2009 Ninth International Conference on Intelligent Systems Design and Applications

A First Approach to Nearest Hyperrectangle Selection by Evolutionary Algorithms

Salvador García
Dept. of Computer Science
University of Jaén
23071, Jaén, Spain
sglopez@ujaen.es

Joaquín Derrac, Julián Luengo, Francisco Herrera
Dept. of Computer Science and A.I.
University of Granada
18071, Granada, Spain
{jderrac, julianlm, herrera}@decsai.ugr.es

***Abstract*—The nested generalized exemplar theory accomplishes learning by storing objects in Euclidean n-space, as hyperrectangles. Classification of new data is performed by computing their distance to the nearest "generalized exemplar" or hyperrectangle. This learning method permits to combine the distance-based classification with the axis-parallel rectangle representation employed in most of the rule-learning systems. This contribution proposes the use of evolutionary algorithms to select the most influential hyperrectangles to obtain accurate and simple models in classification tasks. The proposal is compared with the most representative nearest hyperrectangle learning approaches and the results obtained show that the evolutionary proposal outperforms them in accuracy and requires storing a lower number of hyperrectangles.**

I. Introduction

One of the possible schemes for learning from examples is based on Nested Generalized Exemplar (NGE) theory. It was introduced in [1] and makes several significant modifications to the exemplar-based learning model. The most important one is that it retains the notion of storing verbatim examples in memory but, it also allows examples to be generalized. In NGE theory, generalizations take the form of hyperrectangles in a Euclidean n-space. The hyperrectangles may be nested one inside another and inner hyperrectangles serve as exceptions to surroundings hyperrectangles. A specific instance can be viewed as a minimal hyperrectangle and hyperrectangle is an axis-parallel rectangle representation employed in most of the rule-learning systems.

Several works argue the benefits of using hyperrectangles together with instances to form the classification rule [2], [3], [4]. With respect to instance-based classification [5], the employment of hyperrectangles increases the comprehension of the data stored to perform classification of unseen data and the achievement of a substantial compression of the data, reducing the storage requirements. Considering rule induction [6], the ability of modeling decision surfaces by hybridizations between distance-based methods (Voronoi diagrams) and parallel axis separators could improve the performance of the models in domains with clusters of exemplars or exemplars strung out along a curve. In addition, NGE learning allows capture generalizations with exceptions.

The methods used for generating nearest hyperrectangles classification work in an incremental fashion, such as EACH [1], or in batch mode (BNGE [2] and RISE [3]). The incremental way is dependant on the order of presentation of examples and offers poor results in standard classification. However, it could be used in on-line learning scenarios. Batch mode methods employ heuristics to determine the choice of the exemplars to be merged or generalized in each stage. The results offered are very interesting and usually outperform the results obtained by the 1-nearest neighbour (1NN) classifier, usually used as baseline method.

The problem of creating an optimal number of hyperrectangles to classify a set of points is NP-hard but a finite subset of relevant hyperrectangles can be easily obtained following a heuristic algorithm. Most of the hyperrectangles obtained could be irrelevant and this set of hyperrectangles is thus suitable to be improved by a reduction technique [7] or prototype selection method [8]. Evolutionary Algorithms (EAs) [9] have been used for data reduction with promising results. They have been successfully used for feature selection [10], [11], [12] and instance selection [13], [14], [8].

In this contribution, we propose the use of EAs for hyperrectangles selection in classification tasks. One similar approach is SIA [15], which is a genetics-based machine learning method to obtain set of rules allowing to compute distances among rules. Our objective is to increase the accuracy of this type of representation by means of selecting the best suitable set of hyperrectangles which optimizes the nearest hyperrectangle classification rule. We compare our approach with BNGE and RISE. The empirical study has been contrasted via non-parametrical statistical testing [16], [17], the results show an improvement of accuracy whereas the number of hyperrectangles stored in the final subset is much lower.

To achieve this objective, the rest of the contribution is organized as follows: Section II gives an explanation about hyperrectangle learning. In Section III, the evolutionary algorithm used to tackle this problem is explained. In Section IV the experimentation framework and the results and their analysis are presented. Finally, in Section V, we point out our conclusion.

