A Co-evolutionary Framework for Nearest Neighbor Enhancement: Combining Instance and Feature Weighting with Instance Selection

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

[1] Dept. of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology), University of Granada, 18071 Granada, Spain
{jderrac,triguero,herrera}@decsai.ugr.es
[2] Dept. of Computer Science, University of Jaén, 23071 Jaén, Spain
sglopez@ujaen.es

Abstract. The nearest neighbor rule is one of the most representative methods in data mining. In recent years, a great amount of proposals have arisen for improving its performance. Among them, instance selection is highlighted due to its capabilities for improving the accuracy of the classifier and its efficiency simultaneously, by editing noise and reducing considerably the size of the training set. It is also possible to remark the role of feature and instance weighting as outstanding methodologies for improving further the performance of the nearest neighbor rule.

In this work we present a new co-evolutionary algorithm for combining the former techniques. Its performance is compared with evolutionary approaches performing instance selection, instance weighting and feature weighting in isolation, as well as with the nearest neighbor classifier. The results obtained, contrasted through nonparametric statistical tests, supports the capabilities of co-evolution as a outstanding strategy for joining several proposals for enhancing the nearest neighbor rule.

Keywords: Co-evolution, Instance Selection, Instance Weighting, Feature Weighting, Evolutionary Algorithms, Nearest Neighbor Classifier.

1 Introduction

The k-nearest neighbor classifier (k-NN) is one of the best known techniques in data mining. It is one of the most used algorithms in supervised classification. Due to its simplicity, effectiveness and precision, it has attracted a great interest by the research community [16].

Instance selection is a well-known proposal for improving the performance of the k-NN classifier [10,8]. Its application allows us to reduce the spatial complexity of the classifier and to improve its efficiency, by the deletion of irrelevant instances in the training set, and its precision, by removing noisy instances.

Another interesting proposal is the use of weighting schemes for adjusting the distance function of the k-NN. These schemes can be applied both to the

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 176–187, 2012.

instances [3] and the features [14] of the training set. A proper set of weights for adjusting the distance function can help to train the classifier to the specific domain of the problem considered, enhancing its generalization capabilities.

A great number of the approaches proposed in recent years for improving data mining processes are related to evolutionary computation [9]. Given that the processes of performing instance selection and obtaining proper weights can be defined as search problems, evolutionary algorithms can be applied to tackle them, with promising results [4,13].

Recently, the joint application of several preprocessing techniques over a single classifier has been considered through the use of co-evolutionary algorithms [5]. The field of cooperative co-evolution [11] offers a useful framework in which several optimization techniques can be applied simultaneously, obtaining better results than those expected by using the same techniques in isolation.

In this work we present a co-evolutionary model for instance selection and instance and feature weighting, applied to the k-NN classifier (CIW-NN). This model is composed by 3 populations, where each one is focused on a specific task for improving a 1-NN classifier (instance selection, feature weighting and instance weighting). After its description, we present a full experimental study where the improvements of the model over the preprocessing techniques applied in isolation is shown. These improvements are contrasted by using nonparametric statistical tests [7], which are highly recommended for analyzing the results obtained in data mining experiments such as this one.

The rest of the work is organized as follows: Section 2 presents some preliminary concepts about the techniques used in this work. Section 3 describes the proposed model. Section 4 presents the experimental study performed for testing the behavior of CIW-NN when compared with several non-co-evolutionary techniques. Finally, Section 5 shows the conclusions arrived at.

2 Background

This section surveys some necessary preliminary concepts for describing CIW-NN. Section 2.1 presents co-evolution and some of its most interesting characteristics. Section 2.2 describes the use of instance selection in classification. Finally, Section 2.3 shows how the weighting schemes can be used for improving the precision of the classifiers.

2.1 Co-evolution

Co-evolution is the area of evolutionary computation related to techniques able to manage several different populations simultaneously. Its application consists of splitting the domain of the problem using a *divide and conquer* strategy where each population is focused on tackling a single part of the problem.

Within this field, cooperative co-evolution [11] defines how the different population can cooperate. In general, this is met by using global fitness functions which require an individual of each population for being evaluated. This allows to benefit those individuals who behave well in cooperation with the rest of populations, in contrast with the classical fitness functions which only considers the quality of individuals in isolation.

