Proceedings of the 2005 International Conference on Simulation and Modeling
V. Kachitvichyanukul, U. Purintrapiban, P. Utayopas, eds.

A TWO-LEVEL COEVOLUTIONARY APPROACH TO MULTIDIMENSIONAL PATTERN CLASSIFICATION PROBLEMS

Ki-Kwang Lee and Wan Chul Yoon

Department of Industrial Engineering,
Korea Advanced Institute of Science and Technology
Taejon, South Korea

ABSTRACT

This paper proposes a coevolutionary classification method to discover classifiers for multidimensional pattern classification problems with continuous input variables. The classification problems may be decomposed into two sub-problems, which are feature selection and classifier adaptation. A coevolutionary classification method is designed by coordinating the two sub-problems, whose performances are affected by each other. The proposed method establishes a group of partial sub-regions, defined by regional variable set, and then fits a finite number of classifiers to the data pattern by combining a genetic algorithm and a local adaptation algorithm in every sub-region. A cycle of the cooperation loop is completed by evolving the sub-regions based on the evaluation results of the fitted classifiers located in the corresponding sub-regions. The classifier system has been tested with well-known data sets from the UCI machine-learning database, showing superior performance to other methods such as the nearest neighbor, decision tree, and neural networks.

1 INTRODUCTION

Classification learning systems are useful for decision-making tasks in many application domains, such as financing, manufacturing, control, diagnostic applications, and prediction systems where classifying expertise is necessary (Weiss and Kulikowski 1991). This wide range of applicability motivated many researchers to further refine classification methods in various domains (Jain et al. 2000, Simpson 1992). The major objective of classification is to assign a new data object represented as features (sometimes referred to as attributes or input variables) to one of the possible classes with a minimal rate of misclassification. Solutions to a classification problem have been characterized in terms of parameterized or non-parameterized separation boundaries that could successfully distinguish the various classes in the feature space (Pal et al. 1998). A primary focus of study to build the separation boundaries has been on *learning from examples*, where a classifier system accepts case descriptions that are pre-classified and then the system learns a set of separation surfaces that can classify new cases based on the pre-classified cases (Nolan 2002). Various learning techniques have been contrived to design the separation surfaces, employing a variety of representation methods, such as mathematical functions, neural networks, fuzzy if-then rules, and decision trees.

The method proposed in this paper to construct a classifier system consists of two levels, i.e. determining the feature space and searching the separation boundaries. The number and diversity of possible classifying features would easily dominate the amount of available decision data. When the number of features and the possible patterns are huge, a method of feature selection should be devised to find the most relevant features before automatic classification or decision learning (Liu and Setiono 1998). We represent a feature set as a set of pairs of a feature and its operational range, which actually represents a hyper-rectangular sub-region in the dimensional space. The feature sets are obtained by iteratively adding a feature and its available interval to a current feature set in a sequential increasing manner, so that the sub-region expanded from the added feature can include as many positive examples as possible. In every sub-region, the classifiers, which are delineated by geometrical ellipsoids, adjust their parameters to search the separation boundaries by using a hybrid method of a genetic algorithm (GA) and a heuristic local search algorithm. Abe and Thawonmas (1997) showed that a classifier with ellipsoidal regions had the generalization ability comparable or superior to those of classifiers with the other shapes. Motivated by the result of Abe and Thawonmas (1997), the ellipsoids are adopted to fit the usual non-linear boundaries which hyper-rectangular sub-region cannot represent accurately. After the evolution stabilizes for the ellipsoidal regions in every feature sets represented by sub-regions in the dimensional space, the feature sets themselves are subject to evolution based on the evaluation results of the fitted ellipsoids located in the corresponding sub-regions. The two-level coevolution process is iterated until the termination condition is satisfied.

The rest of this paper is organized as follows. In section 2, we define the pattern classification problem considered in this paper. Section 3 describes the details about a proposed classifier system. Section 4 shows the experimental results from evaluating the performance of the proposed classifier system. Finally, conclusions are stated in section 5.

