A First Study on the Use of Coevolutionary Algorithms for Instance and Feature Selection

Joaquín Derrac[1], Salvador García[2], and Francisco Herrera[1]

[1] University of Granada, Department of Computer Science and Artificial Intelligence, 18071 Granada, Spain
jderrac@correo.ugr.es, herrera@decsai.ugr.es
[2] University of Jaén, Department of Computer Science, 23071 Jaén, Spain
sglopez@ujaen.es

Abstract. Cooperative Coevolution is a technique in the area of Evolutionary Computation. It has been applied to many combinatorial problems with great success. This contribution proposes a Cooperative Coevolution model for simultaneous performing some data reduction processes in classification with nearest neighbours methods through feature and instance selection.

In order to check its performance, we have compared the proposal with other evolutionary approaches for performing data reduction. Results have been analyzed and contrasted by using non-parametric statistical tests, finally showing that the proposed model outperforms the non-cooperative evolutionary techniques.

1 Introduction

One main process in data mining is the one known as data reduction [18]. In classification, it aims to reduce the size of the training set mainly to increase the efficiency of the training phase (by removing redundant instances) and even to reduce the classification error rate (by removing noisy instances).

Instance Selection (IS) and Feature Selection (FS) are two of the most known data reduction techniques in data mining. Both are really effective not only to reduce the size of the train set, but also to filtrate and clean noisy data, thus helping classifiers to improve its accuracy [12,13].

Evolutionary Algorithms (EAs)[5] are general purpose search algorithms that use principles inspired by nature to evolve solutions to problems. EAs have been successfully used in data mining problems[8,9]. Their capacity of tackling IS and FS as combinatorial problems is specially useful [2,14].

Coevolution is a specialized trend of EAs. It tries to simultaneously manage two or more populations (also called species), to evolve them and to allow interactions among individuals of any population. The goal is to improve results achieved from each population separately. The Coevolution model has shown some interesting characteristics in the last years [22]. Also, it has been successfully applied in other problems, like function optimization [11,21].

E. Corchado et al. (Eds.): HAIS 2009, LNAI 5572, pp. 557–564, 2009.

Our proposal combines Evolutionary IS and FS with Coevolution techniques, in order to improve the effectiveness of Evolutionary IS and FS applied to nearest neighbours classifiers in terms of accuracy. We have named our proposed model CoCHC (Cooperative Coevolution model using CHC algorithm). A wide range of classification data sets will be used to compare it with other non-coevolutionary models, in order to highlight the benefits of the use of Coevolution.

The rest of this contribution is organized as follow: Section 2 reviews the preliminary theoretical study. Section 3 explains the Cooperative Coevolutive model proposed. Section 4 describes the experimental framework used and presents the analysis of results. Finally, in Section 5, we point out the conclusions achieved.

2 Background and Related Work

This section shows the main topics of the background in which our contribution is based. Section 2.1 describes some evolutionary techniques applied to IS and FS problems. Section 2.2 shows the EAs in which our model is based. Finally, Section 2.3 highlights the main characteristics of Cooperative Coevolution.

2.1 Evolutionary Instance and Feature Selection

EAs have proved to be good mechanisms for data reduction in data mining. They have been widely used to tackle the FS and IS problems.

The FS problem can be defined as a search process of P features from an initial set of M variables, with $P <= M$. It aims to eliminate irrelevant and/or redundant features and to obtain a simpler classification system. Also, this reduction can improve the accuracy of the model in classification [13].

The IS problem can also be defined as a search process, where a reduced set S of instances is selected from the training set. By choosing the most suitable points in the data set as instances for the training data, the classification process can gets greatly increased both its efficiency and accuracy [12]

In [14] is proposed a hybridization of a genetic algorithm with local search operators for FS. In [2], a complete study of the use of EAs in IS is done, highlighting four EAs to complete this task: Generational Genetic Algorithm (GGA) [10], Steady-State Genetic Algorithm (SGA) [20], CHC Adaptive Search Algorithm(CHC) [6] and Population-Based Incremental Learning (PBIL) [1]. They concluded that EAs outperform classical algorithms both in reduction rates and classification accuracy. They also concluded that CHC is the most appropriate EA to make this task, according to the algorithms they compared.