Thus, the main motivation for using cooperative co-evolution lies in its decomposition capabilities, which can be used under several assumptions to break the *No Free Lunch* barrier present in most optimization problems [15].

2.2 Instance Selection

The main goal of instance selection [10,8] is to isolate the smallest set of instances which enable a data mining algorithm to predict the class of a query instance with the same quality as the initial data set. By minimizing the data set size, space complexity and computational cost of the subsequent data mining algorithms are reduced, improving their generalization capabilities.

It can be defined as follows: Let X be an instance where $X = (x_1, x_2, \cdots, x_M, x_c)$, with X belonging to a class c, given by X_c, and an M-dimensional space in which x_i is the value of the i-th feature of the sample X. Then, let us assume that there is a training set TR composed by N instances, and a test set TS composed by T instances. Let $RS \subseteq TR$ be the subset of selected samples that result from the execution of an instance selection algorithm. Then, each new instance T from TS can be classified by from a data mining algorithm acting over the instances of RS.

2.3 Weighting Schemes

The use of weighting schemes is another interesting enhancement for the classifiers' behavior. Although there are many different approaches for this, in this work we will focus our interest in using the weights for modifying the distance function used by the classifier.

Therefore, it is possible to define weights associated both to the features (that is, real values to weight the importance of each feature in the computation of the similarity between two instances) and to the instances (that is, real values to modify the effective distance between two instances with respect to some related properties, such as, for example, its class attribute). Both schemes have been widely studied in the past [14,3].

The final goal of the inclusion of these schemes is to improve as further as possible the precision of the classifier. Hence, most of these methods are applied through an optimization process using the original training set as reference.

3 Proposed Model

In this section we present the CIW-NN co-evolutionary model. Section 3.1 describes the different subcomponents of the model. Section 3.2 shows the fitness function designed. Finally, Section 3.3 describes the general co-evolutionary model.

3.1 CIW-NN Subcomponents

CIW-NN is based on the simultaneous search of the best possible subset of training instances, and the best possible weighting schemes for instances and features. To do so, three populations are defined and focused on three specific goals:

- Instance selection (IS): Search the best subset of training instances.
- Instance weighting (IW): Search the best weighting scheme for instances.
- Feature weighting (FW): Search the best weighting scheme for features.

Although the three populations perform a search task, they can be discriminated by several characteristics. Table 1 summarizes them:

Table 1. CIW-NN population's characteristics

Topic	IS population	IW population	FW population
Scope	Instances	Instances	Features
Codification	Binary	Real	Real
Granularity	Individual	Class	Individual
Epoch length	Simple	Multiple	Multiple
Objective	Acc./Red.	Accuracy	Accuracy

- **Scope:** Each population is focused on optimizing either instances or features.
- **Codification:** Depending on the concrete enhancement task performed, the individuals of each population will employ binary $(0, 1)$ or real $([0, 1])$ codification. This feature will define the kind of basic search method which the population will carry out, and also has a strong effect on the difficulty of the search task itself, due to real coded search spaces usually being wider and harder to explore.
- **Granularity:** CIW-NN uses two schemes of assignation of weights. Individual weights (one for each instance/feature) are assigned to IS and FW chromosomes, whereas Class weights, shared by instances of the same class, are assigned to IW chromosomes.
- **Epoch length:** CIW-NN defines how the evolution process of its populations will be scheduled, by assigning epochs of different length: Simple, that is, one generation per cycle of the global model, or Multiple, considering more than one generation. This way, CIW-NN equalizes the number of evaluations spent by each population.
- **Objective:** It refers to the objective which each population pursues. A population can cope with maximizing the accuracy obtained by the classifier, or to maximize simultaneously this accuracy and the reduction rate, that is, the ratio between the number of instances discarded and the ones that composed the original training set.

The IS population performs the search using the CHC algorithm [6], considering the configuration shown in [4], where it is highlighted as a proficient method for this task. For improving its reduction capabilities their binary chromosomes are initialized with a certain bias, where only $prob1$ instances are selected (initialized to 1). Moreover, we have modified the original HUX crossover operator, so it only maintains $prob0to1$ instances selected after its application.

In the IW and FW populations the search is guided by a real coded steady-state genetic algorithm. A crossover operator with multiple descendants has been selected due to its good convergence capabilities [12]. Among all the models suggested in that study, the best results have been obtained with the operator 2BLX0.3-4BLX0.5-2BLX0.7, based on the operator BLX-α. Figure 1 depicts its application, performing 4 crossing operations with different values of the α parameter, and selecting the best offspring found. The mutation operator selected is the non-uniform one, following the recommendations of [12].

