A Comparison of Multi-objective Grammar-Guided Genetic Programming Methods to Multiple Instance Learning

Amelia Zafra and Sebastián Ventura

Department of Computer Science and Numerical Analysis, University of Cordoba

Abstract. This paper develops a first comparative study of multi-objective algorithms in Multiple Instance Learning (MIL) applications. These algorithms use grammar-guided genetic programming, a robust classification paradigm which is able to generate understandable rules that are adapted to work with the MIL framework. The algorithms obtained are based on the most widely used and compared multi-objective evolutionary algorithms. Thus, we design and implement SPG3P-MI based on the Strength Pareto Evolutionary Algorithm, NSG3P-MI based on the Non-dominated Sorting Genetic Algorithm and MOGLG3P-MI based on the Multi-objective genetic local search. These approaches are tested with different MIL applications and compared to a previous single-objective grammar-guided genetic programming proposal. The results demonstrate the excellent performance of multi-objective approaches in achieving accurate models and their ability to generate comprehensive rules in the knowledgable discovery process.

1 Introduction

Multiple Instance Learning (MIL) introduced by Dietterich et al. [1] consists of generating a classifier that will correctly classify unseen patterns. The main characteristic of this learning is that the patterns are bags of instances where each bag can contain different numbers of instances. There exists information about the bags, a bag receives a special label, but the labels of instances are unknown. According to the standard learning hypothesis proposed by Dietterich et al. [1] a bag is positive if and only if at least one of its instances is positive and it is negative if none of its instances produce a positive result. The key challenge in MIL is to cope with the ambiguity of not knowing which of the instances in a positive bag are actually the positive examples and which are not. In this sense, this learning problem can be regarded as a special kind of supervised learning problem where the labeling information is incomplete. This learning framework is receiving growing attention in the machine learning community because numerous real-world tasks can be very naturally represented as multiple instance problems. Among these tasks we can cite text categorization [2], content-based image retrieval [3], image annotation [4], drug activity prediction [5,6], web index page recommendation [7], stock selection [5] and computer security [8].

E. Corchado et al. (Eds.): HAIS 2009, LNAI 5572, pp. 450–458, 2009.

The problem of evaluating the quality of a classifier, whether in MIL perspective or in traditional supervised learning, is naturally posed as a multi-objective problem with several contradictory objectives. If we try to optimize one of them, the others are reduced. All previously used proposals to solve this problem from a MIL perspective do not take into account the multi-objective problem and only obtain one optimal solution combining the different objectives to obtain a high quality classifier. However, this approach is unsatisfactory due to the nature of optimality conditions for multiple objectives. It is well-known that in the presence of multiple and conflicting objectives, the resulting optimization problem gives rise to a set of optimal solutions, instead of just one optimal solution. Multiple optimal solutions exist because no single solution can be a substitute for multiple conflicting objectives and it is shown that algorithms which consider the set of optimal solutions obtain better general results.

In this paper, a first comparative study of the most widely analyzed, compared and tested approaches under various problems and criteria which generate Pareto Optimal Front (POF) is elaborated. We design and implement classic multi-objective evolutionary algorithms using Grammar Guided Genetic Programming (G3P) and adapt them to handle multi-instance problems. Our proposals are the Strength Pareto Grammar-Guided Genetic Programming for MIL (SPG3P-MI) based on the Strength Pareto Evolutionary Algorithm (SPEA2)[9], the Non-dominated Sorting Grammar-Guided Genetic Programming for MIL (NSG3P-MI) based on the Non-dominated Sorting Genetic Algorithm (NSGA2) [10] and Multi-objective genetic local search with Grammar-Guided Genetic Programming for MIL (MOGLSG3P-MI) based on Multi-objective genetic local search (MOGLS)[11]. These algorithms represent classification rules in IF-THEN form which make it possible to determine if a bag is positive or negative and the quality of each classifier is evaluated according to two conflicting quality indexes, sensitivity and specificity. Computational experiments show that multi-objective techniques are robust algorithms which achieve better results than G3P-MI [12] other previously used technique based on G3P and a single-objective. Moreover, multi-objective proposals obtain classifiers which contain simple rules which add comprehensibility and simplicity in the knowledge discovery process.

The paper is organized as follows. In Section 2, a description of the approaches proposed is presented. In Section 3, experiments are conducted. Finally, conclusions and some possible lines for future research are discussed in Section 4.

