IFS-CoCo in the Landscape Contest: Description and Results

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

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

Abstract. In this work, we describe the main features of IFS-CoCo, a coevolutionary method performing instance and feature selection for nearest neighbor classifiers. The coevolutionary model and several related background topics are revised, in order to present the method to the ICPR'10 contest "Classifier domains of competence: The Landscape contest". The results obtained show that our proposal is a very competitive approach in the domains considered, outperforming both the benchmark results of the contest and the nearest neighbor rule.

Keywords: Evolutionary Algorithms, Feature selection, Instance selection, Cooperative coevolution, Nearest neighbor.

1 Introduction

Data reduction [15] is one of the main process of data mining. 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).

The k-Nearest Neighbors classifier (NN) [3] is one of the most relevant algorithms in data mining [21]. It is a Lazy learning method [1], a classifier which does not build a model in its training phase. Instead of using a model, it is based on the instances contained in the training set. Thus, the effectiveness of the classification process relies on the quality of the training data. Also, it is important to note that its main drawback is its relative inefficiency as the size of the problem grows, regarding both the number of examples in the data set and the number of attributes which will be used in the computation of its similarity functions (distances) [2].

Instance Selection (IS) and Feature Selection (FS) are two of the most successful data reduction techniques in data mining. Both are very effective in reducing the size of the training set, filtrating and cleaning noisy data. In this way, they are able to enhance the effectiveness of classifiers (including NN), improving its accuracy and efficiency [11,12].

D. Ünay, Z. Çataltepe, and S. Aksoy (Eds.): ICPR 2010, LNCS 6388, pp. 56–65, 2010.

Evolutionary algorithms (EAs)[6] are general purpose search algorithms that use principles inspired by nature to evolve solutions to problems. In recent years, EAs have been successfully used in data mining problems[8,9], including IS and FS (defining them as combinatorial problems) [4,10].

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 over the last few years [20], being applied mainly in the optimization field [19].

In this work, we show the application of IFS-CoCo (Instance and Feature Selection based on Cooperative Coevolution, already published in [5]) over the benchmark domains defined for the contest: Classifier Domains of Competence: The Landscape Contest. The performance of our approach will be tested throughout the S1 benchmark set of the contest, and compared with the 1-NN classifier and other reference results.

This work is organized as follows: Section 2 gives an overview of the background topics related to our approach. Section 3 describes the main features of IFS-CoCo. Section 4 presents our participation in the Landscape contest and the results achieved. Section 5 concludes the work.

2 Background

In this section, two main topics related with our proposal will be reviewed: Evolutionary Instance and Feature Selection (Section 2.1), and Coevolutionary Algorithms (Section 2.2). Definitions and several cases of application will be shown in order to provide a solid background to present our approach.

2.1 Evolutionary Instance and Feature Selection

In recent years, EAs have arisen as useful mechanisms for data reduction in data mining. They have been widely employed 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 remove irrelevant and/or redundant features, with the aim of obtaining a simpler classification system, which also may improve the accuracy of the model in classification phase[12].

The IS problem can also be defined as a search process, where a reduced set S of instances is selected from the N examples of the training set, with $S <= N$. By choosing the most suitable points in the data set as instances for the reference data, the classification process can be greatly improved, concerning both efficiency and accuracy [11].

In [4], a complete study of the use of EAs in IS is done, highlighting four EAs to complete this task: Generational Genetic Algorithm (GGA), Steady-State Genetic Algorithm (SGA), CHC Adaptive Search Algorithm(CHC) [7] and Population-Based Incremental Learning (PBIL). They concluded that EAs outperform classical algorithms both in reduction rate and classification accuracy. They also concluded that CHC is the most appropriate EA to make this task, according to the algorithms they compared. Several researching efforts have been also applied to develop EA based FS methods. For example, [17] studies the capabilities of CHC applied to the FS problem.

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 (e.g. IS), store its results and to apply them to the second technique (e.g. FS), some authors have already tried to get some profit from the combined use of both approaches [10].

