the initial population is around 80% of all bit values generated; and the elitist selection strategy is used. The control parameters of IGA are as follows: $N_{\mathrm{pop}} = 10$, $P_{\mathrm{s}} = 0$, $P_{\mathrm{c}} = 1.0$, and $P_{\mathrm{m}} = 0.1$. Since a larger number of fitness evaluations than that of KGA is needed for one crossover operation, the terminal conditions of both IGA and KGA use the same number of fitness evaluations.

The presented IGA is an efficient general-purpose algorithm for solving large parameter optimization problems. That is, IGA is not specially designed for solving the investigated design problem of nearest neighbor classifiers. The effectiveness of IGA with the used control parameters is discussed as follows. The population size is very small ($N_{\mathrm{pop}} = 10$) and the best individual in the population can surely participate the crossover operation ($P_{\mathrm{c}} = 1.0$). In addition, a bit representing one decision variable is treated as a factor, which is the smallest evaluation unit to be inherited. Therefore, conventional selection step can be disabled ($P_{\mathrm{s}} = 0$) that still results in high performance. Since two offspring chromosomes of IC may differ by just one bit, the diversity of population would be decreased if $P_{\mathrm{s}} \neq 0$. On the other hand, a high mutation rate ($P_{\mathrm{m}} = 0.1$) would increase the diversity of population. Note that the best individual is retained without being subject to the mutation operation. OA specifies a small number of combinations that are uniformly distributed over the whole space of all possible combinations, which equals the solution space if a bit is treated as a factor. Therefore, factor analysis of OED can economically explore the entire solution space and consequently IGA can obtain a global optimal or near-optimal solution.

5. Experiments

Two experiments are used to demonstrate the effectiveness of IGA in designing an optimal 1-nn classifier. In Experiment 1, the same two data sets used in KGA are tested to verify the superiority of IGA. In Experiment 2, various generated data sets are applied to KGA and IGA for demonstrating the capability of solving the problem of designing 1-nn classifiers with high-dimensional patterns with overlapping. Two data sets are described as follows:

(1) The SATIMAGE data from ELENA database (anonymous ftp at ftp.dice.ucl.ac.be, directory pub/neural-nets/ELENA/databases): 36 features, 6 classes, 6435 data points with 3 different training-validation-test splits of the same size: 100/200/6135.

(2) A generated data set (e.g., Jain and Zongker, 1997) with some extensions: J features, 2 classes, 10 different samplings with training-validation-test sizes as 100/200/1000. The classes were equiprobable, distributed as:

$$p_1(x) \sim N(\mu_1, I) \qquad p_2(x) \sim N(\mu_2, I),$$

where

$$\mu_1 = -\mu_2 = \left[\frac{1}{\sqrt{1}}, \frac{1}{\sqrt{2}}, \ldots, \frac{1}{\sqrt{J}}\right]^{\mathrm{T}},$$
$$I = 1, \ldots, 4, \quad J \in \{20, 40, 60, 80, 100\}.$$

5.1. Experiment 1

In this experiment, the SATIMAGE, and the generated data set with $I = 1$ and $J = 20$ are used. The average convergences of KGA and IGA using $\alpha = 0.04$ are shown in Fig. 1. The experimental results using IGA are reported in Tables 2 and 3, compared with those of the Pareto-optimal sets without considering IGA. For the SATIMAGE data, the Pareto-optimal set is $\{\mathrm{W + H + SFS, KGA, W + SFS, SFS}\}$ and for the generated data, $\{\mathrm{W + H + SFS, KGA, SFS + W}\}$, where H (*Hart's condensed nearest neighbor rule*, Hart, 1968) and W (*Wilson's method*, Wilson, 1972) are two basic methods for editing, and SFS is the *Sequential Forward Selection* method (Stearns, 1976). The sign + means a combination of two methods. The scatterplots of IGA, KGA, and some non-GA-based methods are shown in Fig. 2. All the results of non-IGA methods are gleaned from the litera-