2 CLASSIFICATION PROBLEM

Let us assume that a pattern classification problem has c classes in an n-dimensional pattern space $[0, 1]^n$ with continuous input variables. It is also supposed that a finite set of points $X = \{\mathbf{x}_p, p = 1, 2, \ldots, m\}$ is given as the training data. Suppose that each point of X, $\mathbf{x}_p = (x_{p1}, x_{p2}, \ldots, x_{pn})$, is assigned to one of the c classes, and let the corresponding subsets of X, having $N_1, N_2, \ldots, N_c$ points, respectively, be denoted by $X_1, X_2, \ldots, X_c$. Because the pattern space is $[0, 1]^n$, the feature values are $x_{pj} \in [0, 1]$ for $p = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. It is desired that the subset X_i ($i = 1, 2, \ldots, c$) are isolated by classifying regions labeled L_{ij} ($j = 1, \ldots$), so that the new points can be assigned to one of the c classes. An example of such a pattern classification problem is shown in Figure 1 where $c = 2$ (i.e., two-class problem), $n = 2$ (i.e., two-dimensional pattern space) and $m = 40$ (i.e., 40 training patterns).

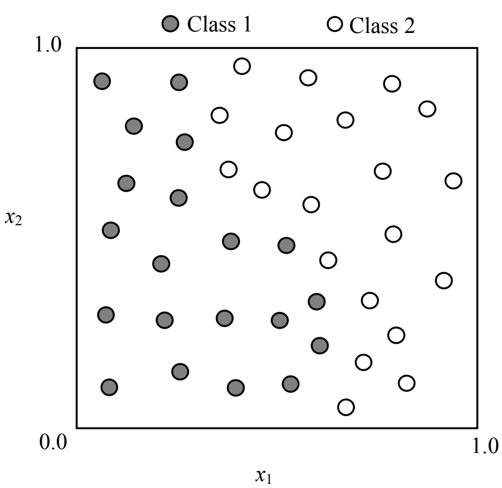


Figure 1: A two-class classification problem with the two-dimensional pattern space $[0, 1]^2$.

3 A COEVOLUTIONARY CLASSIFIER SYSTEM

For the classification problem defined the previous section, we propose a coevolutionary method to construct separating boundaries of classes in feature subsets on the basis of the training data. The procedure for establishing a classifier system, which produces class boundaries, consists of two phases as follows.

- *Phase 1.* Determine the feature subspaces of hyper-rectangles. In each subspace, a subset of selected features is used.
- *Phase 2.* Search the separation boundaries by evolving hyper-ellipsoids in the determined feature subspaces through a genetic algorithm.

Each phase of the procedure will be explained in detail in the following sections.

3.1 Initial determination of the feature subspaces

We propose in this subsection a spatial feature selection method by constructing subspaces, in each of which a specific subset of relevant features is to be considered. As a result, each subspace has different dimensions than the others. The spatial feature set constitutes pairs of a feature and its valid interval, which actually represents a hyper-rectangular subspace in the dimensional space. The feature subspaces are initially established so as to include as many positive examples as possible and to exclude negative examples according to its initial default class.. A spatial feature set is obtained by iteratively adding a feature and its available interval to the current feature set in a sequential manner, so that the subspace expanded by the added feature can include as many positive examples as possible. In later phases, the feature spaces will be subject to an evolutionary process to maximize the performance of ellipsoids in them.

Kudo and Shimbo (1993) proposed a method to obtain the hyper-rectangles that was similar to the initial establishment method of ours. However, their approach differs from ours in the following points. First, their feature selection approach is performed in a backward manner, removing redundant features from the initiated maximum hyper-rectangles. It requires much computation cost to search the maximum hyper-rectangles in a large training dataset, while our method does not require the additional search for the maximum hyper-rectangles due to the forward sequential feature selection. Second, a binarization procedure is needed to apply their method to data with continuous features, while ours can be directly employed to the data with continuous features. The proposed method to build the feature subspaces is presented below and also illustrated by Figure 2.

Step 1. An initial subspace T is constructed by defining the interval in the one-dimensional space of the reference feature v_0. The lower and upper limits of the interval are established on the basis of the nearest negative examples from some consecutive positive examples selected among training data set on the axis v_0.

Step 2. The initial subspace T is split into subspaces, T_i, i=1, 2, …, on the expanded dimensional space of the existing features and the new feature v'. Each T_i is created by defining lower and upper limits of its interval on the new feature v' in the same way of the Step 1.

Step 3. Each of the divided subspaces expands its interval along the existing features to the nearest negative example while keeping the interval of the new feature v'.