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.238

II. LEARNING WITH HYPERRECTANGLES

This section provides an overview on learning with hyperrectangles. First, we explain the needed concepts to understand the classification rule followed by this type of methods (Subsection II-A). After this, the two main proposals of hyperrectangle learning will be briefly described, BNGE in Subsection II-B and RISE in Subsection II-C.

A. Matching and Classification

The matching process in one of the central features in hyperrectangle learning and it allows some customization, if desired. This process computes the distance between a new example and an exemplar memory object (a hyperrectangle). For remainder of this contribution, we will refer to the example to be classified as E and the hyperrectangle as H, independently of H is formed by an unique instance or it has some volume.

The model computes a match score between E and H by measuring the Euclidean distance between two objects. The Euclidean distance is well-known when H is a single point. In case contrary, the distance is computed as follows (numeric attributes):

$$D_{EH} = \sqrt{\sum_{i=1}^{M} \left(\frac{dif_i}{max_i - min_i} \right)^2}$$

where

$$dif_i = \begin{cases} E_{f_i} - H_{upper} \text{ when } E_{f_i} > H_{upper} \\ H_{lower} - E_{f_i} \text{ when } E_{f_i} < H_{lower} \\ 0 \text{ otherwise} \end{cases}$$

M is the number of attributes of the data, E_{f_i} is the value of the ith feature of the example, H_{upper} and H_{lower} are the upper and lower values of H for a specific attribute and max_i and min_i are the maximum and minimum values for ith feature in training data, respectively.

The distance measured by this formula is equivalent to the length of a line dropped perpendicularly from the point E_{f_i} to the nearest surface, edge or corner of H. Note that points internal to a hyperrectangle have distance 0 to that rectangle. In the case of overlapping rectangles, a point falling in the area of overlap belongs to the smaller rectangle. The size of a hyperrectangle is defined in terms of volume. In nominal attributes, if the features are equal, the distance is zero, else it is one.

NGE theory also refers to weights associated to features in examples but they are not considered in this contribution, because they can be used independently to enhance the performance of this type of learners, as [2] showed.

B. BNGE: Batch Nested Generalized Exemplar

BNGE is a batch version of NGE (also known as EACH [1]) and it is proposed to alleviate some drawbacks presented in initial NGE [2]. It changes its incremental fashion to a batch mode and adds some modifications in the matching rule, such as including all possible nominal values in hyperrectangle definition and adding a mechanism to deal with missing values. The generalization of a hyperrectangle is done by expanding its frontiers just to cover the desired example.

BNGE only merges hyperrectangles if the new generalized hyperrectangle dos not cover (or overlap with) any hyperrectangles from any other classes. It does not permit overlapping or nesting, which are two of the identified disadvantages of the basic NGE.

C. RISE: Unifying Instance-Based and Rule-Based Induction

RISE [3] is an approach proposed to overcome some of the limitations of instance-based learning and rule induction by unifying the two. It follows similar guidelines explained above, but it furthermore introduces some improvements regarding distance computations, since the SVDM distance [18] is used in nominal attributes. RISE selects the rule with the highest accuracy (using the Laplace correction used by many existing rule-induction techniques [6]) instead of choosing the smallest rule that covers the example.

BNGE and RISE follow a similar mechanism to produce hyperrectangles. They start from the complete training set and try to merge the nearest examples/hyperrectangles if the global accuracy is not decreased. RISE uses a leave-one-out methodology to compute training accuracy and no avoidance of nesting or overlapping between hyperrectangles is used as well.

III. EVOLUTIONARY SELECTION OF HYPERRECTANGLES

The approach proposed in this contribution, named Evolutionary Hyperrectangle Selection by CHC (EHS-CHC), is fully explained in this section. First, we introduce the CHC model used as EA to perform hyperrectangle selection in Subsection III-A. After this, the specific issues regarding representation and fitness function complete the description of the proposal in Subsection III-B.