2 Using Multi-objective G3P for Classification Rule Generation

In our approach, we use an extension of traditional GP systems, called grammar-guided genetic programming (G3P) [13]. G3P facilitates the efficient automatic discovery of empirical laws providing a more systematic way to handle typing using a context-free grammar which establishes a formal definition of the syntactical restrictions. The motivation to include this paradigm is that it retains a significant position due to its flexible variable length solution representation

and the low error rates that achieves both in obtaining classification rules, and in other tasks related to prediction, such as feature selection and the generation of discriminant functions. On the other hand, the main motivation to include multi-objective strategies in our proposals is due to the measurements to evaluate a classifier are conflictive, so if the value of any of them is maximized, the value of the others can be significantly reduced. Thus, it is very interesting to obtain the POF and introduce preference information to analyze which of them could be the best to classify new examples.

Multi-objective techniques for evolutionary computation have been widely used on classification topics where significant advances in results have been achieved [14]. If we evaluate its use in Genetic Programming (GP), we can find that it provides better solutions than those obtained using standard GP and lower computational cost [15,16].

In this section we specify different aspects which have been taken into account in the design of the these proposals, such as individual representation, genetic operators and fitness function. The main evolutionary process is not described because it is based on the well-known SPEA2 [9], NSGA2 [10] and MOGLS [11].

2.1 Individual Representation

In our systems, as G3P-MI [12], individuals express the information in the form of IF-THEN classification rules. These rules determine if a bag should be considered positive (that is, if it is a pattern of the concept we want to represent) or negative (if it is not).

If ($cond_B$(bag)) **then**
  the bag is an instance of the concept.
Else
  the bag is an instance of the concept.
End-If

where $cond_B$ is a condition that is applied to the bag. Following the Dietterich hypothesis, $cond_B$ can be expressed as:

$$cond_B(bag) = \bigvee_{\forall instance \in bag} cond_I(instance) \tag{1}$$

where $\vee$ is the disjunction operator, and $cond_I$ is a condition that is applied over every instance contained in a given bag. Figure 1 shows the grammar used to represent the condition of the rules.

2.2 Genetic Operators

The process of generating new individuals in a given generation of the evolutionary algorithm is carried out by two operators, crossover and mutator. Depending on the philosophy of the algorithm one or both will be used. In this section, we briefly describe their functioning.

⟨S⟩→⟨cond$_I$⟩
⟨cond$_I$⟩→⟨cmp⟩| **OR** ⟨cmp⟩ ⟨cond$_I$⟩| **AND** ⟨cmp⟩ ⟨cond$_I$⟩
⟨cmp⟩→ ⟨op-num⟩ ⟨variable⟩ ⟨value⟩| ⟨op-cat⟩ ⟨variable⟩
⟨op-cat⟩ → **EQ** | **NOT EQ**
⟨op-num⟩→ **GT** | **GE** | **LT** | **LE**
⟨term-name⟩ → *Any valid term in dataset*
⟨term-freq⟩ → *Any integer value*

Fig. 1. Grammar used for representing individuals' genotypes

Crossover Operator. This operator chooses a non-terminal symbol randomly with uniform probability from among the available non-terminal symbols in the grammar and two sub-trees (one from each parent) whose roots are equal or compatible to the symbol selected are swapped. To reduce bloating, if any of the two offspring is too large, they will be replaced by one of their parents.

Mutation Operator. This operator selects with uniform probability the node in the tree where the mutation is to take place. The grammar is used to derive a new subtree which replaces the subtree underneath that node. If the new offspring is too large, it will be eliminated to avoid having invalid individuals.

2.3 Fitness Function

The fitness function is a measure of the effectiveness of the classifier. There are several measures to evaluate different components of the classifier and determine the quality of each rule. We consider two widely accepted parameters for characterizing models in classification problems: sensitivity (Se) and specificity (Sp). Sensitivity is the proportion of cases correctly identified as meeting a certain condition and specificity is the proportion of cases correctly identified as not meeting a certain condition. Both are specified as follows:

$$sensitivity = \frac{t_p}{t_p + f_n}, \begin{cases} t_p & \text{number of positive bags correctly identified.} \\ f_n & \text{number of negative bags not correctly identified.} \end{cases}$$

$$specificity = \frac{t_n}{t_n + f_p}, \begin{cases} t_n & \text{number of negative bags correctly identified.} \\ f_p & \text{number of positive bags not correctly identified.} \end{cases}$$

We look for rules that maximize both $Sensitivity$ and $Specificity$ at the same time. Nevertheless, there exists a well-known trade-off between these two parameters because they evaluate different and conflicting characteristics in the classification process. Sensitivity alone does not tell us how well the test predicts other classes (that is, the negative cases) and specificity alone does not clarify how well the test recognizes positive cases. It is necessary to optimize both the sensitivity of the test to the class and its specificity to the other class to obtain a high quality classifier.