2.2 Coevolutionary Algorithms

A Coevolutionary Algorithm (CA) is an EA which is able to manage two or more populations simultaneously. Coevolution, the field in which CAs can be classified, can be defined as the co-existence of some interacting populations, evolving simultaneously. In this manner, evolutionary biologist Price [14] defined coevolution as *reciprocally induced evolutionary change between two or more species or populations.* A wider discussion about the meaning of Coevolution in the field of EC can be found in the dissertation thesis of Wiegand [18].

The most important characteristic of Coevolution is the possibility of splitting a given problem into different parts, employing a population to handle each one separately. This allows the algorithm to employ a *divide-and-conquer* strategy, where each population can focus its efforts on solving a part of the problem. If the solutions obtained by each population are joined correctly, and the interaction between individuals is managed in a suitable way, the Coevolution model can show interesting benefits in its application.

Therefore, the interaction between individuals of different populations is key to the success of Coevolution techniques. In the literature, Coevolution is often divided into two classes, regarding the type of interaction employed:

Cooperative Coevolution: In this trend, each population evolves individuals representing a component of the final solution. Thus, a full candidate solution is obtained by joining an individual chosen from each population. In this way, increases in a collaborative fitness value are shared among individuals of all the populations of the algorithm [13].

Competitive Coevolution: In this trend, the individuals of each population compete with each other. This competition is usually represented by a decrease in the fitness value of an individual when the fitness value of its antagonist increases [16].

In this work, we will focus our interest on Cooperative Coevolution, since its scheme of collaboration offers several advantages for the development of approaches which integrate several techniques related, e.g. data reduction techniques.

3 IFS-CoCo: Instance and Feature Selection Based on Cooperative Coevolution

In this section we present IFS-CoCo, providing a description of its most important charasteristics. A full study concerning several advanced topics about its behavior (including optimization of its parameter, capabilities when applied to medium sized data sets, and more) can be found in [5].

Our approach performs several data reduction process (instance selection, feature selection, and both) in order to build a multiclassifier based on the well-known nearest neighbor rule (three 1-NN classifiers whose output is agregated by a majority rule). It defines a cooperative coevolutionary model composed of three populations, which evolve simultaneously:

- An Instance Selection population (IS).
- A Feature Selection population (FS).
- A dual population, performing both Instance and Feature Selection (IFS).

Figure 1 depicts its organization (N denotes the number of instances in the training set, whereas M denotes the number of features). In isolation, the three populations can be seen as genetic-based search methods, where their respective chromosomes encode the features/instances currently selected. However, in contrast to existing evolutionary approaches for Instance Selection [4] or to wrapper approaches for Feature Selection [12], the evaluation of the quality of the chromosomes (i.e. the fitness function) is not performed in isolation.

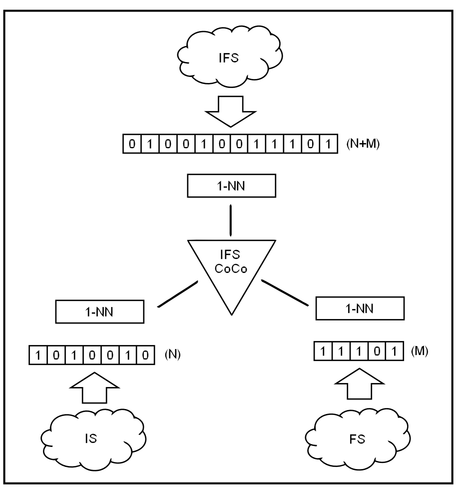

Fig. 1. Population scheme of IFS-CoCo: The three populations (IS, FS and IFS) define three 1-NN classifiers whose output are merged by a majority vote

Performing an evaluation of the fitness function of IFS-CoCo requires three chromosomes (one for each population). Once they have been gathered, the fitness value of a chromosome is computed as follows:

$$\begin{aligned} Fitness(J) = {} & \alpha \cdot \beta \cdot clasRate(J) \\ & + (1-\alpha) \cdot ReductionIS(J) \\ & + (1-\beta) \cdot ReductionFS(J) \end{aligned} \tag{1}$$