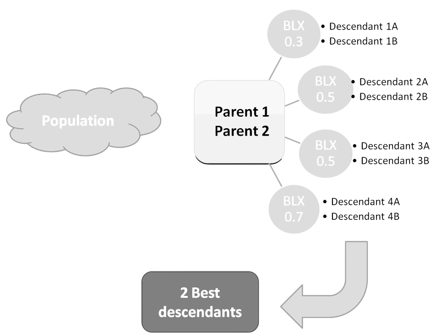

Fig. 1. Crossover operator with multiple descendants

3.2 Fitness Function

The CIW-NN fitness function is composed by two different components:

– **Accuracy:** Precision of the baseline classifier (1-NN) over the training set (using leave-one-out with the configuration of instances and weights which is evaluated).
– **Reduction:** Reduction rate of the subset of instances evaluated, over the full training set.

When performing an evaluation of the fitness function, it is required to use a chromosome from each population. If we define H as a IS population chromosome, I as a IW population chromosome, and J as a FW population chromosome, the fitness value assigned to each one is the following

$$\begin{aligned} Fitness(H) &= \alpha \cdot Ac(H,I,J) + (1-\alpha) \cdot Red(H) \\ Fitness(I) &= Ac(H,I,J) \\ Fitness(J) &= Ac(H,I,J) \end{aligned} \tag{1}$$

where $Ac(H, I, J)$ is the accuracy estimated by the classifier, $Red(H)$ is the reduction rate obtained and α is a real value $[0, 1]$ used for weighting both objectives (we have set this value to $\alpha = 0.5$, following the recommendations given in [4]).

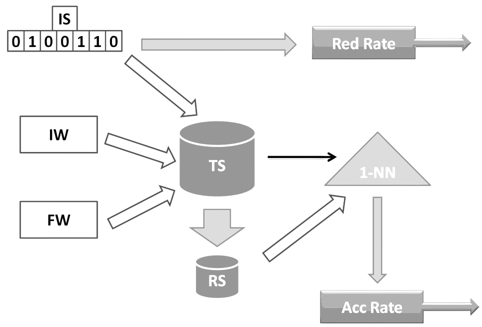

Fig. 2. Fitness function evaluation

Figure 2 shows an scheme of a fitness function evaluation. Accuracy is estimated by preprocessing the training set selecting the instances indicated by the chromosome of the IS population H, and weights defined by the IW and FW chromosomes I and J are attached to obtain the resulting set RS. This set is used as reference for the 1-NN classifier, whose accuracy $Ac(H, I, J)$ is estimated by classifying the original training set.

The similarity function used by the 1-NN classifier used to estimate the accuracy is the euclidean one. CIW-NN defines a modified version of it

$$D(x, y) = IW_{c(y)} * \sum_{i=0}^{M} FW_i \cdot \sqrt{(x_i - y_i)^2} \tag{2}$$

where x is the instance to classify, y a instance from the resulting set TS, $IW_{c(y)}$ denotes the weight assigned to the class attribute of the instance y, and FW_i denotes the weight assigned to the feature i.

3.3 Co-evolutionary Model

CIW-NN merges all the components described in the former sections in a single framework. The three populations evolves in a cycle, consuming an epoch (a fixed number of generations/evaluations) each one before passing the turn to the next population.

Figure 3 depicts the co-evolutionary scheme: The cycle is started by the IS population, performing a single generation (simple epoch). Then, the IW population performs a fixed number of generations (multiple epoch). Afterwards, the FW population performs another multiple epoch, finishing the cycle.

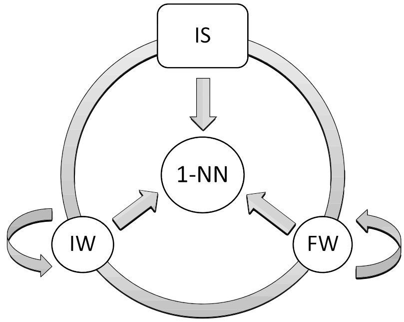

Fig. 3. CIW-NN populations scheme

```
Initialize IS, IW and FW populations;
Select the best individuals of each population;
While the limit of evaluations is not met:
     Perform an IS epoch;
     Perform an IW epoch;
     Perform an FW epoch;
     Update the best individuals found;
End
Return the best individuals found;
```

Fig. 4. CIW-NN general co-evolutionary scheme

Figure 4 summarizes the general scheme of the model. At the end of each cycle, the best individuals of each population are selected (the very first individuals - line 2 of Figure 4 - are selected according to their individual fitness). Their task will be to complement the evaluations of the new individuals generated by the search process. In this way, when a new individual must be evaluated, the best individuals selected at the two other populations are gathered, obtaining then the 3 chromosomes required by the fitness function.

