

# Dimensionality Reduction using GA-PSO

Cheng-Hong Yang<sup>1</sup> Chung-Jui Tu<sup>1</sup> Jun-Yang Chang<sup>2</sup> Hsiou-Hsiang Liu<sup>3</sup> Po-Chang Ko<sup>2</sup>

<sup>1</sup>Dept of Electronic Eng., National Kaohsiung University of Applied Sciences

<sup>2</sup>Dept. of Information Management, National Kaohsiung University of Applied Sciences

<sup>3</sup>Dept. of Tourism Management, National Kaohsiung University of Applied Sciences

## 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 paper, we propose a combination of genetic algorithms (GAs) and particle swarm optimization (PSO) for feature selection. The K-nearest neighbor (K-NN) method with leave-one-out cross-validation (LOCCV) serves as an evaluator for the GAs and the PSO. The proposed method is applied to five classification problems taken from the literature. Experimental results show that our method simplifies features effectively and obtains a higher classification accuracy compared to other feature selection methods.

**Keywords:** Feature Selection, Genetic Algorithms, Particle Swarm Optimization, K-Nearest Neighbor, Leave-one-out cross-validation.

## 1. Introduction

The purpose of the feature selection is to reduce the number of irrelevant features while maintaining acceptable classification accuracy. A good feature selection method can reduce the cost of feature measurement, and increase classifier efficiency and classification accuracy. Feature selection is of considerable importance in pattern classification, data analysis, multimedia information retrieval, medical data processing, machine learning, and data mining applications.

Several methods have been previously used to perform feature selection on training and testing data, for example genetic algorithms [1], sequential search algorithms [2], mutual information [3], neural networks [4], and tabu search [5]. In order to obtain a higher classification accuracy for classification problems, an improved feature selection process is needed. A comparative study on feature selection can be found in

Oh *et al* [6].

In this paper, combining genetic algorithms (GAs) and particle swarm optimization (PSO) is used to implement a feature selection, and K-NN with LOCCV serves as an evaluator for the GAs and the PSO for five multiclass classification problems taken from the literature. PSO improves the inferior performance of populations in the GAs with each generation, and can reduce the number of necessary features, while also avoiding that particles get trapped in a local optimum. The sample classification is accomplished using the K-NN method with LOCCV method based on the Euclidean distance calculation. The results reveal that the proposed method resulted in a higher accuracy compared to the previously published results.

## 2. Methods

### 2.1 Genetic Algorithms

GAs (Genetic Algorithms) were developed by Holland in 1970. GAs are stochastic search algorithms modeled on the process of natural selection, which underlies biological evolution. GAs have been successfully applied in many search, optimization, and machine learning problems [7]. GAs are designed to simulate processes in natural systems necessary for evolution, 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. GAs proceed 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 to every string indicating its fitness for the problem. Standard GAs apply genetic operators such selection, crossover, and mutation on an initially random population in order to compute a whole generation of new strings.

### 2.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique, which was developed by Kennedy and Eberhart in 1995. In PSO, each single candidate solution can be considered "an individual bird of a flock", that is, a particle in the search space. Each particle makes use of its own memory and knowledge gained by the swarm as a whole to find the best solution. All of the particles have fitness values, which are evaluated by a fitness function to be optimized; they also have velocities which direct the movement of the particles. During movement, each particle adjusts its position according to its own experience and according to the experience of a neighboring particle, thus making use of the best position encountered by itself and its neighbor. The particles move through the problem space by following a current of optimum particles. The process is then iterated a fixed number of times or until a minimum error is achieved [8]. In this paper, a simple binary version of a PSO algorithm is used. The position of each particle is given in binary string form, which adequately reflects the simple yes/no choice of whether a feature should be selected or not.

### 2.3 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 [9][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 instance query 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.

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 is simple and provides a reasonable classification performance in most applications. As the 1-NN classifier does not require any user-specified parameters, its classification results are implementation independent.

Initially, each chromosome was coded to a binary alphabetic string  $S = F_1 F_2 \dots F_n$ ,  $n = 1, 2, \dots, m$ ; the bit value {1} represents a selected feature, whereas the bit value {0} represents a non-selected feature. The initial population is generated randomly. The predictive accuracy of a 1-nearest neighbor (1-NN) determined by the leave-one-out cross-validation method is used to

measure the fitness of an individual. The fitness value for the 1-NN evolves according to the Leave-One-Out Cross-Validation (LOOCV) Method for all datasets. The obtained classification accuracy is an adaptive functional value. The rand-based roulette-wheel selection scheme was used in this paper. Standard genetic operators, such as crossover and mutation, are applied without modification. A 2-point crossover operator is used, which chooses 2 cutting points at random and alternatively copies each segment out of the two parents. If a mutation is present, either one of the offsprings is mutated, and their binary representation changes from 1 to 0 or from 0 to 1 after the crossover operator is applied. If the mutated chromosome is superior to both parents, it replaces the worst one of the parents; otherwise, the most inferior chromosome in the population is replaced. Then, PSO is used to improve the inferior performance of the population in the GAs solution with each generation.

