Journal of Electrical Engineering & Technology Vol. 7, No. 2, pp. 245~254, 2012 **245**
http://dx.doi.org/10.5370/JEET.2012.7.2.245

Prototype-based Classifier with Feature Selection and Its Design with Particle Swarm Optimization: Analysis and Comparative Studies

Byoung-Jun Park* and Sung-Kwun Oh†

Abstract – In this study, we introduce a prototype-based classifier with feature selection that dwells upon the usage of a biologically inspired optimization technique of Particle Swarm Optimization (PSO). The design comprises two main phases. In the first phase, PSO selects P % of patterns to be treated as prototypes of c classes. During the second phase, the PSO is instrumental in the formation of a core set of features that constitute a collection of the most meaningful and highly discriminative coordinates of the original feature space. The proposed scheme of feature selection is developed in the wrapper mode with the performance evaluated with the aid of the nearest prototype classifier. The study offers a complete algorithmic framework and demonstrates the effectiveness (quality of solution) and efficiency (computing cost) of the approach when applied to a collection of selected data sets. We also include a comparative study which involves the usage of genetic algorithms (GAs). Numerical experiments show that a suitable selection of prototypes and a substantial reduction of the feature space could be accomplished and the classifier formed in this manner becomes characterized by low classification error. In addition, the advantage of the PSO is quantified in detail by running a number of experiments using Machine Learning datasets.

Keywords: Prototypes, Feature selection, Particle Swarm Optimization (PSO), Wrapper mode of feature selection, Classification, Computational Intelligence (CI)

1. Introduction

One of most widely applied machine learning method is instance-based learning (IBL) which was shown to perform well in a number of challenging learning tasks, cf. [1, 2]. The essence of the method concerns a collection of some stored samples (patterns) using which the ensuing classification tasks are being realized. When an object is provided or the solution to a problem has been found, it is stored in memory for the future use. When a new problem is encountered, memory is searched to find if the same problem has been already solved before. In this sense, it is necessary to define small and consistent subset of data for improving both computing speed and the performance of the method.

Feature selection constitutes a fundamental development phase of pattern recognition and to a significant extent predetermines the effectiveness of the overall classification schemes, cf. [3, 4]. It has become apparent that this task becomes essential both from the standpoint of reduction of an overall computational overload as well as possible enhancements of discriminatory capabilities of the reduced feature space. Any optimization of feature subspaces quite often involves various mechanisms of evolutionary optimization, as evidenced in the pattern recognition literature, see [5-8] including genetic algorithms, evolutionary algorithms, Particle Swarm Optimization (PSO) and others. The spectrum of feature selection techniques is typically split into two main categories such as wrappers and filters, cf. [9, 10]. Filters offer a more general view of the characterization of feature space however they cannot guarantee effectiveness as far as a specific classification scheme is concerned. Wrappers, on the other hand, are focused on the optimization of the feature space which takes into account a specific classification scheme

PSO [11, 12] is an example of an advanced search heuristics inspired by the swarming or collaborative behavior of biological populations. PSO is similar to the genetic algorithms (GAs) in the sense that these two heuristics are population-based search techniques, namely, PSO and GA operate on a population (swarm) and transform it to another set of population in a single iteration with likely improvement using a combination or deterministic and probabilistic rules. PSO is quite often compared with GA [13-17]. Interestingly, most of the literature is concerned with simple comparative scenarios involving experiments exploiting numeric data. An interesting comparison of PSO and GA with a focus on dimensionality aspects of the problems has been offered in [13] and [15], respectively. Statistical comparison using t-test is presented in [16] for several benchmarks. However, this type of statistical comparative analysis has not been completed for Machine Learning data sets .

The objective of this study is to develop a wrapper form

† Corresponding Author: Dep. of Electrical Engineering, The University of Suwon, Korea. (ohsk@suwon.ac.kr)
* Electronics and Telecommunications Research Institute, Korea. (bj_park@etri.re.kr)
Received: January 6, 2011; Accepted: December 12, 2011

of the feature selection scheme based upon a prototype-based classifier driven by the IBL paradigm using PSO and to demonstrate effectiveness (quality of solution) and efficiency (computing cost) through the statistical analysis of results produced by the PSO and GA. In order to improve classification speed and classifier accuracy, it becomes necessary to form a suitable subset of patterns to serve as a collection of "anchor" points of the classifier. In addition to that, given a feature space of high dimensionality, we may anticipate that there is a relatively limited collection of essential features whose discriminatory capabilities arise because of their individual nature and their co-existence in the core set. As the combinatorial nature of the problem of forming the sets of prototypes and features is obvious, we consider the use of PSO as an underlying optimization vehicle. PSO embraces two-level optimization processes in this study. At the first level, PSO chooses P % of patterns as a set of prototypes coming from patterns forming a mixture of c classes. At the second level of the optimization process, PSO is instrumental in the formation of a core set of features that is a collection of the most meaningful and discriminative components of the original feature space. The design of the optimally reduced feature space is investigated in a parametric setting by varying the size of the prototype set (P %) and the size of feature set (d %) used in the proposed construct. In order to emphasize the advantages of the use of PSO for Machine Learning data, we offer a thorough statistical analysis of results produced by the PSO and GA. More specifically, the t-test is used to assess and compare the effectiveness (quality of solution) and efficiency (computing cost) of these two search algorithms. The study provides a comprehensive algorithmic framework of the prototype-based classifier with feature selection and its design with PSO and demonstrates the effectiveness of this approach when being used for a number of data sets. Numerical experiments were carried out and it is shown that a suitable selection of prototypes and a substantial reduction of the feature space could be accomplished that is also accompanied by a lower classification error.