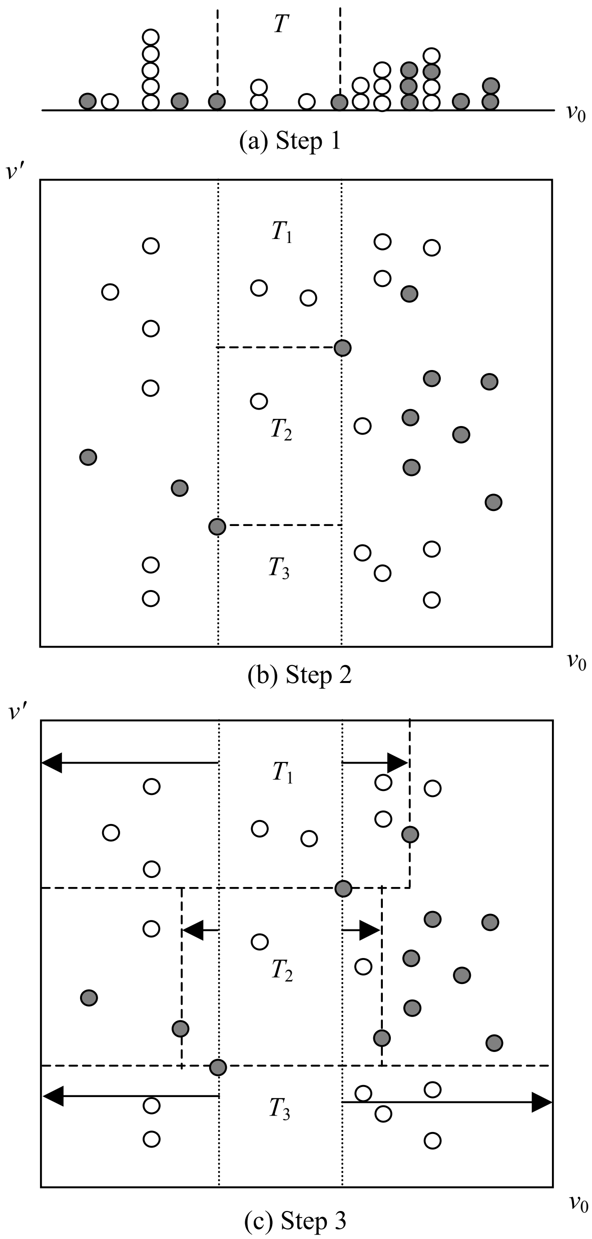


Figure 2: Example of feature subspaces generation

Step 4. If the largest one T^* among subspaces expanded from the initial subspace T includes more positive examples than those of the initial subspace T then repeat Step 2 to Step 3 by considering the largest subspace T^* in Step 3 as an initial subspace T in Step 2. Otherwise the initial subspace T in Step 2 is inserted to the set of final subspace set G on condition that the initial subspace in Step 2 has at least one positive example that is not included in the set of final subspaces.

Step 5. For every example in the training data set, the above procedure from Step 1 to Step 4 repeated to get the set of final subspaces G.

3.2 Evolution of the ellipsoidal classifiers

In every feature subspace, the classifiers represented by ellipsoids adjust their parameters to search the optimal separation boundaries through a hybrid GA. A set of adaptive operations is devised and used for the local search in the hybrid GA.

3.2.1 Classifier representation with ellipsoids

Assume that the data subset X_i for class C_i, where i = 1, …, c, is covered by several ellipsoidal regions L_{ij} (j = 1, …), where L_{ij} denotes the jth region for class C_i. The ellipsoidal region L_{ij} is defined by two foci, $\mathbf{f}_{ij}^{(1)}$ and $\mathbf{f}_{ij}^{(2)}$ and a constant, i.e., size factor, D_{ij} as follows:

$$L_{ij}:\ dist(\mathbf{x}, \mathbf{f}_{ij}^{(1)}) + dist(\mathbf{x}, \mathbf{f}_{ij}^{(2)}) \le D_{ij} \qquad (1)$$

where $dist(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^t(\mathbf{x}-\mathbf{y})}$.

For each ellipsoidal region L_{ij}, we define the following classification rule:

R_{ij}: If $\mathbf{x}$ is in L_{ij} then $\mathbf{x}$ belongs to class C_i (2)

where R_{ij} denotes the label of the jth rule for class C_i.