This is an optimal configuration for modeling the cooperation between populations. The joint evaluation of each individual with the best individuals of the other populations allows to guide the search to more promising areas of the search space, which represent the most desirable properties of each enhancement technique. The use of the epoch model and the common fitness function allows to control how the search progresses in each component, preventing premature convergence and/or a faster convergence process of a given population to the detriment of the rest (which may lead to optimal solutions from the single point of view, but suboptimal in the cooperative sense).

Table 2. Data sets considered in the experimental study

Data set	#In.	#Ft.	#Cl.	Data set	#In.	#Ft.	#Cl.
Australian	690	14	2	Monk-2	432	6	2
Balance	625	4	3	Movement	360	90	15
Bands	539	19	2	New Thyroid	215	5	3
Breast	286	9	2	Pima	768	8	2
Bupa	345	6	2	Saheart	462	9	2
Car	1728	6	4	Sonar	208	60	2
Cleveland	303	13	5	Spectfheart	267	44	2
Contraceptive	1473	9	3	Tae	151	5	3
Dermatology	366	34	6	Tic-tac-toe	958	9	2
German	1000	20	2	Vehicle	846	18	4
Glass	214	9	7	Vowel	990	13	11
Hayes-roth	160	4	3	Wine	178	13	3
Housevotes	435	16	2	Wisconsin	699	9	2
Iris	150	4	3	Yeast	1484	8	10
Lymphography	148	18	4	Zoo	101	16	7

4 Experimental Study

In this section, we describe the experimental study performed to characterize the behavior of CIW-NN in supervised classification problems. Section 4.1 describes the data sets used. Section 4.2 enumerates the algorithms selected for the comparison and describes their parameters. Section 4.3 presents and analyze the results obtained. Finally, Section 4.4 shows the statistical study performed for contrasting the results of the experiment.

4.1 Data Sets

We have selected 30 supervised classification data sets for this study. They have been taken from the *UCI Machine Learning Repository*[1] and *KEEL-dataset Repository*[2]. Table 2 shows their main characteristics: Name, number of instances **#In.** (examples) , number of features **#Ft.** and number of classes **#Cl.**.

Every data set has been partitioned using a 10-folds cross validation procedure. Moreover, the attribute values have been normalized into the interval $[0, 1]$. This will help in equalizing the influence of every attribute with respect to the distance measure selected for the classifiers.

4.2 Algorithms and Parameters

In addition to CIW-NN, in this study we have used as comparison algorithms the three baseline methods of the populations of the co-evolutionary model:

[1] `http://www.ics.uci.edu/~mlearn/MLRepository.html`
[2] `http://www.keel.es/datasets.php`

Table 3. Parameters of the methods

Method	Parameters
CIW-NN	α: 0.5, *prob*0*to*1: 0.25, *prob*1: 0.25, Epoch length: 40 evaluations
	Mutation probability: 0.05 per chromosome
CHC-IS	α: 0.5, *prob*0*to*1: 0.25, *prob*1: 0.25
SSGA-FW	Mutation probability: 0.05 per chromosome
SSGA-IW	Mutation probability: 0.05 per chromosome
Common parameters	Crossover operator (real): 2BLX0.3-4BLX0.5-2BLX0.7,
	Crossover operator (binary): Modified HUX
	Evaluations: 10000, Population size: 50, Base classifier: 1-NN

The CHC algorithm for IS (CHC-IS), a Steady-State Genetic Algorithm with multiple descendants for feature weighting (SSGA-FW) and a Steady-State Genetic Algorithm with multiple descendants for instance weighting (SSGA-IW). Moreover, we have included the 1-NN rule as a basic classifier for reference.

All these methods have been coded in Java, using the KEEL Software [1,2][3]. In the experimental study, we have applied a 5x10-folds cross validation procedure for evaluating their behavior. Table 3 shows the parameters considered.

