Improving Recruitment Effectiveness using Genetic Programming Techniques

[Extended Abstract]

Douglas A. Augusto LNCC, Petrópolis, RJ, Brazil douglas@Incc.br Heder S. Bernardino UFJF, Juiz de Fora, MG, Brazil heder@ice.ufjf.br Helio J.C. Barbosa LNCC, Petrópolis, RJ, Brazil UFJF, Juiz de Fora, MG, Brazil hcbm@Incc.br

ABSTRACT

A real-world problem, namely to improve the recruitment effectiveness of a certain company, is tackled here by evolving accurate and human-readable classifiers by means of grammar-based genetic programming techniques.

Categories and Subject Descriptors

I.5.2 [Pattern Recognition]: [Design Methodology, Classifier Design and Evaluation]; H.2.8 [Database Management]: [Database Applications, Data mining]

General Terms

Design, Algorithms, Experimentation

Keywords

Real-world; competition; machine learning; data classification; formal grammar; distributed genetic programming

1. INTRODUCTION

In early 2012, Aspiring Minds (www.aspiringminds.in) proposed a real-world machine-learning challenge: to improve the recruitment effectiveness of a certain company.

Here we will describe the winning methodology, the solution obtained, and the insights gained from analyzing the inferred models. Our methodology includes two evolutionary computation techniques which are able to produce solutions in symbolic form: distributed grammar-based genetic programming (GGP), and grammatical evolution (GE).

2. PROBLEM STATEMENT

In spite of hiring employees through a comprehensive recruitment process involving numerous tests and biographical information, a certain company found out—after one year of activity—that the performance of many of those individuals turned out to be rather disappointing. Thus, it is important to find ways to predict such unsatisfactory outcomes.

First, the company assessed and classified the employees into three categories according to objective and subjective evaluations: best-performer (BP), mid-performer (MP), and low-performer (LP). With those ratings at hand and with the original test scores as well as biographical information

Copyright is held by the author/owner(s).

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands. ACM 978-1-4503-1964-5/13/07.

of 950 employees, now the company wants to come up with an improved hiring mechanism that would combine those data and then predict the medium-term performance of a candidate at the earliest opportunity, that is, at the selection process. Hence, the goal is to develop predictors capable of foreseeing a candidate's performance in order to avoid hiring potential low-performance employees. Three aspects were adopted to evaluate the entries: prediction accuracy, solution interpretability, and methodology employed. The main attractiveness of the classifiers' interpretability is that it facilitates knowledge discovery, allowing the company to get insights about the underlying factors that govern the medium-term performance of the employees.

The prediction accuracy was calculated using a validation data set—whose class labels were left undisclosed by Aspiring Minds—over two cost matrices, representing both the conservative and liberal policies (Table 1). The original

		Predicted				Predicted		
		\mathbf{LP}	\mathbf{MP}	BP		\mathbf{LP}	\mathbf{MP}	BP
ıal	\mathbf{LP}	0	-2	-4	LP	0	-1	-2
.tu	\mathbf{MP}	-1	0	-1	MP	-1	0	-1
Ā	BP	-2	-1	0	BP	-4	-2	0

Table 1: Conservative and liberal policies' costs.

problem's data set featuring information about 950 candidates was shuffled and separated into two sets: a training data set containing 666 instances, and a validation data set with the remaining 284 instances. The data sets have 30 input attributes in total (plus the output attribute for the training data set, representing the expected classes). To design the classifiers' grammar so that only sensible operations among data are allowed, each attribute was assigned to one of the data types: *numeric, ordinal, or nominal.*

Several records with missing values were found in the provided dataset, but only those with missing data in the classification field were removed. In the remaining cases, which only affected numeric attributes, missing values were replaced by the *mean* value of the corresponding attribute.

3. METHODOLOGY & TECHNIQUES

GGP techniques excel where there is hardly any alternative approach: the automatic generation of programs in arbitrary human-readable languages. By interpreting and analyzing the learned models, their rules and relations, it is often possible to understand at least some of the underlying factors that relate dependent and independent variables.

Algorithm	1:	Algorithm	for	a	conservative	classifier.
0						

if $(CP > 70.15 \text{ and } AS3 > 45)$
or $(DS1 > 45$ and $State \neq D$) then
return BP;
else if $QA3 + DS1 < DT4 + 20$ then
return LP;
else
return MP:

Working along with each other, two techniques, a distributed GGP (DGGP) (our implementation, *gpclassifier*, is Free Software and can be found at gpclassifier.sf.net) and GE were applied to the problem.