3.2.2 Classifier strength and determination of class

For the pattern classification, it is reasonable to assume that the degree of membership of $\mathbf{x}$ for classification rule (2) increases as $\mathbf{x}$ moves toward the center of the ellipsoid L_{ij}, and decreases as $\mathbf{x}$ moves away from the center. To realize this characteristic, the degree of membership of $\mathbf{x}$ for a rule R_{ij} is defined as follows.

$$d_{ij}(\mathbf{x}) = \frac{D_{ij}}{dist(\mathbf{x}, \mathbf{f}_{ij}^{(1)}) + dist(\mathbf{x}, \mathbf{f}_{ij}^{(2)})} \qquad (3)$$

If the value of $d_{ij}(\mathbf{x})$ in (3) is larger than 1, it indicates that point $\mathbf{x}$ is located within the ellipsoid L_{ij}. The value of (3) is less than 1 when $\mathbf{x}$ lies out of the boundary of the ellipsoid. Now the degree of membership of $\mathbf{x}$ for class C_i, denoted as $d_i(\mathbf{x})$, is given by $d_i(\mathbf{x}) = \max_{j=1,\cdots}\{d_{ij}(\mathbf{x})\}$. The class of input $\mathbf{x}$ is then determined as class C_{i^*} such that $d_{i^*}(\mathbf{x})$ is the maximum among $d_i(\mathbf{x})$, i = 1, …, c.

3.2.3 Chromosome representation and population initiation

The chromosomes are represented by strings of a floating-point value in [0, 1], encoding the parameters of ellipsoids. Figure 3 shows the structure of a chromosome in a three-dimensional feature subspace obtained in subsection 3.1 as an example.

	x_1	x_2	x_3	x_1	x_2	x_3	
1	0.2	0.1	0.8	0.9	0.3	0.5	0.4
class	$\mathbf{f}^{(1)}$			$\mathbf{f}^{(2)}$			size factor

Figure 3: An example of a chromosome in three-dimensional feature subspace.

An initial population is generated in such a way that each individual assigned to one of the classes is encoded in terms of two foci, $\mathbf{f}^{(1)}$ and $\mathbf{f}^{(2)}$, and a size factor D, which are randomly allocated in the pattern space $[0, 1]^n$ to ensure sufficient diversity in the population. For each individuals in half of the population, then, one of the foci is seeded with randomly selected training sample point for providing a good starting solution. The number of individuals with a certain class C_i in the population, denoted by *Pop(i)*, is determined in proportion to the number of training data with the same class. Consequently, the size of the population, denoted by *Pop_size*, is defined as the sum of *Pop(i)*, i=1, …, c.

3.2.4 Fitness computation

Two measures are considered to evaluate an ellipsoid based on the classification result of the corresponding classifier: generalization ability and classification rate. In order to obtain good generalization ability of an ellipsoid, the region that the ellipsoid covers needs to grow as large as possible. Therefore, when we divide the training data by ellipsoids, the number of data belonging to an ellipsoid should not be too small. For the high classification rate of an ellipsoid, the number of correctly classified data should be large relative to the number of incorrectly classified data among data belonging to the ellipsoid.

Considering the two measures, the fitness value of each ellipsoid is defined as follows.

$$fitness(L_{ij}) = NC(L_{ij}) - weight(L_{ij}) \times NI(L_{ij}) \qquad (4)$$

where $fitness(L_{ij})$ is the fitness value of the ellipsoid L_{ij}, $NC(L_{ij})$ is the number of training data that are correctly classified by L_{ij}, $NI(L_{ij})$ is the number of training data that are incorrectly classified by L_{ij}, and $weight(L_{ij})$ is the weight value that multiplies $NI(L_{ij})$.

The weight value for an ellipsoid L_{ij} is used to determine the tradeoff between the generalization ability and the expected classification rate of the ellipsoid on the basis of the ratio of the number of data with same class C_i to the total number of remaining data. Given a training data set with a large value of the ratio the ellipsoids is apt to be large with the large fitness value caused by large expected value of $NC(\cdot)$ and small expected value of $NI(\cdot)$. This will over-emphasize the generalization power relative to the classification rate. On the other hand, a small value of the ratio has the ellipsoids be small with the small fitness value caused by small $NC(\cdot)$ and large $NI(\cdot)$, which leads to low generalization ability. If the ratio has a large value, the expected classification rate should be emphasized with a large value of weight. Otherwise the generalization ability should be emphasized with a small value of weight. Based on the above relation, the weight value of each ellipsoid is calculated as follows.

$$weight(L_{ij}) = \exp\left\{\alpha \times \left(\frac{N_i}{N_{remain}} - \beta\right)\right\} \qquad (5)$$

where N_i is the number of data of which class is C_i among remaining training data, N_{remain} is the number of total remaining training data, and α ($\alpha > 1$) and β ($0 < \beta < 1$) is constant.

3.2.5 Genetic operations

In order to generate new offspring for class C_i, a pair of individuals with the same class C_i is selected from the current population. Each individual is selected by the following selection probability based on the roulette wheel selection with the linear scaling:

$$P(L_{ij}) = \frac{fitness(L_{ij}) - fitness_{\min}(S_i)}{\sum_{L_{ik} \in S_i}\{fitness(L_{ik}) - fitness_{\min}(S_i)\}} \qquad (6)$$

where $fitness_{\min}(S_i)$ is the minimum fitness value of the individuals in the current set S_i.

