Coevolution and learning symbolic concepts: statistical validation

Empirical statistical validation of co-evolutive machine learning systems

Filippo Neri
Department of Electrical and Computer Engineering, University of Naples, Naples, email: filippo.neri.email@gmail.com, phone: +39 081812032, Italy

ABSTRACT

The paper reports an empirical study done to statistically validate the preliminary findings obtained in previous research by the author on the small disjunct problem. Thus additional support to the working hypothesis that cooperative evolution (co-evolution) can be successfully applied in learning symbolic concepts and that co-evolution when carefully exploited can produce more robust classification rule (symbolic concepts) with higher statistical validity. In the paper we will compare the effect of applying a specific co-evolutive learning strategy with the results obtained by running a learning system without any coevolution. Thus we can measure the add-on effect produced by the coevolutive strategy. As learning systems we will use the system REGAL that combines distributed learning and genetic algorithms to find symbolic classifiers. As a future extension of this research, we note that the described co-evolutive strategy can be applied to other learning methods.

CCS CONCEPTS

• **Information systems**; • **Clustering and classification.**;

KEYWORDS

Additional Key Words and Phrases: Genetic algorithms, Concept learning, Cooperative learning

ACM Reference Format:

Filippo Neri. 2022. Coevolution and learning symbolic concepts: statistical validation: Empirical statistical validation of co-evolutive machine learning systems. In *2022 7th International Conference on Machine Learning Technologies (ICMLT) (ICMLT 2022), March 11–13, 2022, Rome, Italy*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3529399.3529437

1 INTRODUCTION

Symbolic concept learning [1] consists in finding a symbolic description, usually expressed in propositional or first order logic, that is able to correctly classify positive and negative instances of a given concept. From the computational point of view, concept learning consists in filtering large, potentially infinite, hypothesis spaces containing candidate concept descriptions. Therefore being able to use efficient and effecting searching algorithms become a must. Since several years, learning approaches based on Genetic Algorithms [2, 3] proved their potentialities on a variety of concept learning tasks.

Whereas one of the most efficient and well known method for symbolic concept learning are decision trees [4]. Unfortunately, decision trees suffer from the small disjunct problem. The small disjunct problem consists in the fact that decision trees lose statistical validity in the found concept description when the number of conditions increases. This is due to the way the learning set is partitioned during the decision tree construction thus reducing the number of supporting examples while the decision tree's depth increases.

On the contrary, genetic algorithms applied to concept learning can exploit the whole learning set to evaluate a concept description. Thus avoiding the small disjunct problem. However, a drawback of genetic algorithm is that they require significantly more time and computational power to run compared to decision trees. This is due to their need to evolve a set of candidate solutions time over time (generations) and this multi-solution approach requires intrinsically more time to be performed.

A general approach to deal with high computational cost is to use distributed computation. In fact, genetic algorithms can be easily parallelized and several populations can be made evolving in parallel by using or not a coordination mechanism among them. It is in the case of a coordination mechanism controlling the evolution of several populations that we can talk about co-evolution strategies [5–10].

Research on several forms of cooperative learning includes approaches like ensemble learning: boosting [11] and bagging [12]. These techniques combine a pool of classifiers in order to improve their separate and overall classification performances. Generally they exploit re-sampling or weighting of the learning instances in order to acquire different classifiers to be combined, and they are independent of the specific learning method used.

Also a combination of other artificial intelligent techniques, such as software agents, with evolutionary computation could be a promising testbed to expand this current work [13–18].