4.3 Results Obtained

In the experimental study we have considered as performance measures the accuracy in test phase (accuracy when classifying new examples unseen by the classifier at the training phase) and the reduction rate obtained over the instances of TR, for those methods which are able to perform it (CIW-NN and CHC-IS).

Table 4 shows the results obtained. For each data set, the table shows the average value obtained in each performance measure. Moreover, the best result obtained in each data set is highlighted in **bold**.

Using the results of the table, we can get the following conclusions:

- The proposed approach, CIW-NN, obtains the best average accuracy. Furthermore, it outperforms all the comparison methods in 18 out of 30 problems considered.
- All the methods selected in the study greatly improves the accuracy of the 1-NN classifier.
- Both CIW-NN and CHC-IS are able to reduce the size of the training sets to less of the 10% of its original size, without harming the accuracy of the classifier.

These results supports the capabilities of instance selection and the weighting techniques for improving the performance of the 1-NN classifier. In the case

[3] `http://www.keel.es`

Table 4. Results obtained

Performance	Accuracy (%)					Reduction (%)	
Data set	CIW-NN	CHC-IS	SSGA-FW	SSGA-IW	1-NN	CIW-NN	CHC-IS
Australian	**81.74**	81.45	81.01	80.87	81.45	93.66	97.67
Balance	**85.75**	79.04	73.76	80.33	79.04	94.24	96.62
Bands	**75.52**	74.04	72.75	72.92	74.04	95.49	97.28
Breast	**70.62**	66.04	63.06	69.98	65.35	97.86	97.71
Bupa	60.95	62.51	**62.91**	62.29	61.08	95.36	96.55
Car	**95.89**	85.65	94.91	86.34	85.65	83.78	95.87
Cleveland	56.43	53.14	52.48	**56.45**	53.14	97.14	98.13
Contraceptive	**45.22**	42.63	44.06	44.61	42.77	84.36	97.04
Dermatology	**96.72**	95.35	96.45	94.26	95.35	96.02	96.45
German	**72.10**	70.50	69.50	71.90	70.50	89.13	97.99
Glass	**75.72**	74.50	72.36	69.35	73.61	93.25	93.51
Hayes-roth	72.15	71.01	69.96	**73.03**	35.70	91.92	92.34
Housevotes	**94.93**	91.24	93.78	91.23	91.24	97.80	98.24
Iris	93.33	93.33	**94.00**	**94.00**	93.33	96.37	95.93
Lymphography	**79.30**	73.87	76.54	77.34	73.87	94.23	94.67
Monk-2	**100.00**	95.32	**100.00**	75.09	77.91	93.29	95.40
Movement	83.06	86.39	86.67	**88.06**	81.94	74.69	88.09
New Thyroid	95.82	**97.23**	96.28	95.84	**97.23**	96.95	97.62
Pima	**71.24**	70.33	70.71	70.59	70.33	92.09	97.09
Saheart	**65.37**	64.49	64.06	64.28	64.49	96.34	97.88
Sonar	**87.00**	85.55	85.07	86.02	85.55	91.67	93.11
Spectfheart	77.92	69.70	74.63	**78.68**	69.70	98.17	97.96
Tae	65.71	65.04	**68.38**	63.04	40.50	93.82	94.41
Tic-tac-toe	87.37	82.07	**91.33**	73.07	73.07	88.67	95.62
Vehicle	**71.28**	70.10	71.16	66.55	70.10	90.28	94.48
Vowel	98.28	**99.39**	99.29	98.38	**99.39**	74.97	84.01
Wine	97.16	95.52	96.63	**97.75**	95.52	96.88	96.69
Wisconsin	96.00	95.57	95.57	**96.42**	95.57	94.74	99.21
Yeast	**52.76**	52.23	50.81	52.63	50.47	83.49	97.19
Zoo	**97.50**	96.83	96.83	95.58	92.81	89.99	89.34
Average	**80.09**	78.00	78.83	77.56	74.69	91.89	95.47

of the co-evolutionary model, this improvement is considerable, since it offers simultaneously the best results on accuracy and very high reduction rates over the training sets (which means a great improvement in the efficiency of the classifier, in terms of both storage requirements and running time).