A. CHC Model

As evolutionary computation method, we have used the CHC model [19], [14]. CHC is a classical evolutionary model that introduces different features to obtain a trade-off between exploration and exploitation; such as incest prevention, reinitialization of the search process when it becomes blocked and the competition among parents and offspring into the replacement process.

During each generation the CHC develops the following steps.

- It uses a parent population of size N to generate an intermediate population of N individuals, which are randomly paired and used to generate N potential offspring.
- Then, a survival competition is held where the best N chromosomes from the parent and offspring populations are selected to form the next generation.

CHC also implements a form of heterogeneous recombination using HUX, a special recombination operator. HUX exchanges half of the bits that differ between parents, where the bit position to be exchanged is randomly determined. CHC also employs a method of incest prevention. Before applying HUX to the two parents, the Hamming distance between them is measured. Only those parents who differ from each other by some number of bits (mating threshold) are mated. The initial threshold is set at $L/4$, where L is the length of the chromosomes. If no offspring are inserted into the new population then the threshold is reduced by one.

No mutation is applied during the recombination phase. Instead, when the population converges or the search stops making progress (i.e., the difference threshold has dropped to zero and no new offspring are being generated which are better than any member of the parent population) the population is reinitialized to introduce new diversity to the search. The chromosome representing the best solution found over the course of the search is used as a template to reseed the population. Reseeding of the population is accomplished by randomly changing 35% of the bits in the template chromosome to form each of the other $N-1$ new chromosomes in the population. The search is then resumed.

B. Representation and Fitness Function

Let us assume that there is a training set TR with P instances which consists of pairs $(x_i, y_i), i = 1, ..., P$, where x_i defines an input vector of attributes and y_i defines the corresponding class label. Each one of the P instances has M input attributes. Let us also assume that there is a hyperrectangle set HS with N hyperrectangles which consists of pairs $(H_i, y_i), i = 1, ..., N$, where H_i defines set of conditions $(A_1, A_2, ..., A_M)$ and y_i defines the corresponding class label. Each one of the N hyperrectangles has M conditions which can be numeric conditions, expressed in terms of minimum and maximum values in interval $[0, 1]$; or they can be categorical conditions, by using set of possible values $A_i = \{v_{1i}, v_{2i}, ..., v_{vi}\}$, where v_{ji} denotes all possible nominal values for attribute i, assuming that ir has v different values. Note that we make no distinction between a hyperrectangle with volume and minimal hyperrectangles formed by isolated points. Let $S \subseteq HS$ be the subset of selected hyperrectangles resulted in the run of a hyperrectangle selection algorithm.

Hyperrectangle selection can be considered as a search problem in which EAs can be applied. We take into account two important issues: the specification of the representation of the solutions and the definition of the fitness function.

- *Representation:* The search space associated is constituted by all the subsets of HS. This is accomplished by using a binary representation. A chromosome consists of N genes (one for each hyperrectangle in HS) with two possible states: 0 and 1. If the gene is 1, its associated hyperrectangle is included in the subset of HS represented by the chromosome. If it is 0, this does not occur.
- *Fitness Function:* Let S be a subset of hyperrectangles of HS and be coded by a chromosome. We define a fitness function based on the accuracy (classification rate) evaluated over TR through the rule described in Section II-A.

$$Fitness(S) = \alpha \cdot clas\_rat + (1 - \alpha) \cdot perc\_red.$$

$clas\_rat$ denotes the percentage of correctly classified objects from TR using S. per_red is defined as

$$perc\_red = 100 \cdot \frac{|HS| - |S|}{|HS|}$$

The objective of the EAs is to maximize the fitness function defined, i.e., maximize the classification rate and minimize the number of hyperrectangles selected. We preserve the value of $\alpha = 0.5$ used in previous works related to instance selection [13], [14], [8].

The same mechanisms to perform a classification of a unseen example exposed in [1] are used in our approach. In short, they are:

  - If no hyperrectangle covers the example, the class of the nearest hyperrectangle defines the prediction.
  - If various hyperrectangles cover the example, the one with lowest volume is the chosen to predict the class, allowing exceptions within generalizations.