Beyond these applications, it is important to point out that both techniques can be applied simultaneously. Despite the most natural way to combine these techniques is to use one first (i.e IS), to get its results and to apply them to the second technique (i.e FS), some authors have already tried to get some profit from the joint use of both approaches [7].

2.2 CHC Algorithm

As it is exposed in the previous section, CHC is a good example of EA which can be used in IS and FS. We have studied its main characteristics to select it as the baseline EA which will guide the search process of our model (it will be explained in Section 3). During each generation, the CHC algorithm [6] develops the following steps:

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

CHC also implements HUX recombination operator. HUX exchanges half of the bits that differ between parents, where the bit position to be exchanged is randomly determined. It also employs a method of incest prevention: Before applying HUX to 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. If no offspring is inserted into the new population then the threshold is reduced.

No mutation is applied during the recombination phase. Instead, when the search stops making progress the population is reinitialized to introduce new diversity. The chromosome representing the best solution found is used as a template to re-seed the population, randomly changing 35% of the bits in the template chromosome to form each of the other chromosomes in the population.

We have selected CHC because it has been widely studied, being now a well-known algorithm on evolutionary computation. Furthermore, previous studies like [2] support the fact that it can perform well on data reduction problems.

2.3 Cooperative Coevolution

In the context of evolutionary computation, cooperative coevolution can be defined as the co-existence of some interacting populations, evolving simultaneously. Each population evolves individuals representing a component of the final solution. Thus, a full candidate solution is formed by joining an individual chosen from each population [17].

In the underlying evolutionary search procedure, a special fitness function is used. To evaluate an individual, there must be selected one member from the other population (collaborators). The merge of all collaborators will produce a full solution, which can be evaluated by the fitness function.

There are some different proposals of the process of choosing the collaborators. One way, is to evaluate an individual against every single collaborator in the other population [15]. Although it would be a best way to select the collaborators, it will consume a very high number of evaluations in the computation of the fitness function. To reduce this number, there are other choices as the use of just a random individual or the use of the best individual of last generation [16]. The model proposed in the next section will use this last scheme.

3 Cooperative Coevolutive Model Based on Instance and Feature Selection Using CHC

CoCHC employs three populations which simultaneously coexist. They cooperate to get the best possible solution through the evolutionary search procedure. Each population is focused on one reduction data task:

– The first population performs an instance selection.
– The second population performs a feature selection.
– The third population performs both instance and feature selections.

Algorithm 1 shows a basic pseudocode of the model proposed.

Algorithm 1. CoCHC algorithm basic structure

```
1  Generate ISPopulation,FSPopulation and IFSPopulation Randomly;
2  Select initial bestISArray, bestFSArray and bestIFSArray;
3  Evaluate all populations in the multiclassifier;
4  Select bestISArray, bestFSArray and bestIFSArray from each population;
5  while evaluations < max_evaluations do
6      Select best classifier in last generation;
7      Generate simple classifier output from best individuals of the last
       generation;
8      Do a CHC Generation on every population;
9      Update bestISArray, bestFSArray and bestIFSArray if a better global
       solution has been found;
10 end
   Output: bestISArray, bestFSArray and bestIFSArray
```

Instruction 1 generates the initial random populations. Instruction 2 evaluates all chromosomes by using simple classifiers (see *Multiclassifier structure* paragraph below), and selects the best individual of each population. Instruction 3 evaluates all chromosomes by using the complete multiclassifier, and instruction 4 selects the new best individual of each population.

In instruction 5 the evolutionary process starts. Instruction 6 selects the best classifier of the last generation (the best simple classifier in accuracy). This will help to break ties in the fitness function evaluation. In Instruction 7, the outputs of the simple classifiers from the best individuals of the last generation are saved.