From the selected pair of ellipsoids, the arithmetic crossover for randomly taken genes generates two offspring. For an example of the i-th genes, a_i and b_i of the selected pair of ellipsoids are replaced by $\lambda a_i + (1-\lambda) b_i$ and $(1-\lambda) a_i + \lambda b_i$ respectively, where $0<\lambda<1$. Note that the size factor is determined by a random number drawn from a uniform distribution $U(dist(\mathbf{f}^{(1)\prime}, \mathbf{f}^{(2)\prime}), 1)$ in order to keep the size of the ellipsoid greater than distance between its two modified foci.

Each parameter of ellipsoids generated by the crossover operation is randomly replaced using a random number from $U(0, 1)$ at a pre-specified mutation probability. As in the crossover operation, the size factor is recomputed with the modified distance between the two altered foci.

3.2.6 Adaptive operations

The adaptive operations consist of three operations, i.e., expansion, avoidance, and move. The most probable one of the three operations is selected for each ellipsoid based on its fitness value. The ellipsoid with a positive value of fitness is expanded to have a chance to contain more data patterns. If the fitness value of an ellipsoid is less than zero (i.e., an ellipsoid contains at least one misclassified data), the ellipsoid rotates or contracts to avoid the misclassified examples. Finally if an ellipsoid has a zero value of fitness (i.e., an ellipsoid does not contain any training data), the ellipsoid moves to other location in the pattern space. The fitness value of an ellipsoid can be zero even though the ellipsoid contains data from equation (4). However, there is a bare possibility that the number of correctly classified data is same as the number of misclassified data multiplied by weight value because the value of weight in equation (5) is a real number calculated by an exponential function with a real number of parameter. Nevertheless, the overall performance does not take a sudden turn for the worse.

In summary, each ellipsoid in a pool is updated by iteratively adapting one of three adaptive operations based on the fitness value of the ellipsoid:

- Avoidance: If the fitness value of an ellipsoid is negative, avoid misclassified data located in the ellipsoid.
- Expansion: If the fitness value of an ellipsoid is positive, expand the ellipsoid.
- Move: If the fitness value of an ellipsoid is zero, move the ellipsoid to other location in the pattern space.

The following three subsections describe these operations in detail.

3.2.6.1 Avoidance

We propose three methods to avoid the misclassified examples considering the locations of the misclassified examples. Figure 4 illustrates the three methods in two-dimensional pattern space. The first one is to avoid the misclassified examples by rotating the ellipsoid as shown in Figure 4 (a). The rotation method is selected when the misclassified examples are located in near to boundary of the ellipsoid like the shading region in Figure 4 (a). We rotate the ellipsoid by moving the focus nearby the misclassified examples in parallel to one variable axis while fixing the other focus and its size factor.

The other avoidance method is to shrink the ellipsoid as shown in Figure 4 (b). The contraction method is applied when the misclassified examples are located in the shading area of Figure 4 (b). To avoid the misclassified examples, the ellipsoid is shrunk by moving two foci to the opposite directions of one another and its size factor fixed.

Another avoidance method is carried out when the misclassified examples are located in the shading part of Figure 4 (c). The misclassified examples is hard to avoid by the previous two methods because they are located around the center of the ellipsoid. Therefore we propose the third method that randomly modifies the location of a randomly selected focus to avoid the misclassified examples around the center of the ellipsoid.