- **clasRate(J):** Classification accuracy over the training set. In order to compute this accuracy, a multiclassifier is built based on three 1-NN classifiers. Each of them will employ as reference set only the subset defined by each chromosome selected.
  Thus, each chromosome defines a reduced version of the original training set, which may give a different output than the rest when classifying a training instance. In order to join these outputs, a majority voting process is performed by the 1-NN classifiers. Its result is taken as the final output of the multiclassifier.
  Finally, the *clasRate* is computed as the classification accuracy over the training set by the multiclassifier. This value is assigned to the three chromosomes employed to compute the fitness function.
- **ReductionIS(J):** Ratio of instances discarded from the original training set.
- **ReductionFS(J):** Ratio of features discarded from the original training set.

The search process of the three populations is conducted by the CHC algorithm [7]. The populations evolve sequentially, performing a generation in each step, before starting to evolve the next population. This process is carried out until the specified number of evaluations runs out. Then, the best chromosome of each population is gathered, in order to build a final multiclassifier, ready to classify test instances. This multiclassifier will work in the same manner as all the multiclassifiers employed in the coevolutionary process.

4 The Landscape Contest

In this section, we describe the experimental study performed. Since it is a part of the Landscape contest, data sets (Section 4.1) are fixed by the organizers. Comparison algorithms and configuration parameters employed are also considered (Section 4.2). Results obtained in the study are shown (Section 4.3) and analyzed (Section 4.4), discussing the strengths and limitations of our approach.

4.1 Problems

The data sets considered belong to the S1 benchmark provided by the organization[1]. It consists of 300 real-valued small data sets (with less than 1000 instances)

[1] http://www.salle.url.edu/ICPR10Contest/?page_id=21

and a large data set with roughly 10000 instances, being the majority of them two-class problems.

These data sets have been partitioned by using the ten fold cross-validation (10-fcv) procedure, and their values have been normalized in the interval $[0, 1]$ to equalize the influence of attributes with different ranges (a desirable charasteristic for NN-based classifiers, e.g. IFS-CoCo).

4.2 Comparison Algorithms and Configuration Parameters

In order to test the performance of IFS-CoCo, we have selected two additional methods as reference: 1-NN rule (the baseline classifier whose performance is enhanced by the preprocessing techniques of IFS-CoCo) and the benchmark results offered at the contest web page[2]. We will denote these results as *Benchmark* method throughout the study.

The configuration parameters of IFS-CoCo are the same that were used in its original presentation [5]:

- Number of evaluations: 10000
- Population size: 50 (for each population)
- α weighting factor: 0.6
- β weighting factor: 0.99

Given the scale of the experiment, we have not performed a fine-tuning process of the parameters for each data set. Instead, we have employed the same configuration in all runs, expecting a suitable behavior in all the cases (for a wider discussion about the tuning of α and β parameters, see Section 5.3 of [5]).

4.3 Results Achieved on Benchmark S1

Table 1 shows a summary of the results achieved in the S1 benchmark. Accuracy denotes the average accuracy obtained through a 10-folds cross validation procedure, whereas Wins denotes the number of data sets in which each method achieves the best result of the experiment.

Table 1. Summary of results

Method	Accuracy	Wins
IFS-CoCo	87.73	173
1-NN	74.17	10
Benchmark	81.84	117

Moreover, Figures 2 and 3 depict a comparison between IFS-CoCo and *Benchmark* or 1-NN, respectively. The dots symbolize the accuracy achieved in test

[2] `http://www.salle.url.edu/ICPR10Contest/DataSets/TrainingAccuracy.txt`

phase by the two classifiers in a concrete dataset (thus, 301 points are represented). A straight line splits the graphic, exactly at the points where the accuracy measure of both classifiers is equal. Therefore, those points below (right) of the line represent data sets where IFS-CoCo behaves better than the comparison algorithm, whereas those points above (left) of the line represent the opposite.