Instruction 8 performs a CHC generation on each population (see Section 2.2). Instruction 9 updates the best global solution if a better solution (concerning one chromosome from each population)have been found.

When a fixed number of evaluations run out, the evolutionary process is finished. Then, best global solution founded is returned.

At this point, we have to describe three important issues to completely describe CoCHC: The specification of the representation of the chromosomes, the structure of the multiclassifier defined by a full solution and the definition of the fitness function.

Representation: Let us assume a data set with N instances and M attributes. Each chromosome consists of a determinate number of genes, which can represent either an instance or a feature. A binary representation is used, thus each gene has two possible states: 1, if the corresponding feature/instance is included in the training set represented by the chromosome, or 0 if not. The concrete representation and size of the chromosome depend of the population which it belongs:

- IS Population: Each gene represents a instance (chromosome size: N).
- FS Population: Each gene represents a feature (chromosome size: M).
- IFS Population: The first N genes of the chromosome represent instances. Remaining genes represents features (chromosome size: $N+M$).

Multiclassifier structure: To evaluate an individual, one member of each of the other populations must be selected. The merge of all collaborators will produce a full solution, which can be evaluated by the fitness function.

Let U be a chromosome of IS Population, let V be a chromosome of FS Population and let W be a chromosome of IFS Population. The multiclassifier structure is defined by three simple classifiers, based on the 1-NN rule. The first classifier only uses the instances defined by U. The second classifier only uses the features defined by V. Finally, the third classifier uses the instances defined by first N genes of W, and the features defined by the last M genes of W.

To get the output of the multiclassifier, the output of the three simple classifiers have to be computed. Then, a final output must be calculated for each instance, by using a majority vote (ties are broken by using the output of the *best classifier* defined by the global model).

Fitness function: Let G be a chromosome of one population. To compute its fitness their collaborators must be found (they are the best individuals from last generation of the other populations). Let F be a binary string composed of G and its collaborators. We define the next fitness function:

$$Fitness(G) = \alpha \cdot \beta \cdot clasRate(F) + (1-\alpha) \cdot IRed(G) + (1-\beta) \cdot FRed(G) \quad (1)$$

Where *clasRate(F)* is the percentage of correctly classified objects from the training set by the multiclassifier defined by F, and *IRed(G)* and *FRed(G)* are the percentage of reduction achieved on instances and features respectively on the baseline classifier defined by G (reduction rates can be computed on the baseline classifier because it is independent for each classifier, thus is not needed to measure the multiclassifier reduction rate). Finally, α and β are parameters valued between $[0, 1]$.

The objective of CoCHC is to maximize the fitness function 1, i.e., to maximize the accuracy and reduction rates of the multiclassifier defined by their best chromosomes.

Before finish this section, it is important to point out that the outputs computed at Instruction 7 make possible that only one simple classifier is needed to be built in every call to fitness function on CHC process, (instead of the three originally required by the fitness function). Thus, the CoCHC model efficiency is greatly increased.

4 Experimental Framework and Results

This section describes the methodology followed in the experimental study conducted in this contribution. Data sets used, parameters of our model and the algorithms used in the comparisons are explained.

4.1 Experimental Framework

To check the performance of CoCHC algorithm, we have used 18 data sets taken from the UCI Machine Learning Database Repository [19]. Table 1 shows their main characteristics. For each data set, it is shown the number of examples, attributes and classes of the problem described.