2. Two level processes for prototype-based classifier with feature selection

The prototype-based classifier is a method for classifying objects based on the closest training patterns called prototypes in the feature space. This classifier embraces two selection problems to classify a new pattern to a class. One is the selection of prototype patterns and another one is feature selection.

First level of the optimization process- prototype formation We start with choosing P % of patterns (prototypes) using particle swarm optimization (PSO); these patterns should come from all classes. The prototype-based classifier generates classification results using only P % of patterns. The classifier does not use any model and is only based on computing and using the distance between a pattern and prototypes for achieving classification results. Given a set of N training patterns (prototypes) and a certain pattern without the class label, the classifier finds the prototype being the closest in feature space to this pattern, and then assigns to it the class label of its nearest prototype. The underlying distance between the pattern and the prototype is measured by weighted Euclidean one, that is

$$\|\mathbf{x}-\mathbf{y}\|^2 = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\sigma_i^2} \tag{1}$$

where **x** and **y** are the two patterns positioned in the n-dimensional space while σ_i is the standard deviation of the i-th feature of the patterns.

Second level of the optimization process- feature selection Once the prototypes have been formed, we reduce the overall feature space by choosing a core set of features. Those features are regarded to be the most essential with regard to the classification problem at hand. Quite often their number could be quite reduced in comparison with the dimensionality of the overall feature space. One can consider d % of the total number of features, say 10%, 20%, etc. Namely, "d" expresses a proportion of the number of elements in the sub-feature space to the cardinality of the set of all features to be selected as a core set of features. Considering d % features of the original feature space, we arrive at ${}_nC_{d\times n}$ of possible combinations of the features that could be selected to build this core set. For instance, with n=60 and d % = 20% of features selected to form the core set; we are faced with ${}_{60}C_{0.2\times 60} = {}_{60}C_{12} = 1.399\times 10^{12}$ combinations. This number goes up to 9.25×10^{14} and 3.605×10^{16} when the percentage of the features to be used in the core set of features is equal to 30% and 40%, respectively. Therefore, we use PSO to select d % of features which minimize the classification error.

The overall structure of the two level optimization processes for prototype-based classifier with feature selection is schematically illustrated in Fig. 1. The construction of the core set of features and forming prototypes is computationally challenging and hence requires the use of a suitable optimization mechanism that is capable of dealing with the combinatorial nature of the task.

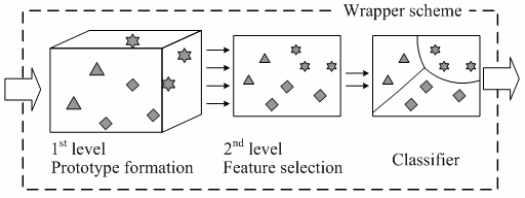


Fig. 1. Overall view of the two level optimization processes for prototype-based classifier leading to the optimal reduction of the feature space

3. Particle Swarm Optimization in the optimal selection

We provide a very brief description of the essence of the PSO and then show its direct use in feature selection.

3.1 Particle swarm optimization

PSO involves two competing search strategies [11, 12]. One deals with a *social* facet of the search. According to this, individuals ignore their own experience and adjust their behavior according to the successful beliefs of individuals occurring in their neighborhood. The *cognition* aspect of the search underlines the importance of the individual experience where the element of population is focused on its own history of performance and makes adjustments accordingly.The basic elements of PSO technique are briefly introduced as follows.

Performance index (fitness). Each particle is characterized by some value of the underlying performance (objective) index or fitness.

Particles. The vectors (particles) of the variables in the n-dimensional search space will be denoted by $\mathbf{p}_1$, $\mathbf{p}_2$, …, $\mathbf{p}_N$. In the search, a swarm is composed of "N" particles involved which give rise to the concept of a swarm.

Best particles. As a particle wanders through the search space, we compare its fitness at the current position with the best fitness value it has so far attained. This is done for each element in the swarm. The location of the particle at which it has attained the best fitness is denoted by **pbest**. Similarly, by **gbest** we denote the best location attained among all **pbest**.

Velocity. The particle is moving in the search space with some velocity which plays a pivotal role in the search process [15, 17]. Denote the velocity of the i-th particle by $\mathbf{v}_i$. From iteration to iteration, the velocity is governed by the following expression

$$v_{ik} = w \cdot v_{ik} + c_1 r_1 (pbest_{ik} - p_{ik}) + c_2 r_2 (gbest_k - p_{ik}) \quad (2)$$

i=1, 2, …, N, k=1, 2,….,n, where, r_1 and r_2 are random values in [0, 1] (viz. coming from a uniform distribution over the unit interval), and c_1 and c_2 are positive constants, called the acceleration constants and referred to as the cognitive and social parameters, respectively. A drawback of the given velocity of PSO is associated with the lack of a mechanism responsible for the control of the magnitude of the velocities, which fosters the danger of swarm explosion and divergence [12]. To address the explosion problem a threshold $\mathbf{v}^{max}$ on the absolute value of the velocity that can be assumed by any particle was incorporated. The particle velocity in the k^{th} coordinate is limited by some maximum value, say v_k^{max}. This limit enhances the local exploration of the problem space and it realistically simulates the incremental changes of human learning. As the above expression shows, c_1 and c_2 reflect the weighting of the stochastic acceleration terms that pull the i-th particle toward $\mathbf{pbest}_i$ and **gbest** positions. Low values allow particles to roam far from the target regions before being tugged back. High values of c_1 and c_2 result in abrupt movement toward, or past, target regions. Typically, the values of these constants are set to 2.0. The inertia factor "w" is a control parameter that is used to establish the impact of the previous velocity on the current velocity. Hence, it influences the tradeoff between the global and local exploration abilities of the particles. For initial stages of the search process, large values enhancing the global exploration of the space are recommended. As the search progresses, the values of "w" are gradually reduced to achieve better exploration at the local level.