3 Experimental and Results

A brief description of the application domains used for comparing along with a description of the experimental methodology are presented in the next section. Then, the results and a discussion about the experimentation are detailed.

3.1 Problem Domains Used and Experimental Setting

The datasets used in the experiments represent two well-known applications in MIL, *drug activity prediction* which consists of determining whether a drug molecule will bind strongly to a target protein [1] and *content-based image retrieval* which consists of identifying the intended target object(s) in images [2] Detailed information about these datasets is summarized in Table 1. All datasets are partitioned using 10-fold stratified cross validation [17] on all data sets. Folds are constructed on bags, so that every instance in a given bag appears in the same fold. The partitions of each data set are available at *http:www.uco.es/grupos/ ayrna/mil*.

Table 1. General Information about Data Sets

Dataset	Bags			Attributes	Instances	Average Bag Size
	Positive	*Negative*	*Total*			
Musk1	47	45	92	166	476	5.17
Musk2	39	63	102	166	6598	64.69
Mutagenesis-Atoms	125	63	188	10	1618	8.61
Mutagenesis-Bonds	125	63	188	16	3995	21.25
Mutagenesis-Chains	125	63	188	24	5349	28.45
Elephant	100	100	200	230	1391	6.96
Tiger	100	100	200	230	1220	6.10
Fox	100	100	200	230	1320	6.60

The algorithms designed have been implemented in the JCLEC software [18]. All experiments are repeated with 10 different seeds and the average results are reported in the results table in the next section.

3.2 Comparison of Multi-objective Strategies

In this section, we compare the different multi-objective techniques implemented, MOGLSG3P-MI, NSG3P-MI and SPG3P-MI. In a first section a quantitative comparison of the performance of different multi-objective algorithms is carried out. Then, the different multi-objective techniques are compared with the accuracy, sensitivity and specificity results of G3P-MI, a previous single-objective G3P algorithm [12].

Analysis of the quality of Multi-Objective strategies. The outcome in the multi-objective algorithms used is an approximation of the Pareto-optimal front (POF). An analysis of the quality of these approximation sets is evaluated to

compare the different multi-objective techniques. Many performance measures which evaluate different characteristics have been proposed. Some of the most popular performance measurements as spacing, hypervolume and coverage of sets [19] are analyzed in this work and their average results on the different data sets studied are shown in Table 2. The spacing [19] metric describes the spread of non-dominated set. According to the results showed the non-dominated front of NSG3P-MI has all solutions more equally spaced than the other algorithms. The hypervolume indicator [19] is defined as the area of coverage of non-dominated set with respect to the objective space. The results show that the non-dominated solutions of NSG3P-MI cover more area than the other techniques. Finally, coverage of two sets [19] is evaluated. This metric can be termed relative coverage comparison of two sets. The results show that NSG3P-MI obtains the highest values when it is compared with the other techniques, then by definition the outcomes of NSG3P-MI dominate the outcomes of the other algorithms. Taking into account all the results obtained in the different metrics, NSG3P-MI achieves a better approximation of POF than the other techniques.

Table 2. Analysis of quality of POFs considering average values for all data sets studied

Algorithm	Hypervolume (HV)	Spacing (S)	Two set Coverage (CS)	
MOGLSG3P-MI	0.844516	0.016428	CS(MOGLSG3P-MI,NSG3P-MI)	0.357052
			CS(MOGLSG3P-MI,SPG3P-MI)	0.430090
NSG3P-MI	0.890730	0.007682	CS(NSG3P-MI,MOGLSG3P-MI)	0.722344
			CS(NSG3P-MI,SPG3P-MI)	0.776600
SPG3P-MI	0.872553	0.012290	CS(SPG3P-MI,MOGLSG3P-MI)	0.508293
			CS(SPG3P-MI,NSG3P-MI)	0.235222

Comparison Multi-Objective strategies with a Single-Objective previous version. We compare the results of accuracy, sensitivity and specificity of different multi-objective techniques implemented with the results of a previous single-objective G3P algorithm [12]. The average results of accuracy, sensitivity and specificity for each data set are reported in Table 3. The Friedman test [20] is used to compare the different algorithms. The Friedman test is a nonparametric test that compares the average ranks of the algorithms. These ranks let us know which algorithm obtains the best results considering all data sets. In this way, the algorithm with the value closest to 1 indicates the best algorithm in most data sets. The ranking values for each measurement are also shown in Table 3.