Table 1. UCI Data sets used in our experiments

Data set	Examples	Attributes	Classes	Data set	Examples	Attributes	Classes
Aut	205	25	6	Housevotes	435	16	2
Bal	625	4	3	Iris	150	4	3
Bands	539	19	2	Mammogr	961	5	2
Bupa	345	6	2	Pima	768	8	2
Car	1728	6	4	Sonar	208	60	2
Cleveland	303	13	5	Tic-tac-toe	958	9	2
Dermat	366	34	6	Vehicle	846	18	4
German	1000	20	2	Wisconsin	699	9	2
Glass	214	9	7	Zoo	101	16	7

The data sets considered are partitioned by using the ten fold cross-validation (10-fcv) procedure. The parameters of CoCHC are: Population size = 50 (for each population), Number of evaluations = 10000, $\alpha = 0.6$, $\beta = 0.98$. The alpha parameter value was taken from the value used on the experiments of [2] (0.5), but slightly increased because the simultaneous use of a FS component. The beta parameter value is near to 1 because in the FS component our model has to remove irrelevant attributes without provoke sudden changes which could decrease the overall accuracy.

Our proposal will be compared with three evolutionary algorithms based on the CHC model, for performing IS, FS and simultaneous IS-FS, respectively. The first one will be denoted by IS-CHC, the second one, FS-CHC; and the last one IFS-CHC. Nearest neighbour rule [3] (1-NN) is also used as a baseline algorithm.

The parameters used for each EA involved in the experimental study are the same as the used by our approach.

4.2 Results

Table 2 shows the average results obtained in test data in terms of accuracy. It also shows the reduction rate achieved in training data. The best results achieved in accuracy for each data set are remarked in bold. Observing Table 2, we can make the following analysis:

– CoCHC achieves the best average result on accuracy.

Table 2. Accuracy obtained in test data

Algorithm	CoCHC		IS-CHC		FS-CHC		IFS-CHC		1-NN
Data set	%Acc.	%Red.	%Acc.	%Red.	%Acc.	%Red.	%Acc.	%Red.	%Acc.
Automobile	79.66	88.72	70.42	91.27	**80.18**	68.00	70.53	98.92	77.43
Bal	88.16	89.28	89.29	98.74	79.04	0.00	**89.43**	98.56	79.04
bands	71.81	82.47	70.14	97.30	71.07	49.47	68.28	99.66	**74.04**
Bupa	68.96	81.96	61.94	96.14	61.93	38.33	**68.98**	99.24	61.08
Car	89.18	70.88	86.69	98.37	**89.58**	18.33	88.66	98.27	85.65
Cleveland	56.76	84.52	**57.81**	97.47	50.83	46.15	57.10	99.46	53.14
Dermatology	95.37	85.60	**97.55**	96.36	95.11	54.71	96.44	99.18	95.35
German	**72.20**	82.80	71.70	98.69	69.30	38.50	71.90	99.89	70.50
Glass	69.99	78.08	69.11	93.14	71.37	43.33	67.06	97.83	**73.61**
Housevotes	**95.14**	89.46	93.32	98.24	94.47	65.00	93.54	99.87	92.16
Iris	**95.33**	81.89	**95.33**	95.56	**95.33**	45.00	94.67	98.11	93.33
Mammographic	**82.00**	97.39	80.23	99.17	72.94	56.00	81.59	99.90	74.72
Pima	72.01	87.73	**76.07**	98.50	68.62	50.00	74.11	99.75	70.33
Sonar	85.55	86.13	76.83	93.75	**86.45**	58.50	79.24	99.76	85.55
Tic-tac-toe	**83.81**	73.29	73.69	97.91	82.78	22.22	75.89	98.91	73.07
Vehicle	**71.99**	82.15	63.36	96.44	71.52	45.56	68.09	99.10	70.10
Wisconsin	**96.28**	91.62	96.27	99.32	95.14	47.78	95.28	99.78	95.57
Zoo	95.58	86.56	**97.00**	86.24	94.75	55.63	87.97	95.76	92.81
Average	**81.66**	84.47	79.26	96.25	79.47	44.58	79.37	99.00	78.75

Table 3. Results of Wilcoxon Signed-Ranks Test

$\alpha = 0.1$	IS-CHC	FS-CHC	IFS-CHC	1-NN
CoCHC	+(.059)	+(.012)	+(.018)	+(.004)

– CoCHC outperforms all the remaining algorithms in 6 of 18 data sets.
– The loss in reduction rate achieved by CoCHC (compared with IS-CHC and IFS-CHC) is not too critical. It increases the average accuracy in 2% with respect to both them and keeps a good reduction rate.