In the past, we investigated how the adoption of cooperative learning into the GA-based system REGAL [8, 17, 19–22] could produce a more efficient learning systems. We extend here our previous work on cooperative coevolution by expanding the performed experiments in the REGAL system by running a full fledge 10 fold cross validation and determing the confidence intervals for some of the collected measures so that we can improve the statistical validity of our findings. Therefore the novelty of this paper is in the experimental section and the reported findings while the rest

ICMLT 2022, March 11–13, 2022, Rome, Italy

ACM ISBN 978-1-4503-9574-8/22/03…$15.00
https://doi.org/10.1145/3529399.3529437

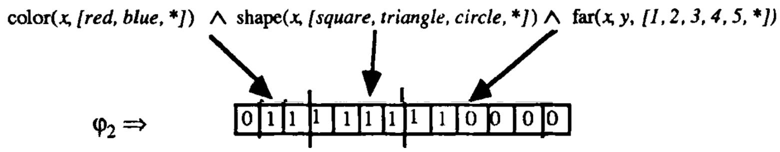

Figure 1: Bitstring and template.

of the paper is just a summary of the experimental environment, made up of our REGAL system, and thus contains descriptions of its inner working that have already being published in previous works. We however believe that is necessary to report a summary of some components of the REGAL system in this paper so that this work becomes self contained.

The paper is then organized as follows. In Section 2 and 3, the REGAL system and a cooperative learning strategy are described. In Section 4, the experimental framework is analyzed. In Section 5, the results are reported. Finally, the conclusion section ends the paper.

2 THE SYSTEM REGAL

We describe here in a synthetic format the REGAL [23] system in order to provide a comprehensive description of the experimental setting used in this work. The REGAL system learns relational disjunctive concept descriptions in a restricted form of First Order Logic by using cooperative evolution. In REGAL an individual is a conjunctive formula (encoded as a fixed length bitstring) and a subset of the individuals in the populations has to be determined to form a disjunctive description for the target concept. For the scope of this work, we concentrate on REGAL's cooperative architecture as a description of the system's other components have already been published.

REGAL's architecture is a network of N processes *GALearners*, coordinated by a *Supervisor* that imposes cooperation among the evolving populations. Metaphorically speaking, each *GALearner* realizes a niche, defined by a subset of the learning instances, where some species lives. Each $GALearners_n$ tries to find a description for a subset of the learning instances LS_n by evolving its population. In addition, the *GALearners* may perform migration (exchange) of individuals. The *Supervisor* coordinates the distributed learning activity by periodically assigning different subsets of the learning instances to the *GALearners*. The composition of these subsets depends on the specific cooperative policy used. Therefore, and this is the key point, a co-evolutive learning strategy can be implemented in REGAL quite easily by changing over the course of time the subset of learning instances assigned to each *GALearners*.

For the sake of completeness, we also show in fig. 1 how it is possible to code a special kind of propositional rule (or a restricted type of first order logic formula) in a bitstring. The bitstring allows to code disjunctive formulas quite simply by selecting or deselecting the disjunctive values that verify the predicates occurring in the formula. In particular, substring in the bitstring are used to codify set of values for each predicate or proposition and depending on the 1 or 0 values, the selected values are used to verify if the represented disjunctive formula can be verified or not on the learning instances. Conversely a given bitstring allows to select a set of instances from the learning set that verify the represented disjunctive formulas. Thus allowing to calculate the set of positive or negative instances that verify the given formula.

3 THE CO-EVOLUTIVE STRATEGY DESCRIBE THOSE STILL UNCOVERED

As said, REGAL's results depend on the emergence of an effective cooperative behavior among its learning processes. As described in the previous section, in the system, cooperation is achieved by periodically adjusting the learning sets assigned to each *GALearner*. Thus, the cooperative learning strategy that determines the composition of these learning sets becomes the very responsible for the learned concept description. As no a priori information is available on what is a successful assignment of learning instances, we experimented in the course of the years with several cooperative learning strategies based on different assumptions.