In this study, the adaptive functional values were data based on the particle features representing the feature dimension; this data was classified by a 1-NN to obtain classification accuracy. Each particle renewal is based on its adaptive value. The best adaptive value for each particle renewal is  $pbest$ , and the best adaptive value within a group of  $pbest$  is  $gbest$ . Once  $pbest$  and  $gbest$  are obtained, we can keep track of the features of the  $pbest$  and  $gbest$  particles with regard to their position and speed. Each particle is updated according to the following equation.

$$v_{pd}^{new} = w \times v_{pd}^{old} + c_1 \times rand_1 \times (pbest_{pd} - x_{pd}^{old}) + c_2 \times rand_2 \times (gbest_d - x_{pd}^{old}) \quad (1)$$

$$S(v_{pd}^{new}) = \frac{1}{1 + e^{-v_{pd}^{new}}} \quad (2)$$

$$\text{If } (rand < S(v_{pd}^{new})) \text{ then } x_{pd}^{new} = 1; \text{ else } x_{pd}^{new} = 0 \quad (3)$$

In order to avoid particles getting trapped in a local optimum, the  $gbest$  value has to be checked before each particle renewal is updated. If  $gbest$  is the same for a preset number of times (meaning the particle is trapped in a local optimum), the  $gbest$  value is reset while  $pbest$  is kept. The feature after renewal is calculated by the function  $S(v_{pd}^{new})$  (Eq. 2), in which  $v_{pd}^{new}$  is the velocity value. If  $S(v_{pd}^{new})$  is larger than a randomly produced disorder number that is within {0.0~1.0}, then its position value  $F_n$ ,  $n = 1, 2, \dots, m$  is represented as {1} (meaning this feature is selected as a required feature for the next renewal). If  $S(v_{pd}^{new})$  is smaller than a randomly produced disorder number that is within {0.0~1.0}, then its position value  $F_n$ ,  $n = 1, 2, \dots, m$  is represented as {0} (meaning this feature is not selected as a required feature for the next renewal).

The GA was configured to contain 20 chromosomes and was run for 100 generations in each trial. The crossover rate and mutation rate was 1.0 and 0.1, respectively. The number of particles used was 20. The two factors  $rand_1$  and  $rand_2$  are random numbers between (0, 1), whereas  $c_1$  and  $c_2$  are acceleration (learning) factors, with  $c_1 = c_2 = 2$ . The inertia weight  $w$  was 0.9. The maximum number of iterations used in our PSO was 100.

### 3. Results and Discussion

The dataset we used in this study was obtained from the UCI Repository [10]. The data format was arranged as shown in Table 1. In this paper we propose to use a combination of GAs with PSO to serve as the feature selection method for test problems. If only GAs are used for feature selection, the optimal or near optimal genes are not selected. PSO is used to improve the inferior performance of populations in the GAs solution. It also reduces the number of features and avoids particles getting trapped in a local optimum. Binary PSO was used to serve as feature selection for classification problems. It improves performance owing to its smaller number of simple parameter settings. The evaluation method used was the 1-NN method with Leave-One-Out Cross-Validation (LOOCV) for all datasets.

Table 1 shows the format of five classification problems, the evaluation method. Table 2 compares experimental results obtained by other methods from the literature [Oh *et al.*, 2004] with the proposed method. The proposed method obtained the highest classification accuracy for the Glass, Letter, and Segmentation classification problems. For the Glass classification problem, the proposed method obtained 100% classification accuracy, while the number of features that need to be selected is smaller than in other methods. This means that not all features are needed to achieve total classification accuracy. The classification accuracies for the Ionosphere and the Sonar test problems were slightly lower than the classification accuracies of the other feature selection methods, but are still comparable. These results indicate that for different classification problems, the proposed method (binary particle swarm optimization) can serve as a pre-processing tool and help optimize the feature selection process, which leads to an increase in classification accuracy. A good feature selection process reduces feature dimensions and improves accuracy.

For each of the data sets, the recognition rates were measured for four values of D (D/5, 2D/5, 3D/5, Using Genetic Algorithms,” IEEE Trans. Evolutionary Computation, vol. 4, no. 2, pp. 164-171, July 2000.

and 4D/5, D is total number of features) [8]. The proposed method could determine the optima of features used and obtained a higher classification accuracy. For the text problem Letter, the classification accuracy obtained by the proposed method was higher than the results in Oh *et al.*, while the number of features selected was the same. For the text problems Glass and Vowel, the classification accuracies obtained by the proposed method was the same as the results obtained in Oh *et al.* while the number of features used was also same. For the text problems Segmentation, classification accuracies obtained by the proposed method were higher than the results obtained in Oh *et al.* while the number of features used was lower.

We used 1-NN with LOOCV as a fitness function of GAs and PSO, which serves two purposes. First, calculation time can be decreased. Secondly, a higher accuracy can be obtained due to the fact that 1-NN classification with the leave-one-out cross-validation method is used to measure data usage.