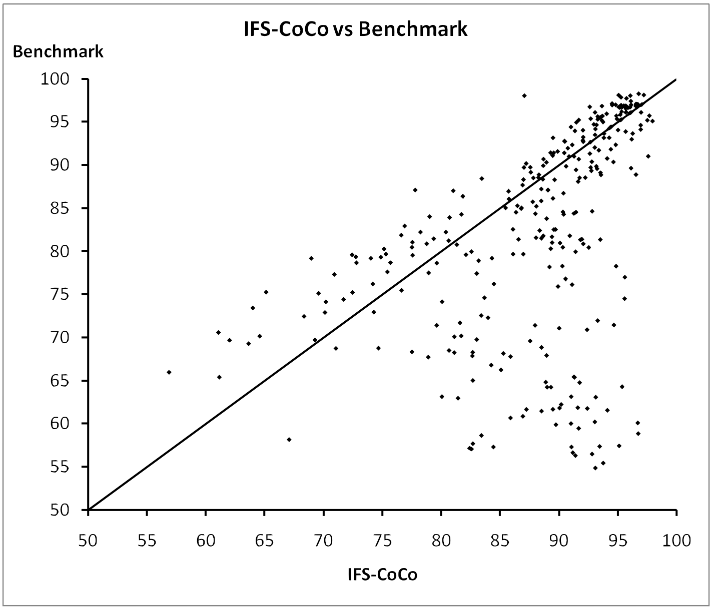

Fig. 2. Graphical comparison of IFS-CoCo vs Benchmark

These graphics emphasize the superiority of IFS-CoCo over the comparison algorithms. In comparison with *Benchmark*, our approach greatly improves its results in a large number of problems. The majority of the problems in which *Benchmark* improves IFS-CoCo are *easy* problems (those in which both classifiers achieved more than a 90% of accuracy), where there are no great differences. On the contrary, in harder problems, differences are much greater.

Furthermore, the differences between IFS-CoCo and 1-NN are even greater: There are only a few points in which 1-NN outperforms our approach, most of them depicting *easy* problems.

4.4 Strength and Limitations of Our Approach

As we have shown in the former subsection, our approach is able to improve the performance of the comparison methods. The simultaneous search for the best instances and features allows IFS-CoCo to dynamically adapt its behavior to different kinds of problems (i.e. giving more importance to features in some

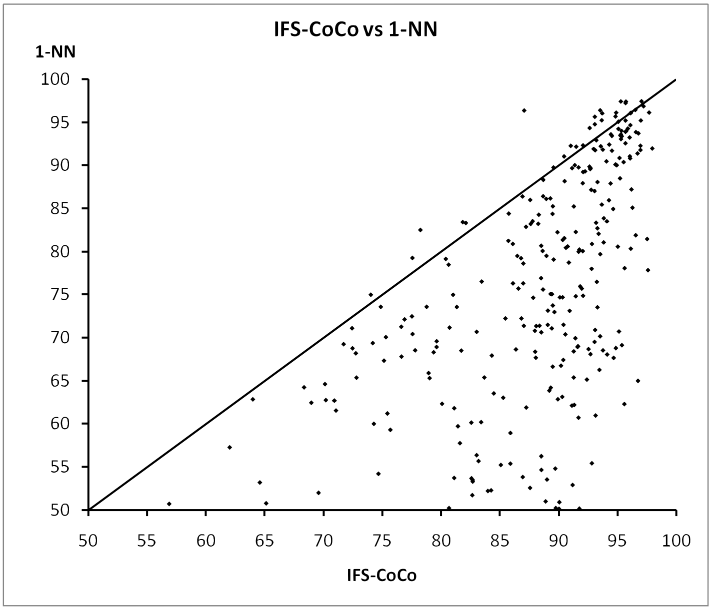

Fig. 3. Graphical comparison of IFS-CoCo vs 1-NN

problems and instances in the rest), showing a robust behavior in most of the domains considered (e.g. the greatest improvement of *Benchmark* over IFS-CoCo is in data set d60, 10.98%, whereas the greatest improvement of IFS-CoCo over benchmark is almost four times higher, 38.32%, in d17).

Moreover, the much reduced size of the subsets selected by IFS-CoCo to build the final classifier allows to classify quickly the test sets, being 1-NN often slower than the multiclassifier in test phase.