Our approach computes the volume of a hyperrectangle by the following way:

$$V_H = \prod_{i}^{M} L_i$$

where L_i is computed for each condition as

$$L_i = \begin{cases} H_{upper} - H_{lower} \\ \text{ if numeric and } H_{upper} \neq H_{lower} \\ 1 \text{ if numeric and } H_{upper} = H_{lower} \\ \frac{\text{num. values selected}}{v} \text{ if nominal} \end{cases}$$

- There is a detail not specified yet. It refers to the building of the initial set of hyperrectangles. In this first approach, we have used a heuristic which is fast and obtain acceptable results. The heuristic yields a

Table I
SUMMARY DESCRIPTION FOR CLASSIFICATION DATA SETS

Data Set	#Ex.	#Atts.	#Num.	#Nom.	#Cl.	Data Set	#Ex.	#Atts.	#Num.	#Nom.	#Cl.
australian	690	14	8	6	2	iris	150	4	4	0	3
breast	286	9	0	9	2	led7digit	500	7	0	7	10
bupa	345	6	6	0	2	lymphography	148	18	3	15	4
cleveland	297	13	13	0	5	newthyroid	215	5	5	0	3
contraceptive	1,473	9	6	3	3	pima	768	8	8	0	2
crx	125	15	6	9	2	sonar	208	60	60	0	2
ecoli	336	7	7	0	8	wine	178	13	13	0	3
glass	214	9	9	0	7	wisconsin	683	9	9	0	2
haberman	306	3	3	0	2	zoo	101	17	0	17	7

hyperrectangle from each examples in the training set. For each one, it finds the $K-1$ nearest neighbours being the Kth neighbour an example of different class. Then each hyperrectangle is built getting the minimal an maximal values (in case of numerical attributes) to represent the interval in such attribute or getting all the different categorical values (in case of nominal attributes) of all the examples belonging to its set of $K-1$ neighbours. Once all the hyperrectangles are obtained, the duplicated ones are removed (keeping one representant in each case), hence $|HS| \leq |TR|$. Note that point hyperrectangles are possible to be obtained using this heuristic.

IV. EXPERIMENTAL FRAMEWORK AND RESULTS

This section describes the methodology followed in the experimental study of the hyperrectangles learning techniques compared. We will explain the configuration of the experiment: used data sets and parameters for the algorithms. The algorithms used in the comparison are: 1NN [5], BNGE [2] and RISE [3].

Table II
PARAMETERS CONSIDERED FOR THE ALGORITHMS.

Algorithm	Parameters
BNGE	It has not parameters to be fixed
RISE	$Q=1,\ S=2$
EHS-CHC	$Pop=50,\ Eval=10000,\ \alpha=0.5$

A. Experimental Framework

Performance of the algorithms is analyzed by using 18 data sets taken from the UCI Machine Learning Database Repository [20]. The main characteristics of these data sets are summarized in Table I. For each data set, it shows the number of examples (#Ex.), number of attributes (#Atts.), number of numerical attributes (#Num.), number of nominal attributes (#Nom.) and number of classes (#Cl.).

The data sets considered are partitioned using the *ten fold cross-validation (10-fcv)* procedure. The parameters of the used algorithms are presented in Table II.

B. Results and Analysis

Table III shows the results in test data obtained by the algorithms compared using the accuracy measure. The best case in each data set is stressed in bold.