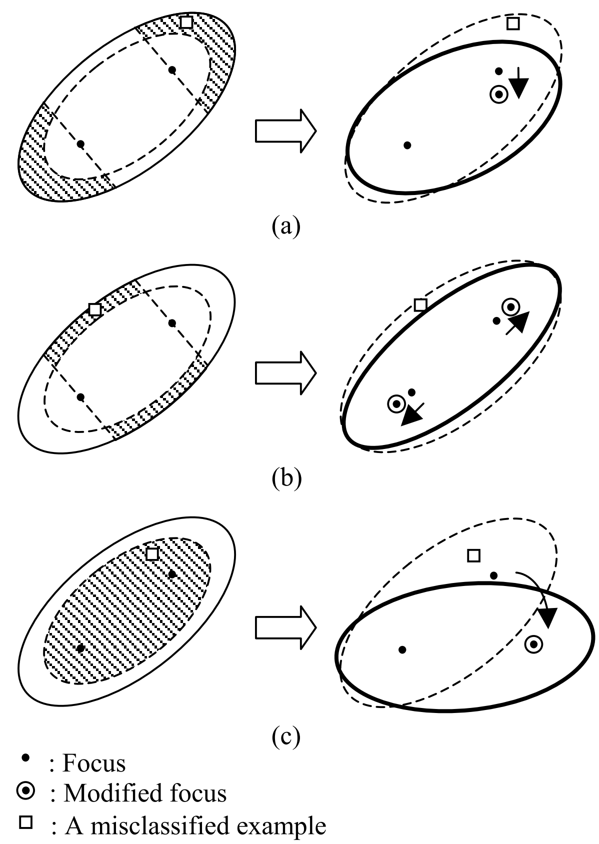
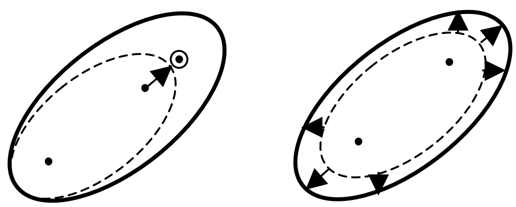


Figure 4: Three methods of avoidance operation

3.2.6.2 Expansion

An ellipsoid is expanded by one of following two methods (i.e., directed and undirected expansion) as shown in Figure 5, which assumes the ellipsoid in a two-dimensional pattern space. In the directed expansion as expressed in Figure 5 (a), the ellipsoid extends its area to the opposite direction of a location-fixed focus, which is randomly selected among two foci. The direction can be chosen rather efficiently by analyzing data around the ellipsoid. The efficient selection of the expanding direction can reduce the number of iterations and help the ellipsoids fit class boundaries more effectively. However, it may not be critical to overall performance of a resulting classifier. The fitness of the ellipsoid expanded to the wrong direction may be degraded. However, the reduced fitness value will be compensated by the avoidance operation invoked by the decreased fitness value in the next iteration. Moreover, the rotation operation can rectify the wrong direction during the avoidance step. Thus we consider the random method only in deciding the expanding direction. The directed expansion method is performed by modifying the locations of the foci and the size factor.

In the undirected expansion, the ellipsoid is enlarged toward all directions as depicted in Figure 5 (b). The undirected expansion of the ellipsoid is carried out by increasing only its size factor.



(a) (b)

• : Focus

⊙ : Modified focus

Figure 5: Two methods of expansion operation

3.2.6.3 Move

If the ellipsoid does not contain any examples, it randomly moves to other location in the pattern space. For the fast adaptation of the ellipsoid, the ellipsoid needs to be moved to the area where no ellipsoids exist and training examples are densely distributed. Thus if there exist training examples which are not included in any ellipsoids, we place a focus to randomly selected one among the training examples.

3.2.7 Update of the population

The proposed hybrid GA procedure applies genetic operations and adaptive operations after population elitist selection (Eshelman 1991). With the population elitist selection, pre-defined *Pop_size* individuals are selected from the current population and a set of the newly generated offspring. This updating method guarantees that the best *Pop_size* individuals seen so far always survived.

3.2.8 Termination Test

The proposed hybrid GA iterates the GA operations and the local improvement procedure (i.e., adaptive operations) until a termination criterion is met. The termination criterion used in this study is to terminate the iteration when either all the training samples are covered by the ellipsoids in the population or the specified maximum number of iterations is exceeded. The final solution obtained by our hybrid GA procedure is not the final population itself but the best ellipsoids in the final population, which cover all the training samples contained by the final population. The selection of the best ellipsoids in the final population for the final output of the algorithm can eliminate the redundant ellipsoids whose removal does not change the recognition capability of the classifier.

3.3 Evolution of the feature subspaces

The feature subspaces evolve on the basis of the performance of the regional ellipsoid classifiers they contain in their regions. The evolution procedure for the feature subspace population is composed of two phases, i.e., creating new individual feature subspaces and updating the current population. A feature subspace is considered to be evolved when the regional agents in the subspace have one of more positive examples included in other feature subspaces. This means that the feature subspace seeks for opportunities to expand to include the corresponding examples by adopting the features of the targeted examples that are not yet included in the subspace

The newly obtained feature subset is inserted into the feature subspace population if any of its positive examples are not included in other feature subspaces. The insertion of new subspaces activates a deletion process that finds and deletes subspaces that are enclosed by those new subspaces.