3.2 Prototypes and features versus PSO

As a generic search strategy, the particles of PSO has to be represented and a fitness function is introduced to solve a given optimization problem.

Fitness (Performance) function. Given the wrapper mode of the prototype formation and feature selection, we consider the minimization of the classification error to be a suitable fitness measure.

$$\text{Fitness} = \frac{\text{Number of misclassified patterns}}{\text{Number of patterns}} \times 100 \quad (3)$$

Particles. The elements of a vector of a particle consist of the number of pattern and the number of features for the optimal subsets of prototypes and features as shown in Fig. 2. In order to solve the combinatorial problem, we adopt the representation scheme of the search space in the form of the (N+n)-dimensional unit hypercube (N is the number of patterns while n denotes the number of features). The content of the particle is ranked viz. each value in this vector is associated with an index the given value assumes in the ordered sequence of all values encountered in the vector. Here, the elements of a particle for prototypes and features are ranked separately. Considering that we are concerned with P % of all patterns and d % of all features, we pick up the first P×N (0<P<1) and d×n (0<d<1) entries of the vector of the each search space for prototypes and fea-

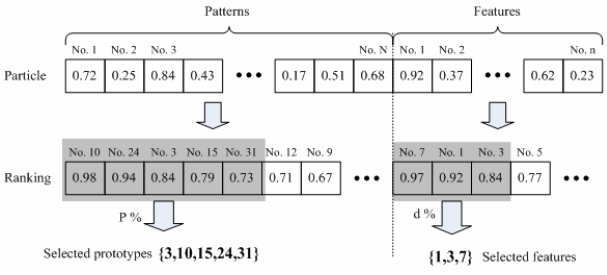


Fig. 2. The PSO formation of the selected prototypes and the reduced feature space

tures, respectively. This produces a set of prototypes defining the training consistent subset and a collection of features forming the reduced feature space. This mechanism of the formation of the prototypes space and the feature space is portrayed in Fig. 3.

Stopping condition. There are two stopping conditions: (a) the algorithm terminates if the objective function does not improve during the last 100 generations, otherwise (b) it terminates after 500 iterations. The size of the population is related to the dimensionality of the search space.

3.3 Metrics and hypothesis testing for comparative analysis

In order to emphasize the advantages of PSO when applying it to problems of Machine Learning, we compare the results produced by PSO with those obtained when running the GA. PSO and GA are similar in the sense of population-based search method and iteration-based updating method for the optimal solution. In other words, PSO and the GA move from a set of points (population) to another set of points in a single iteration with likely improvement using a combination and probabilistic rules. However, the ideas underlie that PSO are inspired not by the evolutionary mechanisms encountered in natural selection, but by the social behavior of flocking organisms, such as bird swarming and fish schooling.

The common features of PSO and GA for the searching procedure are summarized as follows [14-17].

(a) These two techniques are search algorithms based on a population where each individual represents a candidate solution for the optimal solution. This property ensures the algorithms to be less susceptible of getting trapped on local minima.
(b) These two techniques start with a randomly generated population where the fitness values of the individual are used in the evaluation when dealing with the generated population.
(c) These two techniques have various numerical parameters to be carefully selected. There are generation (iteration) and population (swarm) as parameters shown commonly in both algorithms. In the case of the GA, crossover and mutation rates have to be selected. Inertial weight and c1 and c2, need to be chosen in PSO.
(d) These two techniques come with some payoff information (fitness function) to guide the search in the given problem space. Therefore, these can easily deal with non-differentiable functions.
(e) These two techniques use not deterministic rules but probabilistic transition rules. Hence, these two techniques are a kind of stochastic search technique that can explore a complicated and uncertain area.
(f) These two techniques efficiently use historical information to obtain new solutions with enhanced performance and the global nature of a search area.

The advantages of PSO over GA can be summarized as follows [14-17]:

(a) PSO has the control parameter for the balance between the global and local exploration of the search space. This feature enhances the search capability of PSO.
(b) PSO has memory, namely, information of good solutions is retained and shared by all particles; whereas in GA, previous knowledge is destroyed once the population changes.
(b) PSO exhibits algorithmic simplicity. The GA consists of three major operators, selection, crossover and mutation. However, PSO comes with a single operation of velocity calculation.
(c) PSO has a simple implementation and this induces reduction of computation and eliminates the necessity to select the best operator for a given optimization.
(d) Quite often PSO is superior in terms of convergence, speed, and accuracy than other biologically inspired optimization algorithms.

The objective of this section is to statistically compare the performance of the two heuristic search methods, using a representative suite of test problems that are of diverse properties. The t-test is used to assess and compare the effectiveness and efficiency of these search algorithms.

In this study, two hypotheses are tested. The first test is related to the effectiveness (minimum of classification error) of the algorithms and the second is related to the efficiency (computational cost) of the algorithms. Effectiveness is defined as the ability of the algorithm to repeatedly approach at sufficiently near global solutions when the algorithm is started from many random different points in the solution space. In other words, effectiveness is defined as finding a high quality solution for classification error as shown in (3). For the t-test of effectiveness between PSO and GA, the null hypothesis is assumed as the averages of classification error on PSO and GA are equal at the α=0.05 significance level. The second hypothesis is the computational cost test (efficiency). This test directly compares the computational effort required by PSO and GA to solve each of the given problems. Efficiency is defined as a speedy reach at given classification error. For the t-test of efficiency between PSO and GA, the null hypothesis is defined as the means of the stopped iteration on PSO and GA are equal at the α=0.05 significance level.

Furthermore we use some metrics such as minimum, maximum, average, standard deviation and running time for comparison of performance.