4.4 Statistical Study

Nonparametric statistical tests for multiple comparisons may be used to contrast the experimental results achieved. Their use in data mining is specially

Table 5. Ranks of the Friedman test and p-values of the post-hoc methods

Control method: CIW-NN (Rank: 1.850)				
Method	Rank	Holm	Hochberg	Finner
CHC-IS	3.267	0.00156	0.00156	0.00104
SSGA-FW	3.117	0.00384	0.00384	0.00256
SSGA-IW	2.967	0.00623	0.00623	0.00623
1-NN	3.800	0.00001	0.00001	0.00001

recommended in those cases in which it is necessary to contrast the results of a new proposal with several comparison methods [7].

In this study, we will use the Friedman test for detecting significant differences between accuracy results. Holm, Hochberg and Finner procedures will be used as *post-hoc* tests for characterizing the differences found.[4]

After the application of the Friedman test, significant differences among the algorithms ($p = 0.00006$) are found. Hence, CIW-NN is selected as the control method (the one with the lowest rank) for the post-hoc procedures.

Table 5 summarizes the results obtained. CIW-NN improves statistically the results of the comparison methods at a $\alpha = 0.01$ level of significance (the three *post-hoc* methods obtain p-values lower than 0.01 in every case). Hence, the study contrast that the improvement of CIW-NN over CHC-IS, SSGA-FW, SSGA-IW and 1-NN is significant.

5 Conclusions and Future Work

In this work it is proposed a new approach hybridizing several data preprocessing and adjusting methods for the k-NN classifier, within a co-evolutionary framework. The experimental study performed supports the use of co-evolution as a practical tool for improving the results of the selected techniques.

Several ideas arise as future work, including the comparison of the model with a set of representative data reduction and weighting approaches of the state of the art, and the evaluation of the performance of CIW-NN in large classification domains. Moreover, the efficacy of the method could be further improved if more accurate fitness functions are developed.

Acknowledgements. This work was supported by Projects TIN2011-28488 and TIC-2010-6858. J. Derrac holds a FPU scholarship from Spanish Ministry of Education. I. Triguero holds a scholarship from the University of Granada.

[4] More information can be found at the SCI2S thematic website on *Statistical Inference in Computational Intelligence and Data Mining* `http://sci2s.ugr.es/sicidm/`

References

1. Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M., Ventura, S., Garrell, J., Otero, J., Romero, C., Bacardit, J., Rivas, V., Fernández, J., Herrera, F.: KEEL: A software tool to assess evolutionary algorithms to data mining problems. Soft Computing 13, 307–318 (2009)
2. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic and Soft Computing 17(2-3), 255–287 (2011)
3. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally weighted learning. Artificial Intelligence Review 11, 11–73 (1997)
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. 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 Mateo (1991)
7. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180, 2044–2064 (2010)
8. García, S., Derrac, J., Cano, J.R., Herrera, F.: Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(3), 417–435 (2012)
9. Ghosh, A., Jain, L.C. (eds.): Evolutionary Computation in Data Mining. Springer, Heidelberg (2005)
10. Liu, H., Motoda, H. (eds.): Instance Selection and Construction for Data Mining, ser. The Springer International Series in Engineering and Computer Science. Springer, Heidelberg (2001)
11. Potter, M.A., Jong, K.A.D.: Cooperative coevolution: An architecture for evolving coadapted subcomponents. Evolutionary Computation 8(1), 1–29 (2000)
12. Sánchez, A.M., Lozano, M., Villar, P., Herrera, F.: Hybrid crossover operators with multiple descendents for real-coded genetic algorithms: Combining neighborhood-based crossover operators. International Journal on Intelligent Systems 24(5), 540–567 (2009)
13. Triguero, I., García, S., Herrera, F.: IPADE: Iterative Prototype Adjustment for Nearest Neighbor Classification. IEEE Transactions on Neural Networks 21(12), 1984–1990 (2010)
14. Wettschereck, D., Aha, D.W., Mohri, T.: A review and empirical evaluation of feature weigthing methods for a class of lazy learning algorithms. Artificial Intelligence Review 11, 273–314 (1997)
15. Wolpert, D.H., Macready, W.G.: Coevolutionary free lunches. IEEE Transactions on Evolutionary Computation 9(6), 721–735 (2005)
16. Wu, X., Kumar, V. (eds.): The Top Ten Algorithms in Data Mining. Data Mining and Knowledge Discovery. Chapman & Hall,CRC (2009)