Table III
ACCURACY OBTAINED BY THE HYPERRECTANGLE LEARNING METHODS STUDIED

dataset	1NN	BNGE	RISE	EHS-CHC
australian	0.8145	0.8464	0.8058	**0.8478**
bre	0.6535	0.6327	0.6710	**0.7346**
bupa	0.6108	**0.6481**	0.6468	0.6167
cleveland	0.5314	**0.5837**	0.4919	0.5583
contraceptive	0.4277	0.4733	0.4494	**0.4983**
crx	0.7957	0.8391	0.8159	**0.8464**
ecoli	0.8070	**0.8302**	0.7621	0.7948
glass	**0.7361**	0.6654	0.6946	0.6287
haberman	0.6697	0.6862	0.6405	**0.7122**
iris	0.9333	**0.9600**	0.9400	0.9267
led7digit	0.4020	0.6260	0.6520	**0.6820**
lym	0.7387	0.7996	0.7612	**0.8334**
newthyroid	**0.9723**	0.9537	0.9580	0.9632
pima	0.7033	0.7318	0.6432	**0.7384**
sonar	**0.8555**	0.6021	0.7690	0.7650
wine	0.9552	**0.9660**	0.9438	0.9490
wisconsin	0.9557	**0.9628**	0.9456	0.9599
zoo	0.9281	**0.9683**	**0.9683**	0.9300
AVERAGE	0.7495	0.7653	0.7533	**0.7770**

Table IV shows the average number of hyperrectangles (or rules) obtained by each one of the methods considered (obviously, excluding 1NN).

From Tables III and IV, the following aspects can be pointed out:

- EHS-CHC proposal obtains the best average result in accuracy. It outperforms the other hyperrectangle learning methods (BNGE and RISE) and 1NN.
- We have to remark that EHS-CHC obtained the best accuracy rates in 8 of 18 data sets, 6 of which present nominal attributes. Note that in 6 of 7 data sets (all except *zoo* data set) with nominal values, EHS-CHC outperforms the rest of methods. This behaviour could be very interesting when dealing with hybrids data sets that contains both numeric and categorical information.
- The number of hyperrectangles needed by EHS-CHC to achieve such accuracy rates is much lower than the needed by BNGE and RISE.

Table IV
AVERAGE NUMBER OF HYPERRECTANGLES OBTAINED BY THE HYPERRECTANGLE LEARNING METHODS STUDIED

dataset	BNGE	RISE	EHS-CHC
australian	232.3	226	**5.7**
bre	82.1	208.7	**4.9**
bupa	184.3	201.5	**9.8**
cleveland	125.9	161.7	**5**
contraceptive	1208	1018.4	**12.7**
crx	106.9	286.9	**6.1**
ecoli	88.6	169.2	**11.1**
glass	78.5	92.2	**12.2**
haberman	212.4	132.4	**4.4**
iris	12	37.1	**3.4**
led7digit	403	279	**10.5**
lym	29.9	81	**6.2**
newthyroid	17.3	33.9	**4.7**
pima	326.9	436.9	**11**
sonar	64.3	59.1	**10.3**
wine	10.9	29.7	**3.6**
wisconsin	66.3	182.9	**3.8**
zoo	8.9	25.6	**5.6**
AVERAGE	181.03	203.46	**7.28**

- In some data set, the number of hyperrectangles selected by EHS-CHC is a little higher or equal to the number of classes, indicating us that it is able to learn general concepts, in terms of rules, which collect the most relevant examples of each class. It is the case of *cleveland*, *iris*, *led7digit*, *wine* and *zoo* data sets.

We have included a second type of table accomplishing a statistical comparison of methods over multiple data sets. It is recommended a set of simple, safe and robust non-parametric tests for statistical comparisons of classifiers [16], [17]. One of them is Wilcoxon Signed-Ranks Test [21]. Table V collects results of applying Wilcoxon's test between our proposed methods and the rest of algorithms studied in this contribution over the 18 data sets considered. This table is divided into two parts:

- In the first part, the measure of performance used is the accuracy classification in test set.
- In the second part, we accomplish Wilcoxon's test by using as performance measure the number of hyperrectangles yielded.

Each part of this table contains one column, representing our proposed methods, and N_a rows where N_a is the number of algorithms considered in this study. In each one of the cells can appear three symbols: +, = or -. They represent that the proposal outperforms (+), is similar to (=) or is worse (-) in performance than the algorithm which appears in the column (Table V). The value in brackets is the p-value obtained in the comparison and the level of significance considered is $\alpha = 0.05$.