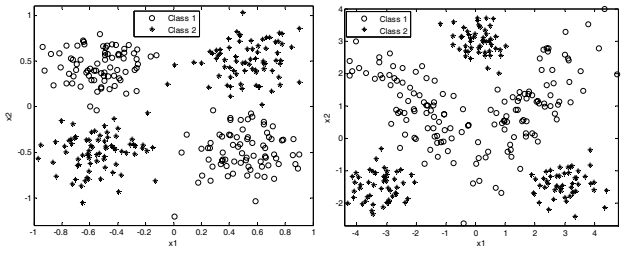
4. Experiments

The numerical studies presented here provide some ex-

perimental evidence behind the effectiveness of the PSO approach. The detailed setup of an extensive suite of experiments is reflective of the methodology we outlined in the previous sections. We used the following values of the parameters: maximum number of generations is 500; swarm size is 150; maximal velocity, v_{max}, is 20% of the range of the corresponding variables; w_{min}=0.4; w_{max}=0.9; and acceleration constants c_1 and c_2 are set to 2.0. The maximal velocity was set to 0.2 for the search carried out in the range of the unit interval [0, 1].

4.1 Synthetic datasets

We start with a series of two-dimensional synthetic examples. The primary objective is to illustrate the classification performance of the proposed classifier. The collections of data involve 2 classes as shown in Fig. 3. Each class consists of 150 patterns. Each group is governed by some Gaussian distribution described by its covariance matrix and the mean vector. In Fig. 3, $\mathbf{m}_i$ denote mean vector of *i*-group and Φ_i is the covariance matrix for the given i-th group.



(a) (b)

Fig. 3. Two-class synthetic datasets:

$$\text{(a)}\ \mathbf{m}_{11}=\begin{bmatrix}-0.5\\0.5\end{bmatrix},\ \mathbf{m}_{12}=\begin{bmatrix}0.5\\-0.5\end{bmatrix},\ \mathbf{m}_{21}=\begin{bmatrix}0.5\\0.5\end{bmatrix},$$

$$\mathbf{m}_{22}=\begin{bmatrix}-0.5\\-0.5\end{bmatrix},\ \Phi=\begin{bmatrix}0.2^2 & 0.0\\0.0 & 0.2^2\end{bmatrix};$$

$$\text{(b)}\ \mathbf{m}_{11}=\begin{bmatrix}-2.0\\1.0\end{bmatrix},\ \Phi_{11}=\begin{bmatrix}1.0 & -0.8\\-0.8 & 1.0\end{bmatrix},$$

$$\mathbf{m}_{12}=\begin{bmatrix}2.0\\1.0\end{bmatrix},\ \Phi_{12}=\begin{bmatrix}1.0 & 0.8\\0.8 & 1.0\end{bmatrix},$$

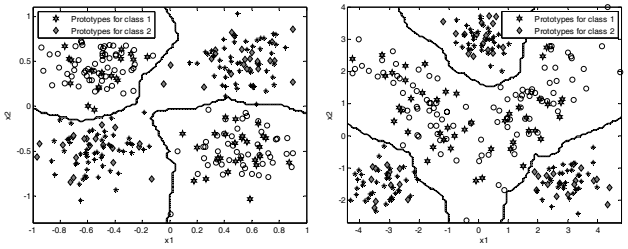
$$\mathbf{m}_{21}=\begin{bmatrix}0.0\\3.0\end{bmatrix},\ \mathbf{m}_{22}=\begin{bmatrix}-3.0\\-1.5\end{bmatrix},\ \mathbf{m}_{32}=\begin{bmatrix}3.0\\-1.5\end{bmatrix},$$

$$\Phi_2=\begin{bmatrix}0.5^2 & 0.0\\0.0 & 0.5^2\end{bmatrix}$$

Table 1 shows classification error and selected features (SF) for the proposed prototype-based classifier. For the given synthetic datasets, 30 % of all patterns were required to be used prototypes so that the patterns can be correctly classified. The classification results and set of prototypes for the four types of data are shown in Fig. 4.

Table 1. Classification error regarded as a function of "P"

P (%)		30	40	50	60	70
Synthetic 2	Error (%)	0.0	0.0	0.0	0.0	0.0
	SF	x_1,x_2	x_1,x_2	x_1,x_2	x_1,x_2	x_1,x_2
Synthetic 3	Error (%)	0.0	0.0	0.0	0.0	0.0
	SF	x_1,x_2	x_1,x_2	x_1,x_2	x_1,x_2	x_2



(a) Synthetic 1 (b) Synthetic 2

Fig. 4. Classification boundaries produced by the prototypes (30 % of data)

4.2 Machine learning datasets

Here we consider a collection of datasets coming from the Machine Learning repository (http://www.ics.uci.edu/~mlearn/MLRepository.html). Table 2 summarizes the pertinent details of the data such as the number of features, number of patterns, and the number of classes.

Table 2. Machine Learning datasets used in the experiments and their essential characteristics

Dataset	Number of features	Number of patterns	Number of classes
Glass	9	214	6
Wine	13	178	3
Vehicle	18	846	4
Image	19	210	7
Dermatology	34	358	6
Sonar	60	208	2

When reporting results, we concentrate on the determination of relationships between the collections of features and obtained classification rates. We also look at the optimal subsets of features constructed with the use of the method. All classification results are reported for the testing data sets.

For the Glass dataset, the relationship between the percentage of features used in the PSO optimization, values of "P" and the resulting classification error is presented in Table 3. Here, "No. of F" is the number of selected features for d % of entire features, "AVG" and "STD" indicates average and standard deviation, respectively. The classification error was computed over 10-fold realization of the experiments.