In this work, however, the focus is on a specific co-evolutive learning strategy that we named Describe Those Still Uncovered. This co-evolutive strategy is characterized by the fact that it forces the learners in dealing as soon as possible with examples difficult to cover. Essentially, as soon as a promising concept description emerges, the instances not covered by it are included into all the learning sets, whereas each covered instance is inserted into only one learning set. This approach should reduce the probability that "small disjuncts" appear. The detailed description of the strategy follows.

```
   CoopLSDTSU (Concept, E, C, w, LSn, N)
   /* Concept is the current concept description */
   /* E is the set of the available concept instances */
   /* C is the set of the available non concept instances */
   /* w is the class of the concept instances */
   /* LSn is the set of niches' definitions */
   /* N is the number of available  GALearners */
        LS = E U C
        NotCovered = E - Uj ∈ Concept PosCov(j, LS,  w)
       for n=1 to N
        LSn  = C U NotCovered
        endfor
Assigned = empty_set
c-list = < sort j ∈ Concept by decreasing value of c( j, LS, w) >
n=1
while not empty(c-list) do
k = FirstElem(c-list)
c-list = c-list - k
LSn = LSn U  e | e ∈ PosCov(j, LS, w) and e not Assigned)
Assigned = Assigned U LSn
n = (n + 1) mod N
endwhile
return(LSn )
```

By observing the algorithm, one can note that the procedure $\text{CoopLS}_{\text{DSTU}}$ includes the learning instances not covered by the current concept description into each new niche definition. Afterwards, the $\text{CoopLS}_{\text{DSTU}}$ strategy orders the formulas in the current concept description *Concept* according to their c-value. The c-value aims to evaluate how well the formula covers the positive instances and is obtained by multiplying the fitness function of the formulas by the number of positive instances covered. We refer to our previous work for a discussion of the fitness function whose formulas takes care of consistency and simplicity of the formula itself.

Then, the i-th GALearner get the task of learning a description covering the instances not covered by the first i-1 formulas in c-value list, plus the instances not covered by *Concept*.

According to this policy, the learning instances covered by *Concept* are included into only one niche definition. Instead, those instances not covered by any formula appear in all the niche definitions. As soon as an instance is covered, the number of niches, containing it, drops to one. Considering the extensions of the found concept description, this form of cooperation biases the learning activity towards descriptions that do not cover the same instances, i.e. they tend to have almost non overlapping extensions.

4 CHARACTERISTICS OF THE DATA DOMAIN: MUSHROOMS

As applicative domain for our experimental evaluation, we selected a well known concept learning dataset: the Mushrooms dataset [24]. The Mushrooms problem consists in recognizing mushrooms from the Agaricus and Lepiota families as Edible (the firsts) and Poisonous (the seconds). The dataset contains 8124 instances, 4208 of edible mushrooms and 3916 of poisonous ones. Each instance is described by a vector of 22 discrete attributes, each of which can assume from 2 to more than 6 different values. By defining a predicate for each <attribute, value> pair, the language template for this application could be coded as a bitstring of 126 bits.

The key point in selecting this dataset is that this problem is characterized by the absence in its hypothesis spaces of a purely conjunctive concept description and by the existence in its hypothesis spaces of at least a disjunctive concept description with perfect classificationpower. The knowledge about this hypothesis space comes from results appeared in the literature and experiments done over the years by the author.

From previous experiments, we know that the Mushrooms application admits as good description for the poisonous mushrooms concept that requires 15 conditions to be tested.

In a previous works, we used as experimental data three randomly selected sets of 4000 instances (2000 edible plus 2000 poisonous) to be used as learning sets, while the remaining 4124 instances have been used for testing.

We expand here the experimental activity, to improve the statistical confidence in our findings, by performing a 10 draws / 10 runs where for each of them 4000 instances (2000 edible plus 2000 poisonous) are to be used as learning set whereas the remaining 4124 instances are used as testsets. Note therefore that we decided not to use a cross-validation approach because we wanted to maintain backward compatibility in the novel experimental data so that the concept descriptions learned in new experiments and in old experiments performed over the years can be compared. Using a cross validation approach would made this comparison impossible as the extensions of the formulas would significantly change in sizes.

5 EMPIRICAL EVALUATION

As known in concept learning, the effectiveness of any concept learning system is primarily evaluated on the basis of the estimate of its average prediction error. However, in order to provide a closer insight in a system behavior, additional measures may be used, such as, for instance, measures accounting for the structure of the acquired concept description. The comparison of REGAL's performances in terms of its average prediction error has already been analyzed for instance in [8, 23, 25]. We are here interested in the qualitative evaluation of how cooperation affects the structure of the found concept descriptions. Consequently, we will study REGAL's behavior with and without a cooperative strategy at work and considering the effect of migration.