3.4 Experimental Results

We applied the proposed methods to three data sets both to introduce a simple example of ellipsoids adaptation result and to verify the effectiveness of our methods. The data sets are available from the UCI Machine Learning Repository (Blake and C.J. Merz 1998). Each example of the data sets has continuous variables. As a preprocessing of the data for our classifier system, the value of each variable is normalized as having the maximum value of one and the minimum value of zero.

3.4.1 Illustrative Example

In order to demonstrate the applicability of the proposed adaptation method, the iris data is taken as an example. The iris data consist of 150 examples with four featues (sepal length, sepal width, petal length and petal width) and three types of iris plant, i.e., three classes (iris setosa, iris versicolour and iris virsinica). There are 50 examples in each class and the statistical summary of four variables is shown in table 1.

Table 1: Statistical summary of variables in the iris data

Feature	Min	Max	Mean	SD*	Correl**
Sepal length	4.3	7.9	5.84	0.83	0.7826
Sepal width	2.0	4.4	3.05	0.43	-0.4194
Petal length	1.0	6.9	3.76	1.76	0.9490
Petal width	0.1	2.5	1.20	0.76	0.9565

* SD: Standard Deviation
** Correl: Correlation between class and corresponding feature

The proposed coevolutionary procedure of feature subsets and ellipsoids was conducted in four-dimensional space made by normalization of the values of the two selected variables into 0 to 1. The best four ellipsoids that cover the maximum number of examples are evolved from feature subsets formed by two features of petal length and petal width. Accordingly, the best ellipsoids are defined in two-dimensional space with the two features, petal length and petal width, which have the relatively high correlation values between class and variables as shown in Table 1. The final result of the ellipsoids adaptation procedure is shown in Figure 6, where four ellipsoids are constructed to cover 149 examples. One example could not be covered as the result of eliminating ellipsoids that cover only one example to avoid the overfitting problem.

From the example of iris data, we can see that the feature subspace determination can choose highly correlated variables with classes and the proposed hybrid GA method can fit ellipsoids to the data patterns in the dimensional space defined by the chosen feature space. In the following section, the comparative evaluation is conducted for more complicated classification problems.

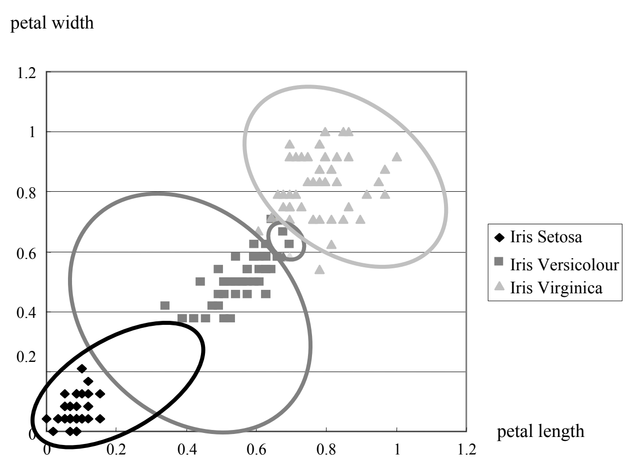


Figure 6: A set of ellipses generated from the hybrid GA procedure for the iris data

3.4.2 Data sets used for performance evaluation

The performance of the proposed hybrid GA procedure was evaluated on two data sets which are usually employed as benchmarks for classification applications (i.e., glass, and ionosphere data). The data sets were normalized so that continuous variables ranged from [0, 1], and then each data set was partitioned into training set and test set. The summary of the data sets is described in Table 2 and the detailed explanation is presented in the following subsections.

Table 2: Summary of data sets used for evaluation

Data set	# features	# classes	# Instances
Glass	9	6	214
Ionosphere	33	2	351

3.4.2.1 Glass data

The glass data set consists of 214 examples with nine continuous features from six classes, which classifies the types of glasses based on the mixture ratio of constituents. This data set is available from UCI database. We evaluated the performances for test data using a random sub-sampling technique. In the computer simulations, 2/3 of the given examples were independently and randomly selected for training data and the other 1/3 were used as test data. The random split of the given data was iterated 15 times.

3.4.2.2 Ionosphere data

The proposed method is applied to the ionosphere data that includes 351 examples with 33 continuous features and 2 classes, which determines “good” or “bad” one by the information of the received returns from the ionosphere through 16-arrayed antenna. Since this data set has many features, we can evaluate the effectiveness of the proposed method combined with automated feature selection by using the data. The ten-fold cross validation (10-fold CV) was used for evaluating the performance of our classification method.