With the increasing values of "d", the classification error decreases substantially; in the case of P=30% it drops from 34.4 to 10.6 when increasing the number of features from 10% to 80%. The similar downward tendency occurs when dealing with any P % and considering the same increase in

Table 3. Classification error for the Glass

d % (No.off)	P %					AVG±STD over P
	30	40	50	60	70	
10 (1)	34.4±1.82	33.3±1.61	33±2.43	30.3±3.89	25.7±4.04	31.3±4.23
20 (2)	19.1±1.77	18±2.12	11.8±1.69	11±2.34	7.8±2.56	13.6±4.8
30 (3)	14.3±1.68	9.4±1.22	8.4±1.67	6.3±1.25	3.5±2.41	8.4±3.96
40 (4)	11.7±2.33	9.1±1.66	4.6±1.89	4±1.66	2.2±1.49	6.3±3.99
60 (5)	11±2.57	8.4±2.3	5.2±1.63	3.5±1.64	1.5±1.03	5.9±3.88
70 (6)	11±2.13	8.7±2.29	5.7±1.62	3.8±2.51	1.2±1.41	6.1±4.01
80 (7)	10.6±1.92	7.2±1.6	5.4±1.49	3.1±1.46	1.2±0.65	5.5±3.57
90 (8)	10.8±2.48	8.4±2.27	5.2±1.57	3.4±2.08	1.2±1.21	5.8±3.95
100 (9)	12.2±1.61	8.9±1.39	6.6±1.58	6±1.8	1.4±1.53	7±3.91

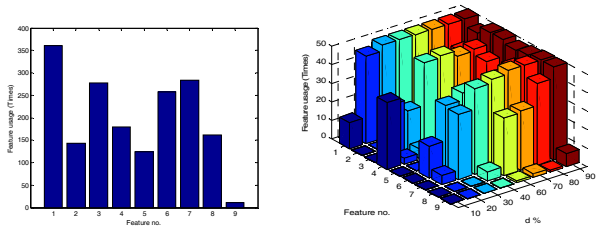
Table 4. Cumulative number of individual features being selected over all "P"

Feature No.	d % (No. of selected features)								Sum
	10(1)	20(2)	30(3)	40(4)	60(5)	70(6)	80(7)	90(8)	
1	14	47	50	50	50	50	50	50	361
2	0	0	1	10	21	26	39	46	143
3	0	28	22	45	46	43	44	50	278
4	36	3	5	4	12	28	41	50	179
5	0	0	1	0	7	32	36	48	124
6	0	17	36	40	34	37	45	49	258
7	0	5	35	46	48	50	50	50	284
8	0	0	0	5	31	32	44	50	162
9	0	0	0	0	1	2	1	7	11

the percentage of features. On the contrary, the use of all features leads ton the lower accuracy of classification. Changes in the values of "P" have far less effect on the classification rate. For any d %, the reduction in the classification error is about 10% over the values of P varying from 30% to 70%. In all cases, we observe that there are an optimal number of features leading to the lowest value of the classification error. Depending upon the values of "d", the dimensionality of the reduced feature space varies in-between 4 and 6.

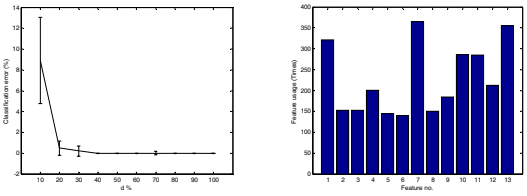
We report the number of occurrences of the features for all experiments. This indicator becomes more illustrative and offers an interesting view at the suitability of the features when forming various reduced feature spaces and using different prototype set sizes. The results obtained in this case are illustrated in Table 4 and Fig. 5. The number of occurrences of a given feature is computed across all values of "P" and "d". Interestingly, there are several dominant features such as refractive index (feature 1) and percentage of magnesium (feature 3) along with two other features, the percentage of potassium (feature 6) and calcium (feature 7). Sodium (feature 2), silicon (feature 5) and iron (feature 9) are of lowest relevance.

The results for some other Machine Learning data sets are reported in terms of the classification error and the number of feature occurrences contributing to the formation of the reduced feature space, see Fig. 6. The results reveal interesting dependencies as to the discriminatory character of the feature space. By inspecting the plots in Fig. 8, there is an evident effect of an "optimal" subset of

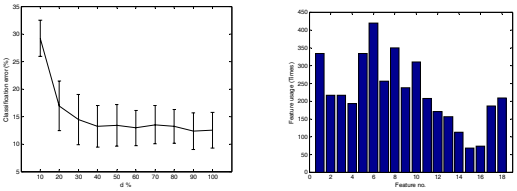


(a) (b)

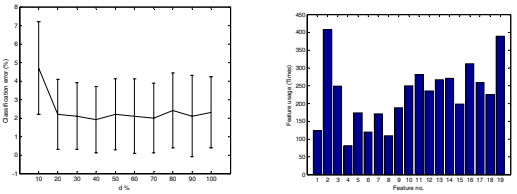
Fig. 5. Cumulative number of occurrence of individual features for the Glass data: (a) Feature usage index over all values of d and P; (b) feature usage index over all P for each d.



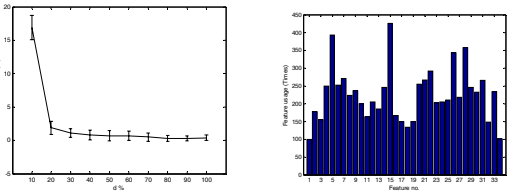
(a) Wine



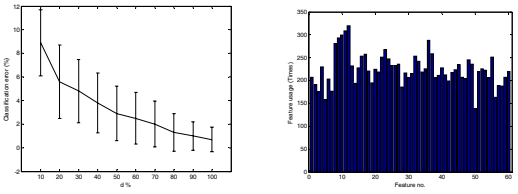
(b) Vehicle



(c) Image



(d) Dermatology



(i) Sonar

Fig. 6. Classification error and the number of occurrences of features obtained in the reduced feature space for the datasets

the feature space: clearly a subset of the original features leads to better classification results when compared with the outcomes of the classifier operating on the entire feature space. There are also datasets in which relationship between classification error and reduced feature space shows a downwards trend linearly for the use of more features. That makes the choice of the reduced feature space less apparent. We observe this phenomenon in the case of the Sonar.