Given all the previous considerations, setting up a suitable experimental context involves selecting the characteristics of concept descriptions that should be measured. Of course as we want to expand the experimental evaluation that we performed in the past in order to increase its statistical validity, we are bound to use the metrics and the frame of the experimental settings that we used in our previous research.

We then remind that the metrics that we selected in previous research and that we will keep here are: (a) the average prediction error (ϵ) (b) the complexity (C) of a concept description defined as the number of conditions to be tested in order to verify it; (c) the number of conjuncts (NC) in the concept description; (d) the maximum (MXC), average (AVC) and minimum (SMC) number of positive examples covered by any conjunct in the concept description; (e) and the user waiting time (T), i.e. the cpu time of the slowest learner to complete its task.

In order to be able to compare the learned concept descriptions with respect to reasonable target ones, we chose an applicative domain (the Mushrooms domain) whose best concept descriptions are known thanks to the many experiments done by the research community and the authors in the past. These target concept descriptions are characterized by a perfect predictive power and by a low complexity value.

5.1 The Past and now Standard Experimental Setting

In this work, we used the usual parameter setting as reported in Table 1. We remind that the same experimental setting has to be used in order to be able to expand and thus compare past research with the current one. In the table, a migration rate of 0.5 means that half of one population migrates toward other GAs. The following configurations, corresponding to the parameter settings appearing in Table 1, have been considered:

- CONF1 (16 GAlearners and $\mu = 0.0$) - A basic distributed approach: 16 *GA_Learners*, each one evolving a population of 100 individuals. No coevolution is used thus the learners just evolve independently without any coordination. Also

Table 1: REGAL's configurations used in this work.

Parameter	Value
Population size	1600
Number of GA learners	16
Crossover probability p_c	0.6
Mutation probability p_m	0.0001
Migration rate μ	0.0 or 0.5
Generation limit	200
Generation gap	0.9
Cooperation	None/DTSU

this means that every learner exploits the whole learning set for all the duration of the run.

- CONF2 (16 GA learners and $\mu = 0.5$) - As CONF1 plus migration of individuals among the *GA_learners*. Therefore in this case the learners still use all the learning set during the run without modification and the only difference from CONF1 is that some individuals are echanged among learners.

In addition to CONF1 and CONF2, we add two configurations where we add the co-evolutive strategy DTSU previously described. Therefore in the latter two cases, the composition of the learning sets used by the 16 learners will change during REGAL's run according to the algorithm of the cooperative strategy that takes into account to so far found concept description.

In Table 2 and 3, the experimental results are reported. The leftmost column of the table shows the configuration's identifier. The other columns of the table contains the parameters already described plus the 'Cons & Compl' field that summarizes whether the learned concept description is complete and consistent on the learning set. Finally, the rows, with the value "Target", report the features of the target concept. For each configuration settings ten runs have been performed.

The experimental findings can be summarized as follows: Table 2 and 3 show that without a co-evolutive strategy, REGAL cannot learn a complete and consistent concept description. Yet the largest conjunct covering 1946 positive instances can be learned thus found under any experimental conditions. However, the smallest conjunct of a perfect concept description cannot be found when no coevolution is used.

Please note that when no co-evolution is used, REGAL finds a smallest conjunct covering 1139 positive instance, however this two conjuncts found are not covering, thus explaining, the whole set of positive examples. Therefore the third small conjunct is actually missing from the no-coevolution setup.

Proceeding, the coevolutive strategy DTSU always find a complete and consistent concept descrip- tion thus solving the learning problem of discriminating between poisonous and edible mushrooms. The found concept descriptions have all a perfect classification errors on the test set. We also observe that DTSU finds 4 and 5 conjuncts as concept description and that the smallest conjunct covers 317 instances. This is surprising. Especially considering that according to past esperiments with even more computational power and a variety of configuration settings, we thought that the largest smallest conjunct would cover 197 instances. Consider in fact the

Table 2: REGAL learning the "Poisonous mushrooms" concept. Data based on 10 runs.