In addition to Table 2, we have performed a two-tailed Wilcoxon Signed-Ranks Test [4], to statistically analyse the results obtained in the experiment. Table 3 shows the p-values obtained by Wilcoxon test.

The results offered by the test indicate us that the proposed model outperforms FS-CHC and IFS-CHC with a level of significance $\alpha = 0.05$, and it is better than IS-CHC considering a level of significance $\alpha = 0.1$.

5 Concluding Remarks

The purpose of this contribution is to present a cooperative coevolutionary model developed to tackle data reduction tasks to improve the classification based on the nearest neighbours technique. The proposal combines processes of evolutionary instance selection and feature selection techniques.

The results show that the use of cooperative coevolution in data reduction based on feature and instance selection can obtain promising results to optimize the performance of nearest neighbour classification.

Acknowledgement. This work was supported by TIN2008-06681-C06-01.

References

1. Baluja, S.: Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical report, Pittsburgh, PA, USA (1994)
2. Cano, J.R., Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study. IEEE Transactions on Evolutionary Computation 7, 561–575 (2003)
3. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13, 21–27 (1967)
4. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
5. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
6. Eshelman, L.J.: The CHC adaptative search algorithm: How to safe search when engaging in nontraditional genetic recombination. In: Foundations of Genetic Algorithms, pp. 265–283 (1990)
7. Fragoudis, D., Meretakis, D., Likothanassis, S.: Integrating feature and instance selection for text classification. In: 8th ACM SIGKDD international conference on KDD, pp. 501–506 (2002)
8. Freitas, A.A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer, New York (2002)
9. Ghosh, A., Jain, L.C.: Evolutionary Computation in Data Mining. Springer, Berlin (2005)
10. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)
11. Au, C., Leung, H.: Guided Mutations in Cooperative Coevolutionary Algorithms for Function Optimization. In: IEEE International Conference on Tools with Artificial Intelligence, pp. 407–414 (2007)
12. Liu, H., Motoda, H.: Instance Selection and Construction for Data Mining. The Springer International Series in Engineering and Computer Science (2001)
13. Liu, H., Motoda, H.: Computational Methods of Feature Selection. Chapman & Hall/Crc Data Mining and Knowledge Discovery Series (2007)
14. Oh, I., Lee, J., Moon, B.: Hybrid Genetic Algorithms for Feature Selection. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 1424–1437 (2004)
15. Panait, L., Wiegand, R.P., Luke, S.: Improving coevolutionary search for optimal multiagent behaviors. In: International Joint Conferences on Artificial Intelligence, pp. 653–658 (2003)
16. Panait, L., Luke, S., Harrison, J.F.: Archive-Based Cooperative Cooevolutionary Algorithms. In: Genetic and Evolutionary Computation Conference, pp. 345–352 (2006)
17. Potter, M.A., De Jong, K.A.: Cooperative coevolution: an architecture for evolving coadapted subcomponents. Evolutionary Computation 8, 1–29 (2000)
18. Pyle, D.: Data Preparation for Data Mining. The Morgan Kaufmann Series in DMS (1999)
19. Newman, D.J., Hettich, S., Merz, C.B.: UCI repository of ML databases (1998)
20. Whitley, D.: The GENITOR Algorithm and selective preasure: Why Rank Based Allocation of Reproductive Trials is Best. Genetic Algorithms, 116–121 (1989)
21. Jansen, T., Wiegand, R.P.: The Cooperative Coevolutionary (1+1) EA. Evolutionary Computation 12, 405–434 (2004)
22. Wolpert, D., Macready, W.: Coevolutionary Free Lunches. IEEE Transactions on Evolutionary Computation 9, 721–735 (2005)