The quality of the reduced feature space is quantified with the use of the classification errors produced by the prototype-based classifiers on the testing set. As Table 5 demonstrates, we have achieved substantially lower classification errors than those already reported in the literature. The results are reported for selected scenarios of the experiments. First, the best classification error obtained across all combinations of "d" and "P" is given. For the size of the prototype set being 50 % of the overall data, the classification error attains higher values which are not surprising given that the sizes of the prototype sets for the best classification error were typically higher in the range of 70 %. To assess the effect of the size of the prototype set, we computed an average classification taken over the range of the sizes of the prototype sets used in the experiments. This provides us with a better view at the diversity of the results implied by the size of the prototype set.

Table 5. Comparison of classification errors reported in the experiments and available in the literature

Dataset	Dimensionality of reduced space & reduction rate (%)	Minimal error (%) and associated values of P	Error (%)at P=50%	Average error (%) over P	Error (%) reported in the literature
Glass	5(44)	1.2±0.65 (70)	5.2 ±1.63	5.9 ±3.88	29.5[18], 25.0[19], 4.7[20]
Wine	5(62)	0.0±0.0 (60)	0.0 ±0.0	0.0 ±0.0	1.5[19], 0.6[20], 1.2[21]
Vehicle	11(39)	7.5±1.62 (70)	12.7 ±1.74	12.9 ±3.22	21.5[20], 28.2[21], 31.2[22]
Image	8(58)	0.0±0.0 (70)	1.4 ±0.67	1.9 ±1.79	2.1[19], 8.4[20]
Dermatology	17(50)	0.0±0.0 (70)	0.4 ±0.36	0.7 ±0.75	3.0[20], 6.8[23], 2.6[24]
Sonar	30(50)	0.0±0.0 (70)	2.9 ±2.1	2.9 ±2.32	14.8[24]

We have looked at more references where some research was completed with respect to the use of instance-based learning classifiers however with the use of different feature selection schemes. Kudo et al. have carried out a comparative study of algorithms for large-scale feature selection (where the number of feature is over 50). The suitability (relevance) of a feature subset is expressed by the leave-one-out correct-classification rate of a nearest-neighbor (1-NN) classifier. There are lots of algorithms for feature selection such as SFS, SBS, PTA, GA, etc. [28]. Selective k-NN classifiers have been considered in [25]. The subset was established by means of sequential feature selection methods. Feature selection using the naïve Bayes rule has presented for the case of multiclass datasets [26]. The expectation maximization algorithm was used to estimate a mixture of modes for each class projected over the features. Tahir et al. proposed a hybrid approach for simultaneous feature selection and feature weighting of k-NN rule based on Tabu Search heuristic [27]. These results are compared with our results for the dimensionality of reduced feature space and classification error averaged over P as shown in Table 6. We can conclude that the proposed method led to better classification results on the reduced feature space than those obtained in some previous studies.

Table 6. Comparison of feature selection for the dimensionality of reduced feature space and classification error

Dataset	Classificationerror (%)	Dimensionality of reduced feature	Reported in the literature		
			literature	Classification error (%)	Dimensionality of reduced feature space
Glass	5.9±3.88	5	[25]	21.5	5
			[26]	25.9	4
			[27]	19.6	6
Wine	0.0±0.0	5	[26]	1.2	6
Vehicle	12.9±3.22	11	[25]	24	13
			[26]	27.4	15
			[27]	25.4	13
			[28]	16~18	8 or 9
Image	1.9±1.79	8	[26]	2.6	15
Sonar	2.9±2.32	30	[27]	5.8	17
			[28]	5~10	20~40

4.3 PSO versus GA for machine learning datasets

To complete a thorough comparative analysis, we carried out experiments with the use of GAs which are one of the commonly used methods of evolutionary optimization. For the machine learning datasets, two performance tests, effectiveness and efficiency, are carried out for both algorithms under consideration, namely, PSO and GA. The parameters used when running PSO and GA are presented in Table 7.

Table 7. Parameters used in PSO and GA

PSO	GA
Max. iterations: 500	Max. generations: 500
Swarm size: 150	Population size: 150
Data type: real	Data type: real
vmax: 20% of a search range	Selection: Roulette
[wmin wmax] = [0.4 0.9]	Mutation (rate): Uniform (0.1)
c1, c2 : 2.0	Crossover (rate): One point (0.75)

In order to come up with representative results concerning computing time, the same conditions were used when running the GA. This concerns the use of the same fitness function, stopping condition, the use of the maximal num-

ber of generations and the size of the population. Furthermore, the results shown in Fig. 10 and Table 8 were obtained when running the experiments in the same computing environment (C programming language and Linux Cluster which offer a 64-bit environment with two dual-core processors and 4GB of RAM) for the case of P=50% and d=50 %. The results represent the outcome of the experiments repeated 10 times. In addition, we used statistical significance measure (t-test) to assure the reliability of the results when being compared with those obtained when running the GA.