### 4. Conclusions

In this paper, we used a combination of GAs and PSO to perform feature selection, and the 1-NN method served as an evaluator for the GAs and PSO fitness functions. Experimental results show that the proposed method simplified feature selection and the total number of parameters needed effectively, thereby obtaining a higher classification accuracy compared to other feature selection methods. The classification accuracy obtained by the proposed method has the highest classification accuracy in three of the five data test problems, and is comparative to the classification accuracy of the other two test problems. The proposed method can serve as an ideal pre-processing tool to help optimize the feature selection process, since it increases the classification accuracy and, at the same time, keeps computational resources needed to a minimum. The proposed method could also be applied to problems in other areas in the future.

### 5. Acknowledgements

This work is partly supported by the National Science Council in Taiwan under grants NSC94-2622-E-151-025-CC3 and NSC93-2213-E-214-037.

### 6. References

- [1] Raymer, M.L., Punch, W.F., Goodman, E.D., Kuhn, L.A., and Jain, A. K., “Dimensionality Reduction
- [2] Pudil, P., Novovicova, J., and Kittler, J., “Floating Search Methods in Feature Selection,” Pattern

- Recognition Letters, vol.15, pp. 1119-1125, 1994.
- [3] Roberto B., "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on Neural Networks*, 5(4):537-550, 1994.
- [4] Brill, F., Brown, D., & Martin, W. (1992). Fast Genetic Selection of Features for Neural Network Classifiers. *IEEE Transactions on Neural Networks*, 3(2), 324--328.
- [5] Zhang, H. and Sun, G.. Feature selection using tabu search method. *Pattern Recognition*, 35: 701-711, 2002.
- [6] 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, Nov. 2004.
- [7] Holland, J. (1992), *Adaptation in Nature and Artificial Systems*, MIT Press, 1992.
- [8] Kennedy, J. and Eberhart, R.C., Particle swarm optimization, IN proceedings of the 1995 IEEE International Conference on Neural Networks, volume 4, pages 1942-1948, Perth, Australia, 1995.
- [9] Cover, T. and Hart, P., "Nearest neighbor pattern classification," *Proc. IEEE Trans. Information Theory*. IT-11, pp. 21-27, 1967.
- [10] P.M. Murphy, and D.W. Aha, "UCI Repository of Machine Learning Databases," technical report, Department of Information and Computer Science, University of California, Irvine, Calif., 1994. Available: <http://www.ics.uci.edu/~mllearn/MLRepository.html>.

Table 1. Format of classification text problems

Datasets	Number of samples	Number of classes	Number of features	Evaluation Method
Glass	214	7	10	1-NN
Letter	15000/5000	26	16	1-NN
Segmentation	210/2100	7	19	1-NN
Ionosphere	201/150	2	34	1-NN
Sonar	104/104	2	60	1-NN

**Legends:** x/y: indicate that x and y represent the number of testing and training samples, respectively.

Table 2. Classification accuracy for the tested data sets

Datasets	d*	SFS	PTA	SFFS	SGA	HGA (1)	HGA (2)	HGA (3)	HGA (4)	GA-PSO	
										d*	%
Glass (D=10)	2	99.07	99.07	99.07	99.07	99.07	99.07	NA	NA	4	<b>100</b>
	4	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>		
	6	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>		
	8	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	NA		
Letter (D=16)	3	47.09	47.09	47.09	47.09	47.09	47.09	47.09	NA	13	<b>96.58</b>
	6	86.20	87.60	87.60	87.60	87.60	87.60	87.60	87.60		
	10	96.12	96.35	96.35	96.35	96.35	96.35	96.35	96.35		
	13	96.42	96.42	96.42	96.42	96.42	96.42	96.42	96.42		
Segmentation (D=19)	4	92.81	92.81	92.81	92.81	92.81	92.81	92.81	92.81	5	<b>97.01</b>
	8	92.95	92.95	92.95	92.95	92.95	92.95	92.95	92.95		
	11	92.95	92.95	92.95	92.95	92.95	92.95	92.95	92.95		
	15	92.57	92.57	92.57	92.57	92.57	92.57	92.57	92.57		
Ionosphere (D=34)	7	93.45	93.45	93.45	95.44	<b>95.73</b>	<b>95.73</b>	<b>95.73</b>	<b>95.73</b>	9	95.16
	14	90.88	92.59	93.79	94.87	<b>95.73</b>	<b>95.73</b>	<b>95.73</b>	<b>95.73</b>		
	20	90.03	92.02	92.88	94.30	94.30	94.30	94.02	94.30		
	27	89.17	91.17	90.88	91.45	91.45	91.45	91.45	91.45		
Sonar (D=60)	12	87.02	89.42	92.31	93.75	94.71	95.67	95.19	95.67	22	95.67
	24	89.90	90.87	93.75	95.67	96.63	96.63	<b>97.12</b>	<b>97.12</b>		
	36	88.46	91.83	93.27	95.67	96.15	96.15	96.15	96.15		
	48	91.82	92.31	91.35	92.79	92.79	93.27	93.27	93.27		