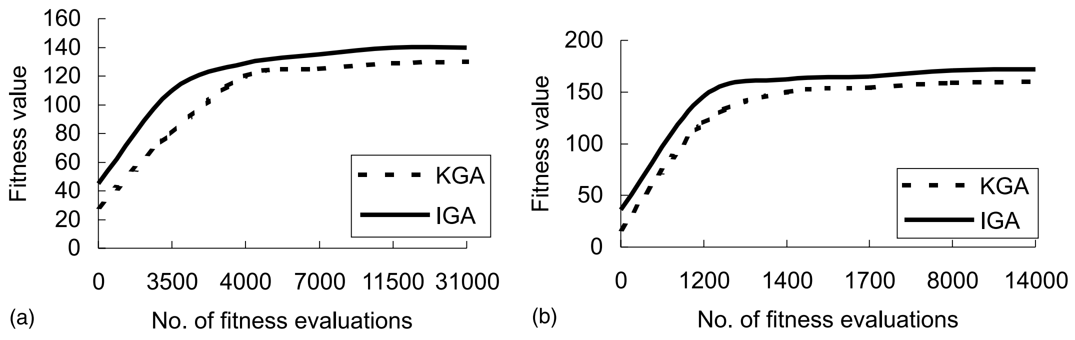


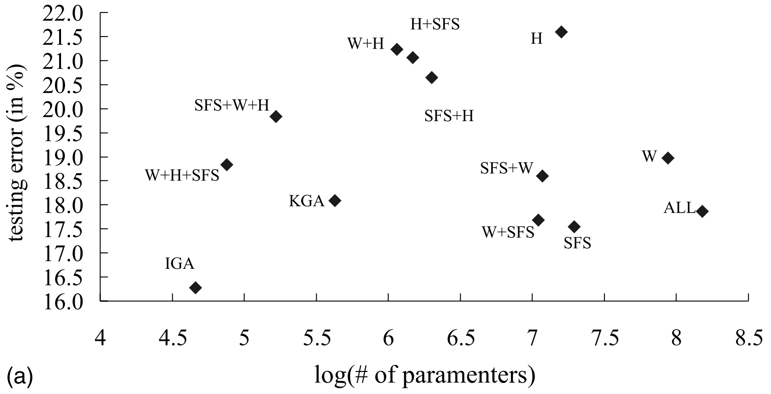
Fig. 1. The comparisons of convergences of KGA and IGA: (a) SATIMAGE data, (b) generated data.

Table 2
Average results with the SATIMAGE data (three experiments)

Method	Testing error (%)	Card(S_1)	Card(S_1)	N_p	Pareto-optimality
All	17.87	100	36	3600	Dominated
SFS	17.54	100	14.67	1467	Dominated
W + SFS	17.68	78.33	14.67	1149	Dominated
W + H + SFS	18.83	12	11	132	Dominated
KGA	18.09	27	10.33	279	Dominated
IGA	16.28	17.66	6.0	106	Pareto-optimal

Table 3
Average results with the generated data (10 experiments)

Method	Testing error (%)	Card(S_1)	Card(S_2)	N_p	Pareto-optimality
All	11.94	100	20	2000	Dominated
W + H + SFS	10.95	20	9.7	194	Dominated
SFS + W	8.41	91.1	13.9	1266	Dominated
KGA	9.17	26.28	8.64	227	Dominated
IGA	7.3	11.66	9.0	105	Pareto-optimal