Table 8 shows the comparison of effectiveness in terms of statistical results of PSO and GA approaches. In here, 'Min', 'Max', 'Avg' and 'Std' denote the minimum, the maximum, the average and the standard deviation of classification error over 10 independent runs, respectively. CpT is the computing time per individual and calculated by dividing the total algorithm time (iteration and population). Interestingly, in all cases the PSO results are better than those produced by the GA. Likewise the computing time per iteration was also shorter when using the PSO. The last column in Table 8 contains the results of t-test completed for the PSO and GA. We use t-test with α=0.05. Given the results reported for the t-test, we can reject the null hypothesis for most of datasets except Dermatology. This means that the classification errors produced by the PSO are statistically different from those provided by the GA.

Table 8. Comparison of effectiveness in terms of statistical results of PSO and GA approaches

Datasets	Method	Min (%)	Max (%)	Avg (%)	Std (%)	CpT (sec)	t-test	
Glass	PSO	2.9	7.6	5.8	1.9	0.706	h	1
	GA	8.6	13.3	11	1.4	0.725	ρ	0
Wine	PSO	0	0	0	0	0.537	h	0
	GA	0	0	0	0	0.591	ρ	1
Vehicle	PSO	10.2	21.6	13.2	3.8	15.006	h	1
	GA	20.9	24.6	22.9	1.2	16.264	ρ	0
Image	PSO	0	4.8	1.8	1.7	0914	h	1
	GA	1.9	4.8	3.7	1	1.014	ρ	0.0076
Dermatology	PSO	0	1.1	0.6	0.3	3.834	h	0
	GA	0	1.7	0.8	0.5	4.173	ρ	0.2758
Sonar	PSO	1.9	5.8	3.1	1.4	2.174	h	1
	GA	2.9	5.8	4.5	0.9	2.157	ρ	0.0179

As the second comparison of PSO and GA, the results of efficiency test for the same convergence criteria are shown in Table 9. 'Given error' is used as the stopping condition for the PSO and GA, namely, the algorithms are stopped when the classification error is faced with the lower value than the given error while algorithms run. The last generation concerns the results being produced by the algorithm before it was terminated (viz. The stopping criterion has been satisfied). As shown by the results, the average convergence speed of the PSO for the given classification error is higher than that of the GA. For the t-test, the null hypothesis is defined as the mean of the last generation of PSO and GA are equal at the confidence level α=0.05. These results led to the rejection of the null hypothesis for Vehicle, Image and Sonar, and the acceptance of the hypothesis for the Glass, Wine and Dermatology. Based on the results reported in Table 9, the computational effort required by PSO to converge to a solution is substantially lower than that of the GA when considering the same convergence criteria.

Table 9. Comparison of efficiency in terms of statistical results produced by the PSO and GA for the given classification error as the Max value in Table 8

Datasets	Given error (%)	Method	Min	Max	Avg	Std	t-test	
Glass	13.3	PSO	8	27	14.9	5.4	h	0
		GA	3	146	38.2	42.5	ρ	0.1028
Wine	0	PSO	1	6	3.3	1.8	h	0
		GA	2	10	3.9	2.8	ρ	0.5743
Vehicle	24.6	PSO	2	11	6.2	3.2	h	1
		GA	16	89	42	24.0	ρ	0.0002
Image	4.8	PSO	4	62	17.5	17.1	h	1
		GA	5	131	48.5	41.9	ρ	0.0439
Dermatology	1.7	PSO	1	29	13.9	8.9	h	0
		GA	3	46	18.2	14.7	ρ	0.4398
Sonar	5.8	PSO	4	18	9.5	5.0	h	1
		GA	4	95	30.1	26.9	ρ	0.0286

5. Conclusions

In this study, we have introduced a prototype-based classifier with feature selection developed in the framework of Particle Swarm Optimization (PSO). The two-level optimization process of forming the prototypes and the feature space is reflective of the conjecture on the importance of forming a set of prototypes and a core set of features whose discriminatory capabilities emerge through their co-occurrence in these set. The use of the prototype-based classifier is also justifiable considering that this classification scheme is the simplest that could be envisioned in pattern classification.

While the experimental results provide sound evidence behind the selection process showing that the reduced feature spaces led to the better classification results than those obtained in some previous studies, they are also quite revealing in showing that the reduction of the feature space could exhibit different effectiveness. In some cases, the reduction of the dimensionality of the feature space could be high but there could be cases where the elimination of subsets of features could not be strongly justifiable. In addition, when dealing with the selected Machine Learning datasets, the obtained results show that the PSO is preferred over GA in terms of effectiveness (quality of solution) and efficiency (computing cost) of the solutions.

Acknowledgements

This work was supported by National Research Foundation of Korea Grant funded by the Korean Government (NRF-2009-0074928), the GRRC program of Gyeonggi province [GRRC SUWON 2011-B2, Center for U-city Security & Surveillance Technology], and the Converging Research Center Program funded by the Ministry of Education, Science and Technology (No. 2011K000655).

References

[1] F. Fdez-Riverola, E.L. Iglesias, F. Diaz, J.R. Mendez, J.M. Corchado, "SpamHunting: an instance-based reasoning system for spam labeling and filtering," *Decision Support Systems*, vol. 43, pp. 722-736, 2007.

[2] C. Gonzalez, J.F. Lerch, C. Lebiere, "Instance-based learning in dynamic decision making," *Cognitive Science*, vol. 27, pp. 591-635, 2003.

[3] C.M. Bishop, *Neural networks for Pattern Recognition*, Oxford Univ. Press, 1995.

[4] J.-X. Huang, K.-S. Choi, C.-H. Kim, Y.-K. Kim, "Feature-Based Relation Classification Using Quantified Relatedness Information," *ETRI Journal*, vol. 32, no. 3, pp. 482-485, 2010.

[5] X. Wang, J. Yang, X. Teng, W. Xia R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognition*, vol. 28, no. 4, pp. 459-471, 2007.

[6] I.-S. Oh, J.-S. Lee, B.-R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1424-1437, 2004.