CoopLS	μ	ND (avg)	MXD (avg)	SMD (best)
CONF1				
			1946 +/-0.0	1139
None	0.0	2 +/-0.0		
DTSU	0.0	5 +/-0.0	1946 +/-0.0	329
CONF2				
			1946 +/-0.0	1161
None	0.5	2 +/-0.0		
DTSU	0.5	4 +/-0.0	1946 +/-0.0	317
Target		3	1946	197

Table 3: REGAL learning the "Poisonous mushrooms" concept. Data based on 10 runs.

CoopLS	μ	T	e[%]	Cons. and Comp.
CONF1				
			0.02 +/-0.01	No
None	0.0	76 +/-7		
DTSU	0.0	73 +/-8	0.0 +/-0.0	Yes
CONF2				
			0.04 +/-0.01	No
None	0.5	97 +/-8		
DTSU	0.5	99 +/-9	0.0 +/-0.0	Yes
Target		-	0	Yes

Target line in the table that report the best concept description ever found in the past by the author. So with these experimental settings we have found an alternate form of target/best concept descriptions for the Mushrooms dataset one that either contains 4 or 5 conjuncts and whose smallest disjunct covers more than 300 instances.

The reader may ask why these two concept descriptions are better than the 'Target' one. Let us remember that the focus of this research is to deal with the small disjunct problem that is the fact that decision trees find concept description that lose statistical validity while increasing the number of conditions to be tested. Or, in other words, if the decision tree has a high depth value. This result in the fact that conjuncts found by the decision trees tends to be less predictive if they have more than few conditions to be tested. Thus losing predictive power also on the training set. This is why being able to deploy a learning system that is able to learn concept description with large extension is important: their conjunct will be more statistically supported by the learning set and thus have more predictive power on the test set as well.

A final observation, the run experiments confirm that using coevolution does not significantly increase the system running cost. The reason is that the computational cost of adjusting the learning set is minimal with respect to the computational cost of evolving a population of candidate solutions.

6 CONCLUSION

The paper reports an empirical study done to statistically validate the preliminary findings obtained in previous research by the author when tackling the small disjunct problem in symbolic machine learning. In particular, decision trees are greatly affected by the small disjunct problem that leads long concept descriptions, or rules with many conditions, or deep decision trees (in decision tree language), to lose statistical validity of the found rules.

In the paper, we show how coevolution can be used to both learn symbolic rules in a relative fast way by using genetic algorithms and how coevolution can be successful in dealing with the small disjunct problem by performing niche differentiation in the learning phase.

Our results add statistical validity to our past hypothesis about the efficacy of coevolution: the experimental results here are averaged over 10 runs for statistical validity. And special care has been taken to maintain backward comparability with previous experimental results.

As coevolution is independent of the specific learning algorithm used, here we use genetic algorithms, as a future extension of this paper, other learning methods could be also explored in their efficacy when combined with coevolution.

Further works include the application of co-evolution to parameter optimization [26] and to financial domains [27].

REFERENCES

[1] T. Dietterich and R. Michalski, "A comparative review of selected methods for learning from examples," in *Machine Learning, an Artificial Intelligence Approach* (J. Carbonell, R. Michalski, and T. Mitchell, eds.), Morgan Kaufmann, 1983.

[2] J. H. Holland, *Adaptation in Natural and Artificial Systems.* Ann Arbor, Mi: The University of Michigan Press, 1975.

[3] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading, Ma: Addison-Wesley, 1989.

[4] J. R. Quinlan, *C4.5: Programs for Machine Learning.* California: Morgan Kaufmann, 1993.

[5] P. Husbands and F. Mill, "A theoretical investigation of a parallel genetic algorithm," in *Fourth International Conference on Genetic Algorithms, (Fairfax, VA), pp. 264–270, Morgan Kaufmann, 1991.*

[6] W. D. Hillis, "Co-evolving parasites improve simulated evolution as an optimization procedure," in Artificial Life II (C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, eds.), vol. X, pp. 313–324, Santa Fe Institute, New Mexico, USA: Addison-Wesley, 1990 1992.