On the other hand, the main limitation of IFS-CoCo is the computation time in training phase. The cost of computing 10000 times the fitness function by means of three 1-NN classifiers is high, thus an important amount of time is required in order to let IFS-CoCo select the best possible subsets from the training data (in S1 phase, IFS-CoCo spent almost 4 days to finish the 10-folds cross validation procedure over the 301 data sets).

5 Conclusions

In this work, we have shown the preliminary results of IFS-CoCo in The Landscape contest. These results highlight the good performance of our approach in general classification domains, outperforming those achieved by the 1-NN rule. Moreover, it also outperforms the benchmark results offered by the organizers of the contest.

The main capability of our approach (the ability of working simultaneously both in the instances' and features' space) has been the key to the robust behavior shown in these problems, performing well in most of the domains considered. However, a future analysis of the characteristics of the data sets employed could give a new insight about the strength and limitations of our approach.

Acknowledgments. This work was supported by Project TIN2008-06681-C06-01. J. Derrac holds a research scholarship from the University of Granada.

References

1. Aha, D.W. (ed.): Lazy Learning. Springer, Heidelberg (1997)
2. Chen, Y., Garcia, E.K., Gupta, M.R., Rahimi, A., Cazzanti, L.: Similarity-based classification: Concepts and algorithms. Journal of Machine Learning Research 10, 747–776 (2009)
3. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13, 21–27 (1967)
4. 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(6), 561–575 (2003)
5. Derrac, J., García, S., Herrera, F.: IFS-CoCo: Instance and feature selection based on cooperative coevolution with nearest neighbor rule. Pattern Recognition 43(6), 2082–2105 (2010)
6. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Natural Computing. Springer, Heidelberg (2003)
7. Eshelman, L.J.: The CHC adaptative search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In: Rawlins, G.J.E. (ed.) Foundations of Genetic Algorithms, pp. 265–283. Morgan Kaufmann, San Francisco (1991)
8. Freitas, A.A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer, Heidelberg (2002)
9. Ghosh, A., Jain, L.C. (eds.): Evolutionary Computation in Data Mining. Springer, Heidelberg (2005)
10. Ho, S.Y., Liu, C.C., Liu, S.: Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm. Pattern Recognition Letters 23(13), 1495–1503 (2002)
11. Liu, H., Motoda, H. (eds.): Instance Selection and Construction for Data Mining. The International Series in Engineering and Computer Science. Springer, Heidelberg (2001)
12. Liu, H., Motoda, H. (eds.): Computational Methods of Feature Selection. Chapman & Hall/Crc Data Mining and Knowledge Discovery Series. Chapman & Hall/CRC, Boca Raton (2007)
13. Potter, M.A., De Jong, K.A.: Cooperative coevolution: an architecture for evolving coadapted subcomponents. Evolutionary Computation 8, 1–29 (2000)
14. Price, P.W.: Biological Evolution. Saunders College Publishing, Philadelphia (1998)
15. Pyle, D.: Data Preparation for Data Mining. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, San Francisco (1999)

16. Rosin, C.D., Belew, R.K.: New Methods for competitive coevolution. Evolutionary Computation 15, 1–29 (1997)
17. Whitley, D., Guerra-Salcedo, C.: Genetic search for feature subset selection: A comparison between CHC and GENESIS. In: Proceedings of the Third Annual Conference on Genetic Programming, Wisconsin, pp. 504–509 (1998)
18. Wiegand, R.P.: An Analysis of Cooperative Coevolutionary Algorithms. Ph. D. thesis, George Mason University, Fairfax, Virginia (2003)
19. Jansen, T., Wiegand, R.P.: The Cooperative Coevolutionary (1+1) EA. Evolutionary Computation 12, 405–434 (2004)
20. Wolpert, D., Macready, W.: Coevolutionary Free Lunches. IEEE Transactions on Evolutionary Computation 9, 721–735 (2005)
21. Wu, X., Kumar, V. (eds.): The Top Ten Algorithms in Data Mining. Data Mining and Knowledge Discovery. Chapman & Hall/CRC, Boca Raton (2009)