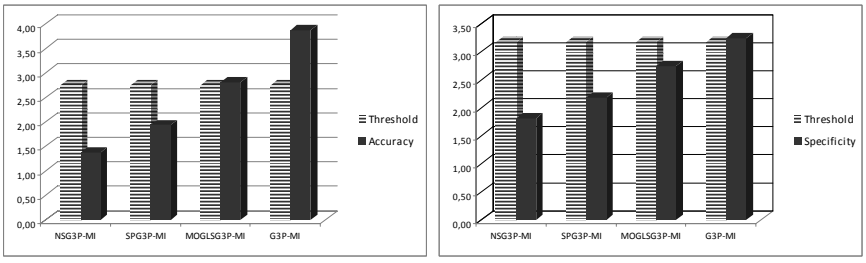
The Friedman test results are shown in Table 4. This test indicates that these are significantly differences both in accuracy and specificity measurements and there is no significant difference for sensitivity measurement. A post-hoc test was used, the Bonferroni-Dunn test [20], to find significant differences occurring between algorithms. Figure 2(a) shows the application of this test on accuracy. This graph represents a bar chart, whose values are proportional to the mean

Table 3. Experimental Results

Algorithm	MOGLSG3P-MI			NSG3P-MI			SPG3P-MI			G3P-MI		
	Acc	*Se*	*Sp*	*Acc*	*Se*	*Sp*	*Acc*	*Se*	*Sp*	*Acc*	*Se*	*Sp*
Elephant	0.8900	0.8700	0.9100	0.9400	0.9400	0.9400	0.9250	0.9400	0.9100	0.8800	0.9300	0.8300
Tiger	0.8850	0.9400	0.8300	0.9350	0.9200	0.9500	0.9200	0.9200	0.9200	0.8700	0.9400	0.8000
Fox	0.7600	0.7800	0.7400	0.7800	0.8600	0.7000	0.8350	0.8900	0.7800	0.7050	0.7900	0.6200
MutAtoms	0.8421	0.9385	0.6333	0.9158	0.9462	0.8500	0.8790	0.9308	0.7667	0.8526	0.8462	0.8167
MutBonds	0.8421	0.9077	0.7000	0.8737	0.9231	0.7667	0.8684	0.9308	0.7333	0.8210	0.8462	0.7833
MutChains	0.8737	0.9462	0.7167	0.9211	0.9462	0.8667	0.9053	0.9000	0.9167	0.8105	0.9231	0.7333
Musk1	0.9778	0.9600	1.0000	1.0000	1.0000	1.0000	0.9667	0.9800	0.9500	0.9445	1.0000	0.9000
Musk2	0.9400	0.9500	0.9333	0.9301	0.9607	0.9095	0.9400	0.9750	0.9167	0.8800	1.0000	0.9000
RANKING	2.8125	3.0000	2.7500	1.3750	2.0000	1.8125	1.9375	2.3750	2.1875	3.8750	2.6250	3.2500

Table 4. Results of the Friedman Test (p=0.1)

	Valor Friedman	Valor $\chi_2(1-\alpha=0.1)$	Conclusion
Acc	17.1375	4.642	Reject null hypothesis
Se	2.5500	4.642	Accept null hypothesis
Sp	5.7375	4.642	Reject null hypothesis



(a) Bonferroni-Dunn for Accuracy (b) Bonferroni-Dunn for Specificity

Fig. 2. Bonferroni Dunn Test ($p < 0.1$)

rank obtained from each algorithm. This test sets a *Threshold* (represented with one of the grated bars); those values that exceed this bar are algorithms with significantly worse results than the control algorithm (associated in this case with NSG3P-MI because it is the lowest rank value). The threshold in this case is fitted to 2.7486 (with, $1-\alpha = 0.1$). Observing this figure, the algorithms that exceed the threshold determined by Bonferroni are MOG3P-MI and G3P-MI, therefore they could be considered worse proposals.

With respect to the specificity measurement, Figure 2(b) shows the application of Bonferroni-Dunn post-hoc test on it. The threshold in this case is 3.1861 (with, $1 - \alpha = 0.1$). Observing this figure, the algorithm that exceeds the threshold determined by Bonferroni is G3P-MI, again NSG3P-MI is the best proposal.