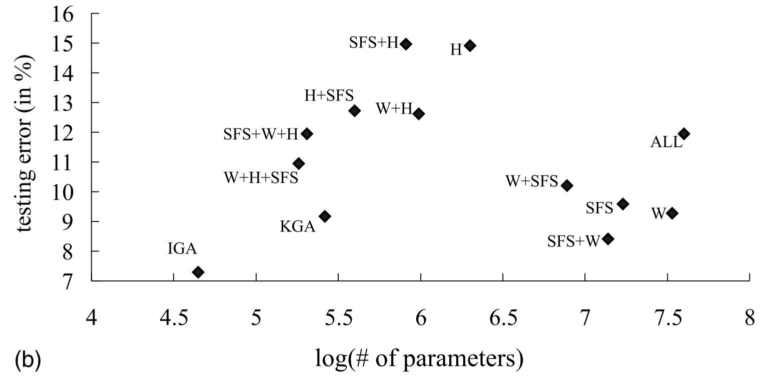


Fig. 2. Scatterplot of various methods: (a) SATIMAGE data, (b) generated data.

ture (e.g., Kuncheva and Jain, 1999). The reported results of various methods are cited here to demonstrate the high performance of the proposed method in optimally designing compact 1-nn classifiers with high classification power. That the solution of IGA dominated all solutions of the existing methods reveals that IGA outperforms all participated methods.

5.2. Experiment 2

In this experiment, the generated data sets with various I and J values are used to compare the performances of IGA and KGA. The terminal conditions of IGA and KGA use 10 000 fitness evaluations. It is well recognized that the weight α may affect the performance of the designed classifier when using the weighted-sum approach for solving the bi-criteria optimization problem. We demonstrate the superiority of IGA to KGA using various α values. An efficient generalized multi-objective evolutionary algorithm (e.g., Ho and Chang, 1999) without using weights based on OA and factor analysis can also be used to effectively solve the investigated problem for obtaining a set of Pareto-optimal solutions.

The generated data sets with $I = 2$ and $J = 20$ for various α values are tested and the experimental results are reported in Table 4. The experimental results using the generated data sets with $\alpha = 0.04$ and $J = 100$ for various I values (overlapping degrees) are reported in Table 5. The experimental results using the generated data sets with $\alpha = 0.04$ and $I = 4$ for various J values (dimensionalities) are reported in Table 6. IGA outperforms KGA in terms of fitness value, classification error, and the number of parameters (N_p) for various α, I, and J values. These results reveal that the IGA-based method can robustly handle high-dimensional patterns with overlapping. Of

Table 4
Average results with the generated data for various α values

Method	KGA					IGA				
α	Fitness F_{KGA}	Error (%)	Card(S_1)	Card(S_1)	N_p	Fitness F_{IGA}	Error (%)	Card(S_1)	Card(S_1)	N_p
0.1	177.0	17.6	12.1	38.1	461	180.6	17.5	15.1	19.2	290
0.2	174.1	19.2	7.2	37.1	267	180.0	18.2	11.1	14.2	158
0.3	172.3	21.9	12.1	30.1	364	178.3	16.5	12.3	17.1	210
0.4	166.6	18.7	10.1	31.2	315	172.3	17.9	11.2	14.1	158
0.5	162.5	21.2	13.1	28.1	368	172.0	14.2	10.1	10.2	103
0.6	157.6	23.7	8.3	25.2	209	169.8	18.4	7.1	15.1	107
0.7	154.6	19.2	11.1	21.1	234	167.3	14.5	9.1	12.2	111
0.8	159.8	23.4	9.2	20.3	187	166.3	13.1	10.1	12.1	122
0.9	152.3	18.9	10.1	24.2	244	161.1	18.4	10.1	11.1	112

Table 5
Average results with the generated data for various I values

Method	KGA					IGA					$F_{\text{IGA}}/F_{\text{KGA}}$
I	F_{KGA}	Error (%)	Card(S_1)	Card(S_1)	N_p	F_{IGA}	Error (%)	Card(S_1)	Card(S_1)	N_p	
1	191.6	12.1	48.0	49.0	2352	197.8	9.0	40.0	14.3	572	1.03
2	176.9	26.4	49.0	44.7	2190	191.2	23.5	47.0	22.0	1034	1.08
3	164.0	30.0	50.3	49.0	2468	182.9	26.3	49.0	29.3	1436	1.12
4	160.8	34.0	47.0	50.7	2383	183.8	31.3	46.7	25.7	1200	1.14