[7] X. Wang, J. Yang, R. Jensen, X. Liu, "Rough set feature selection and rule induction for prediction of malignancy degree in brain glioma," *Computer Methods and Programs in Biomedicine*, vol. 83, pp. 147-156, 2006.

[8] F. Zhu, S. Guan, "Feature selection for modular GA-based classification," *Applied Soft Computing*, vol. 4, pp. 381-393, 2004.

[9] M.E. Farmer, A.K. Jain, "A wrapper-based approach to image segmentation and classification," *IEEE Trans. Image Processing*, vol. 14, pp. 2060-2072, 2005.

[10] Y. Liu, Y.F. Zheng, "FS_SFS: A novel feature selection method for support vector machines," *Pattern Recognition*, vol. 39, pp. 1333-1345, 2006.

[11] J. Kennedy, "The particle swarm: social adaptation of knowledge," *Proc. IEEE Int. Conf. Evolutionary Comput*, pp. 303-308, 1997.

[12] K.E. Parsopoulos, M.N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Trans. Evolutionary Computation*, vol. 8, pp. 211-224, 2004.

[13] B. Bhanu, Y. Lin, "Genetic algorithm based feature selection for target detection in SAR images," *Image and Vision Computing*, vol. 21, pp. 591-608, 2003.

[14] R. Hassan, B. Cohanim, O. de Weck, "A comparison of particle swarm optimization and the genetic algorithm," *Proc 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural, Dynamics & Materials Conference*, pp. 1-13, 2005.

[15] B. Liu, L. Wang, Y.-H. Jin, F. Tang, D.-X. Huang, "Improved particle swarm optimization combined with chaos," *Chaos, Solitons & Fractals*, vol. 25, pp. 1261-1271, 2005.

[16] J. Kennedy, W.M. Spears, "Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on multimodal problem generator," *Proc IEEE Int Cong Evol Comp*, pp. 78-83, 1998.

[17] A.E. Yilmaz, M. Kuzuoglu, "Calculation of optimized parameters of rectangular microstrip patch antenna using particle swarm optimization," *Microwave and Optical Technology Letters*, vol. 49, pp.2905-2907, 2007.

[18] E.L. Allwein, R.E. Schapire, "Reducing multiclass to binary: a unifying approach for margin classifiers," *The Journal of Machine Learning Research*, vol. 1 pp. 113-141, 2001.

[19] S. Dzeroski, B. Zenko, "Stacking with multi-response model trees," *Proc. of The Third Int. Workshop on Multiple Classifier Systems, MCS*, pp. 201-211, 2002.

[20] T. Li, S. Zhu, M. Ogihara, "Using discriminant analysis for multi-class classification," *Third IEEE Int. Conf. on Data Mining ICDM 2003*, pp. 589-592, 2003.

[21] A.J. Perez-Jimenez, J.C. Perez-Cortes, "Genetic algorithms for linear feature extraction," *Pattern Recognition Letters*, vol. 27, pp. 1508-1514, 2006.

[22] X. Zhang, G. Dong, K. Ramamohanarao, "Information-based classification by aggregating emerging patterns, Intelligent Data Engineering and Automated Learning," LNCS, vol. 1983, pp. 48-53, 2000.

[23] C.K. Loo, M.V.C. Rao," Accurate and reliable diagnosis and classification using probabilistic ensemble simplified fuzzy ARTMAP," *IEEE Trans. Knowledge and Data Engineering*, vol. 17, pp. 1589-1593, 2005.

[24] M. Rocha, P. Cortez, J. Neves, "Simultaneous evolution of neural network topologies and weights for classification and regression, Computational Intelligence and Bioinspired Systems," *LNCS*, vol. 3512, pp. 59-66, 2005.

[25] F. Pernkopf, "Bayesian network classifiers versus selective k-NN classifier," *Pattern Recognition*, vol. 38, pp. 1-10, 2005.

[26] J.M. Sotoca, J.S. Sanchez, F. Pla, "Attribute relevance in multiclass data sets using the naïve bayes rule," *Proc. of the 17th International Conference on Pattern Recognition*, vol. 3, pp. 426-429, 2004.

[27] M.A. Tahir, A. Bouridane, F. Kurugollu, "Simultaneous feature selection and feature weighting using hybrid tabu search/k-nearest neighbor classifier," *Pattern Recognition Letters*, vol. 28, pp. 438-446, 2007.
[28] M. Kudo, J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," *Pattern Recognition,* vol. 33, pp. 25-41, 2000.

Byoung-Jun Park He received the B.S., M.S., and Ph.D. degrees in control and instrumentation engineering from Wonkwang University, Korea, in 1998, 2000, and 2003, respectively. From 2005 to 2006, he held a position of a Postdoctoral Fellow in the Department of Electrical and Computer Engineering, University of Alberta, Canada. From 2008 to present, he worked as a Senior Researcher at ETRI. His research interests encompass computational intelligence, pattern recognition, granular and relational computing, and IT Convergence.

Sung-Kwun Oh He received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Yonsei University, Seoul, Korea, in 1981, 1983, and 1993, respectively. During 1983-1989, he was a Senior Researcher of R&D Lab. of Lucky-Goldstar Industrial Systems Co., Ltd. From 1996 to 1997, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada. He is currently a Professor with the Department of Electrical Engineering, University of Suwon, Suwon, South Korea. His research interests include fuzzy system, fuzzy-neural networks, automation systems, advanced computational intelligence, and intelligent control. He currently serves as an Associate Editor of the *KIEE Transactions on Systems and Control*, *International Journal of Fuzzy Logic* and *Intelligent Systems of the KFIS*, and *International Journal of Control, Automation, and Systems of* the ICASE, South Korea.