[7] M. Potter, The Design and Analysis of a Computational Model of Cooperative Coevolution. PhD thesis, Department of Computer Science. George Mason University, VA, 1997.

[8] F. Neri, First Order Logic Concept Learning by means of a Distributed Genetic Algorithm. PhD thesis, Department of Computer Science. University of Torino,Italy, 1997.

[9] J. L. Shapiro, "Does data-mod co-evolution improve generalization performances of evolving learners?," Lecture Notes in Computer Science, vol. LNCS 1498, pp. 540–549, 1998.

[10] F. Neri, "Relational concept learning by cooperative evolution," ACM Journal of Experimental Algorithmics, vol. 7, 2003.

[11] R. E. Schapire, "A brief introduction to boosting," in Sixteenth International Joint Conference on Artificial Intelligence, pp. 1401–1406, 1999.

[12] T. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," Machine Learning, vol. 40, pp. 139–158, 2000.

[13] I. García-Margariño, I. Plaza, and F. Neri, "Abs-mindburnout: An agent-based simulator of the effects of mindfulness- based interventions on job burnout," Journal of Computational Science, vol. 36, 2019.

[14] F. Neri, "Combining machine learning and agent based modeling for gold price prediction," in Artificial Life and Evolutionary Computation (S. Cagnoni, M. Mordonini, R. Pecori, A. Roli, and M. Villani, eds.), pp. 91–100, Springer, 2019.

[15] F. Neri, "Agent-based modeling under partial and full knowledge learning settings to simulate financial markets," AI Communications, vol. 25, no. 4, pp. 295–304, 2012.

[16] F. Neri, "How to identify investor's types in real financial markets by means of agent based simulation," in ICMLT 2021: Proceedings of the 2021 6th International Conference on Machine Learning Technologies. In press, 2021.

[17] F. Neri and I. Margariño, "Simulating and modeling the dax index and the uso etf financial time series by using a simple agent-based learning architecture," Expert Systems, vol. 37, no. 4, 2020.

[18] F. Neri, "Case study on modeling the silver and nasdaq financial time series with simulated annealing," in Trends and Advances in Information Systems and Technologies (Rocha, Álvaro *et al.*, ed.), pp. 755–763, Springer, 2018.

[19] M. Camilleri and F. Neri, "Parameter optimization in decision tree learning by using simple genetic algorithms," WSEAS Transactions on Computers, vol. 13, pp. 582–591, 2014.

[20] M. Camilleri, F. Neri, and M. Papoutsidakis, "An algorithmic approach to parameter selection in machine learning using meta-optimization techniques," WSEAS Transactions on Systems, vol. 13, no. 1, pp. 203–212, 2014.

[21] F. Neri, "Traffic packet based intrusion detection: decision trees and genetic based learning evaluation," WSEAS Transaction on Computers, vol. 4, no. 9, pp. 1017–1024, 2005.

[22] F. Neri, "Cooperative evolutive concept learning: an empirical study 559," WSEAS Trans. on Information Science and Applications, vol. 2, pp. 559–563, 2005.

[23] F. Neri and L. Saitta, "Exploring the power of genetic search in learning symbolic classifiers," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-18, no. 11, pp. 1135–1142, 1996.

[24] J. S. Schlimmer, "Concept acquisition through representational adjustement," Tech. Rep. TR 87-19, Dept. of Information and Computer Science, University of Californina, Irvine, CA, 1987.

[25] F. Neri and L. Saitta, "An analysis of the universal suffrage selection operator," Evolutionary Computation, vol. 4 (1), pp. 89–109, 1996.

[26] F. Neri, "Unpublished result: Mapping learning algorithms on data, a useful step for optimizing performances and their comparison. available at https://arxiv.org/abs/2107.06981,'' U.P.

[27] F. Neri, "Unpublished result: Domain specific concept drift detectors for predicting financial time series. submitted, available at https://arxiv.org/abs/2103.14079,'' tbd, U.P.