Table 6
Average results with the generated data for various J values

Method	KGA					IGA					$F_{\text{IGA}}/F_{\text{KGA}}$
J	F_{KGA}	Error (%)	Card(S_1)	Card(S_1)	N_p	F_{IGA}	Error (%)	Card(S_1)	Card(S_1)	N_p	
40	167.3	33.9	20.3	47.7	968	174.8	32.2	21.3	25.7	547	1.04
60	165.7	35.1	25.0	44.3	1108	177.7	31.3	30.3	26.7	809	1.07
80	170.1	34.5	37.6	44.7	1681	185.9	31.2	41.3	26.7	1103	1.09
100	160.8	34.0	47.0	50.7	2383	183.8	31.3	46.7	25.7	1200	1.14

course, domain knowledge, heuristics, and a specific set of IGA parameters can further improve the performance.

6. Conclusions

In this paper, we have proposed a method of designing an optimal 1-nn classifier using a novel IGA with an IC based on orthogonal experimental design. Since the solution space is large and complex, IGA superior to conventional GAs is successfully used to solve the large parameter optimization problem. It has been shown empirically that the IGA-designed classifier outperforms existing GA-based and non-GA-based classifiers in terms of classification accuracy and total number of parameters of the reduced sets. Furthermore, IGA can be easily used without domain knowledge to efficiently design 1-nn classifiers with high-dimensional patterns with overlapping.

Acknowledgements

The work of this paper was supported by the National Science Council of ROC under the contract NSC 89-2213-E-035-047.

References

Goldberg, D.E., 1989. Genetic Algorithm in Search Optimization and Machine Learning. Addison-Wesley, Reading MA.

Hart, P.E., 1968. The condensed nearest neighbor rule. IEEE Trans. Inform. Theory 16, 515–516.

Ho, S.-Y., Shu, L.-S., Chen, H.-M., 1999. Intelligent genetic algorithm with a new intelligent crossover using orthogonal arrays. In: GECCO-99: Proc. of the Genetic and Evolutionary Computation Conf., July 14–17, Orlando, Florida, USA, pp. 289–296.

Ho, S.-Y., Chang, X.-I, 1999. An efficient generalized multiobjective evolutionary algorithm. In: GECCO-99: Proc. of Genetic and Evolutionary Computation Conf., July 14–17, Orlando, Florida, USA, pp. 871–878.

Ho, S.-Y., Chen, Y.-C., 2001. An efficient evolutionary algorithm for accurate polygonal approximation. Pattern Recognit. 34, 2305–2317.

Horowitz, E., Sahni, S., Rajasekaran, S., 1997. Computer Algorithms. Computer Science, New York.

Jain, A., Zongker, D., 1997. Feature selection: evaluation, application and small sample performance. IEEE Trans. Pattern Anal. 19 (2), 153–158.

Kuncheva, L.I., Bezdek, J.C., 1998. Nearest prototype classification: clustering, genetic algorithms or random search. IEEE Trans. Systems Man Cybernet. C 28 (1), 160–164.

Kuncheva, L.I., Jain, L.C., 1999. Nearest neighbor classifier: Simultaneous editing and feature selection. Pattern Recognition Lett. 20, 1149–1156.

Leung, Y.-W., Wang, Y., 2001. An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Trans. Evolut. Comput. 5 (1), 41–53.

Mori, T., 1995. Taguchi Techniques for Image and Pattern Developing Technology. Prentice-Hall, New Jersey.

Raudys, S.J., Jain, A.K., 1990. Small sample size effects in statistical pattern recognition: Recommendations for practitioners and open problems. In: Proc. 10th Internat. Conf. on Pattern Recognition. Atlantic City, New Jersey, pp. 417–423.

Stearns, S., 1976. On selecting features for pattern classifiers. In: 3-d Internat. Conf. on Pattern Recognition, Coronado, CA, pp. 71–75.

Wilson, D.L., 1972. Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. Systems Man Cybernet. SMC-2, 408–421.