Initially we have used GE, which managed to infer very compact and easily understandable models. This gave us a hint about the correlation between the classifier's complexity and its accuracy. Then, we thought we could decrease the classification error without compromising the readability, so we tried to relax the size of the candidate classifiers. Unfortunately, more complex solutions would translate into more computational effort, and our GE's implementation, which is sequential, was not prepared for such a heavy workload. Therefore, we set up the desirable complexity level in DGGP and let it run on a high-performance computing environment until classifiers with good trade-off between complexity and accuracy were evolved, for both policies considered.

The classifiers were evolved according to a weighted error function defined by the particular policy, conservative or liberal, in terms of a cost matrix. More specifically, under a certain policy, the classifier's fitness is the sum of the per-class negative score of each misclassified individual.

4. INFERRED MODELS

Algorithms 1 and 2 present, respectively, the inferred conservative and liberal models. The attributes used are: College Percentage (CP), Analytical Skills 1 (AS1), Analytical Skills 3 (AS3), Domain Skills 1 (DS1), Domain Test 4 (DT4), Quantitative Ability 3 (QA3), English 4 (E4), and State.

The inferred conservative model (Algorithm 1) is composed by only two conditionals, which are respectively responsible for (i) indicating if a certain individual corresponds to a BP case; and (ii) when this individual is not classified as BP, check if it is an LP case. If it is not, then it is assumed as being MP. With respect to the first conditional, one can verify that a given individual is classified as BP when presenting a good performance in both "CP" (> 70.15)and "AS3" (> 45) attributes. Moreover, when a good performance is verified for "DS1" (> 45) in an individual that does not come from a particular "State" (D), then it is also classified as BP. It is then clear the importance of achieving high scores in the attributes which correspond to a basic evaluation concept, such as "CP", or performance measures of interest, such as "AS3" or "DS1". However, one can notice that for individuals not coming from state "D", a good performance in "DS1" suffices to rank them as BP thus suggesting a negative bias towards that particular state. The classification of the remaining individuals, that is, as LP or MP, is not so clearly interpretable as in the previous case. Now the choice depends on a combination of the values in "QA3", "DS1", and "DT4". A low combined performance in "QA3" and "DS1" when compared to the performance in

Algorithm 2: Algorithm for a liberal classifier.			
if $QA3 + DS1 + AS3 + 13 < E4 + DT4$ then			
return LP;			
else if $CP \ge 75.3$ or $AS1 \ge 55$ or $State \ne D$ then			
\mathbf{ret}	urn BP;		
else			
\mathbf{ret}	urn MP;		

"DT4" tends to indicate an LP classification. Again, it is easy to understand the preference for performance metrics of interest, such as "QA3" and "DS1". However, an inverse correlation with "DT4" is not intuitive. In fact, observing the histograms of the LP and MP individuals with respect to "DT4" it is possible to verify a counter-intuitive tendency in the individuals classified as MP in achieving lower performance in this test than the ones classified as LP.

The liberal model (Algorithm 2) is also composed by only two conditionals. The first conditional evaluates whether the given individual is LP; if not, the second one tests whether he or she belongs to BP. If both conditionals fail, the individual is classified as MP. The first rule indicates that when the sum of the scores in the attributes "QA3", "DS1", and "AS3", plus the constant 13, is still lower than the sum of the "E4" and the "DT4" scores, then the individual should be classified as an LP. As in the conservative model, the same counter-intuitive correlation of higher values for "DT4" as a condition for LP classification was observed. Now, in the liberal model, the score in "E4" also plays a similar role to that of "DT4" in the classification as LP. Inspection of the histograms of "E4" also indicates the same (although smaller) tendency that individuals classified as MP obtain worse results when compared to those classified as LP. The classification of the individuals not belonging to the LP class happens to be more understandable. In fact, high scores in either "CP" (≥ 75.3) or "AS1" (≥ 55) define the individual as BP. Another observed condition for the same (BP) classification is when the individual does not come from the "D" State. Again, the preference for good scores on basic metrics (such as "CP") as well as those of interest (like "AS1"), is clear, but the same bias exists with respect to the individuals coming from the "D" State.

Although designed using different scores rules, both classifiers tend to predict the best-performing class to the individuals with high scores in "CP", "AS1", "AS3", and "DS1" as well as those not coming from the "D" State. Also, the low-performing class is assigned to those with a low performance in "QA3" and "DS1".

5. CONCLUSION

Using GP's (i) symbolic human-readable solutions, (ii) fully adjustable hypothesis space through the definition of the language and its bias via formal grammars, and (iii) highdegree of parallelism, one might produce excellent models for real-world problems, both in accuracy and simplicity.

6. ACKNOWLEDGMENTS

We would like to congratulate Aspiring Minds for its initiative, and the assistance provided. The authors acknowledge the support from CNPq (grants 308317/2009-2 and 300192/2012-6) and FAPERJ (grant E-26/102.025/2009).