3.4.3 Results of computational evaluation

Here we present the results achieved by the proposed hybrid GA approach and compare them with the performances of existing well-known classifiers, i.e., a *k* nearest neighbor (Weiss and Kulikowski 1991), a decision tree with C4.5 (Quinlan 1993), and a neural network with backpropagation (Haykin 1994). We tried to guarantee the proper prediction power of the classifiers even under insufficient training data for scarce classes by adopting rather simple structures such as *k*=3 and 1 hidden layer. Table 3 shows the comparison results of the average classification rate for the three data sets mentioned in the previous subsection. We can see that the proposed hybrid GA classification method with ellipsoidal regions achieved a superior classification rate of test data in comparison with other popular methods such as the *k* nearest neighbor, the decision tree, and the neural network.

Table 3. Comparison results in terms of classification rate

Classifier	**Classification rate (%)**	
	Glass	Ionosphere
k Nearest Neighbor ($k = 3$)	63.74	84.90
C4.5	65.70	88.89
Neural Network	63.36	90.03
This study	67.10	90.88

4 CONCLUSION

This paper proposes a coevolution-based classification method for multidimensional pattern classification problems. The method consists of two layers. Feature sets, pairs of a feature variable and its range, determine the sub-regions where they apply. For each sub-region, a pool of ellipsoids is developed to fit the data patterns in the training examples. The ellipsoids are subject to the inner loop of adaptation process whereas the evolution of the feature sets forms the outer loop.

The proposed representation of ellipsoids, whose parameters are two foci and a size factor, has the advantage of interpretability, tractability and robust generalization ability. The GA procedure to fit the ellipsoids to the data patterns is expedited by a few common adaptive operations: expansion, avoidance, and move. The feature-ellipsoid coevolution allows robust performance in problems with a large number of features

The proposed coevolutionary classification method was applied to well-known data sets with various numbers of continuous variables (i.e., 4 to 33 features). The performance results showed the superiority of the proposed method to the existing methods regardless of the number of the features.

ACKNOWLEDGMENTS

This research was supported by the Korean Ministry of Science & Technology.

REFERENCES

Abe, S. and R. Thawonmas. 1997. A fuzzy classifier with ellipsoidal regions. *IEEE Transactions on Fuzzy Systems* 5 (3): 358-368.

Blake, C.L. and C.J. Merz. 1998. UCI Repository of machine learning databases. Available online via <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

Eshelman, L.J. 1991. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In *Foundations of genetic algorithms*, ed. G.J.E. Rawlins, 265-283. San Mateo, CA: Morgan Kaufman.

Haykin, S. 1994. *Neural networks: a comprehensive foundation*. Reading, New Jersey: Prentice-Hall.

Jain, A. K., R.P.W. Duin, and J. Mao. 2000. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (1): 4-37.

Liu, H. and R. Setiono. 1998. Incremental feature selection. *Applied Intelligence* 9: 217-230.

Nolan, J.R. 2002. Computer systems that learn: an empirical study of the effect of noise on the performance of three classification methods. *Expert Systems with Applications* 23: 39-47.

Pal, S.K., S. Bandyopadhyay, and C.A. Murthy. 1998. Genetic algorithms for generation of class boundaries. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* 28 (6): 816-828.

Quinlan, J. R. 1993. *C4.5: Programs for machine learning.* Reading, San Mateo, CA: Morgan Kaufmann.

Simpson, P.K. 1992. Fuzzy min-max neural networks-part1: classification. *IEEE Transactions on Neural Networks* 3 (5): 776-786.

Weiss, S.M. and C.A. Kulikowski. 1991. *Computer systems that learn*. Reading, CA: Morgan Kaufmann.

AUTHOR BIOGRAPHIES

KI-KWANG LEE is an engineer of System Lab. in Information and Telecommunication Research Center at LG Electronic, Inc., Korea. He also a PhD student of the Department of Industrial Engineering at Korea Advanced Institute of Science and Technology (KAIST). He received his BSc in 1994 from Han Yang University, MSc in 1996 from KAIST. His research interests include intelligent systems and data mining. His email address is <kiklee @naver.com>

WAN CHUL YOON is Professor of the Department of Industrial Engineering at Korea Advanced Institute of Science and Technology (KAIST), Korea. He received his BSc in 1977 from Seoul National University, MSc in 1979 from KAIST, and PhD in 1987 in industrial and systems engineering from Georgia Institute of Technology. His research interests include application of artificial intelligence, human decision-making and aiding, information systems, and joint intelligent systems. His email address is <wcyoon@kaist.ac.kr>