We can conclude that statistically there are hardly any differences between multi-objective proposals, except for the accuracy values of the MOGLS algorithm. On the other hand, the differences are more noticeable for the single-objective G3P algorithm that obtains worse results than the rest of the techniques for all measurements and for accuracy and specificity obtains significant differences statistically. Moreover, a better trade-off between the different measurements can be seen in the multi-objective techniques.

4 Conclusions and Future Works

This paper has done a first comparative study of multi-objective evolutionary algorithms on MIL. To do so, the renowned algorithms, SPEA2, NSGA2 and MOGLS have been adapted to work with a G3P paradigm and to handle a MIL scenario. The comparison between the different multi-objective techniques and a previous single-objective G3P algorithm (G3P-MI) has shown that all multi-objective proposals obtain more accurate models. The Friedman test determine that NSG3P-MI is the best proposal with respect to the rest of the algorithms for all measurements considered. Statistically, it can be concluded that there are significant differences between the algorithms with respect to accuracy and specificity values. For these values, a post-test is carried out and this the Bonferroni-Dunn test concludes that G3P-MI is considered to be statistically worse algorithm for both measurements.

This is only a preliminary study and there are still some forthcoming considerations. Thus, it would be interesting to make a more detailed study which evaluate the performance of multi-objectives proposals. Moreover, it would be interesting to do a thorough investigation involving the most representative MIL algorithm in the rest of paradigms used in MIL.

References

1. Dietterich, T.G., Lathrop, R.H., Lozano-Perez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artifical Intelligence 89(1-2), 31–71 (1997)
2. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: NIPS 2002: Proceedings of Neural Information Processing System, Vancouver, Canada, pp. 561–568 (2002)
3. Pao, H.T., Chuang, S.C., Xu, Y.Y., Fu, H.: An EM based multiple instance learning method for image classification. Expert Systems with Applications 35(3), 1468–1472 (2008)
4. Yang, C., Dong, M., Fotouhi, F.: Region based image annotation through multiple-instance learning. In: Multimedia 2005: Proceedings of the 13th Annual ACM International Conference on Multimedia, New York, USA, pp. 435–438 (2005)
5. Maron, O., Lozano-Pérez, T.: A framework for multiple-instance learning. In: NIPS 1997: Proceedings of Neural Information Processing System 10, Denver, Colorado, USA, pp. 570–576 (1997)
6. Zhou, Z.H., Zhang, M.L.: Solving multi-instance problems with classifier ensemble based on constructive clustering. Knowledge and Information Systems 11(2), 155–170 (2007)

7. Zafra, A., Ventura, S., Romero, C., Herrera-Viedma, E.: Multiple instance learning with genetic programming for web mining. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 919–927. Springer, Heidelberg (2007)
8. Ruffo, G.: Learning single and multiple instance decision tree for computer security applications. PhD thesis, Department of Computer Science. University of Turin, Torino, Italy (2000)
9. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Gloriastrasse 35 (2001)
10. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)
11. Jaszkiewicz, A., Kominek, P.: Genetic local search with distance preserving recombination operator for a vehicle routing problem. European Journal of Operational Research 151(2), 352–364 (2003)
12. Zafra, A., Ventura, S.: G3P-MI: A genetic programming algorithm for multiple instance learning. In: Information Science. Elsevier, Amsterdam (submitted)
13. Whigham, P.A.: Grammatically-based genetic programming. In: Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications, Tahoe City, California, USA, pp. 33–41 (1995)
14. Shukla, P.K., Deb, K.: On finding multiple pareto-optimal solutions using classical and evolutionary generating methods. European Journal of Operational Research 181(3), 1630–1652 (2007)
15. Parrott, D., Xiaodong, L., Ciesielski, V.: Multi-objective techniques in genetic programming for evolving classifiers. In: IEEE Congress on Evolutionary Computation, vol. 2, pp. 1141–1148 (September 2005)
16. Mugambi, E.M., Hunter, A.: Multi-objective genetic programming optimization of decision trees for classifying medical data. In: KES 2003: Knowledge-Based Intelligent Information and Engineering Systems, pp. 293–299 (2003)
17. Wiens, T.S., Dale, B.C., Boyce, M.S., Kershaw, P.G.: Three way k-fold cross-validation of resource selection functions. Ecological Modelling 212(3-4), 244–255 (2008)
18. Ventura, S., Romero, C., Zafra, A., Delgado, J.A., Hervás, C.: JCLEC: A java framework for evolutionary computation soft computing. Soft Computing 12(4), 381–392 (2008)
19. Coello, C.A., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems. In: Genetic and Evolutionary Computation, 2nd edn. Springer, New York (2007)
20. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)