We make a brief analysis of results summarized in Table V:

- The use of Wilcoxon's test confirms the improvement caused by EHS-CHC over RISE. However, there is no statistical evidence of the improvement over BNGE and 1NN, although the behaviour in accuracy between our proposal and each one of them is similar.
- The number of hyperrectangles obtained by EHS-CHC is clearly much inferior than the obtained by BNGE and RISE.
- With similar performance in accuracy but with a much lower quantity of hyperrectangles, EHS-CHC can be considered a competitive approach in hyperrectangle learning.

Table V
WILCOXON'S TEST RESULTS OVER ACCURACY AND NUMBER OF HYPERRECTANGLES OBTAINED

algorithm	EHS-CHC *Accuracy*	EHS-CHC num. hyperrec.
1NN	= (.170)	—
BNGE	= (.777)	+ (.001)
RISE	+ (.031)	+ (.001)

V. CONCLUSIONS

The purpose of this contribution is to present a proposal of Evolutionary Hyperrectangle selection for hyperrectangle learning based on NGE theory. The results show that our proposal obtains very accurate models with a low number of hyperrectangles. The accuracy of the approach is very competitive with respect to classical hyperrectangle learning methods, such as BNGE and RISE, and 1-nearest neighbbour classifier. The interpretability of the models obtained is significantly increased.

REFERENCES

[1] S. Salzberg, "A nearest hyperrectangle method," *Machine Learning*, vol. 6, pp. 151–276, 1991.

[2] D. Wettschereck and T. G. Dietterich, "An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms," *Machine Learning*, vol. 19, pp. 5–27, 1995.

[3] P. Domingos, "Unifying instance-based and rule-based induction," *Machine Learning*, vol. 24, pp. 141–168, 1996.

[4] O. Luaces and A. Bahamonde, "Inflating examples to obtain rules," *International Journal of Intelligent Systems*, vol. 18, no. 11, pp. 1113–1143, 2003.

[5] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, no. 1, pp. 37–66, 1991.

[6] J. Fürnkranz, "Separate-and-conquer rule learning," *Artificial Intelligence Review*, vol. 13, no. 1, pp. 3–54, 1999.

[7] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," *Machine Learning*, vol. 38, no. 3, pp. 257–286, 2000.

[8] S. García, J. R. Cano, and F. Herrera, "A memetic algorithm for evolutionary prototype selection: A scaling up approach." *Pattern Recognition*, vol. 41, no. 8, pp. 2693–2709, 2008.

[9] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Springer-Verlag, 2003.

[10] D. Whitley, R. Beveridge, C. Guerra, and C. Graves, "Messy genetic algorithms for subset feature selection." in *Proceedings of the International Conference on Genetic Algorithms*, 1998, pp. 568–575.

[11] C. Guerra-Salcedo, S. Chen, D. Whitley, and S. Smith, "Fast and accurate feature selection using hybrid genetic strategies." in *CEC*, 1999, pp. 177–184.

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

[13] J. R. Cano, F. Herrera, and M. Lozano, "Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study." *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 6, pp. 561–575, 2003.

[14] ——, "Evolutionary stratified training set selection for extracting classification rules with trade-off precision-interpretability." *Data and Knowledge Engineering*, vol. 60, pp. 90–108, 2007.

[15] G. Venturini, "SIA: A supervised inductive algorithm with genetic search for learning attributes based concepts," in *ECML*, 1993, pp. 280–296.

[16] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Rechearch*, vol. 7, pp. 1–30, 2006.

[17] S. García and F. Herrera, "An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons," *Journal of Machine Learning Research*, vol. 9, pp. 2677–2694, 2008.

[18] D. R. Wilson and T. R. Martinez, "Improved heterogeneous distance functions," *Journal of Artificial Intelligence Research*, vol. 6, pp. 1–34, 1997.

[19] L. J. Eshelman, "The CHC adaptive search algorithm: How to safe search when engaging in nontraditional genetic recombination." in *Foundations of genetic algorithms*, G. J. E. Rawlings, Ed., 1991, pp. 265–283.

[20] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html

[21] D. Sheskin, *Handbook of parametric and nonparametric statistical procedures.* Chapman & Hall/CRC, 2006.