A General Approach to Automatic Programming Using Occam's Razor, Compression, and Self-Inspection

Extended Abstract

Peter Galos, Peter Nordin, Joel Olsén, and Kristofer Sundén Ringnér

Complex Systems Group
Department of Physical Resource Theory
Chalmers University of Technology
S-412 96 Göteborg, Sweden

This paper describes a novel general method for automatic programming which can be seen as a generalization of techniques such as genetic programming and ADATE. The approach builds on the assumption that data compression can be used as a metaphor for cognition and intelligence. The proof-of-concept system is evaluated on sequence prediction problems. As a starting point, the process of inferring a general law from a data set is viewed as an attempt to compress the observed data. From an artificial intelligence point of view, compression is a useful way of measuring how deeply the observed data is understood. If the sequence contains redundancy it exists a shorter description i.e. the sequence can be compressed. Consider the sequence:

$$ABCDEABCDEABCDEABCDEABC$$

This sequence could be described with a program.

```
for i = 1..5 loop
   print 'ABCDE'
end loop
print 'ABC'
```

```
for i = 1..infinity loop
   print 'ABCDE'
end loop
```

The principle of Occam's Razor states that from two possible solutions (cf. the programs above) to a problem the simpler one (the program to the right) should be chosen. According to algorithmic information theory, the information content of the simplest program that without any prior information describes a data series is called the Kolmogorov complexity. It has been shown that prediction of the continuation of a sequence is more likely by a simpler program [VL1]. Solomonoff's induction method [So1] is used to produce the target data as well as a continuation the target data. Since it is not known whether a program will halt or not before execution, the Kolmogorov complexity is in general not computable. However, it can be approximated, in this case by programs consisting of recursively applied compression methods. The learning problem is therefore turned into a search problem in the space of all possible permutations of the set of compression methods. The fitness criterion is based on algorithmic information theory [Sh1].

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2724, pp. 1806–1807, 2003.

It is possible to search the space of hypotheses randomly in order to find a solution. However, a method that takes into account previous search results before suggesting a new hypothesis can give rice to an accelerated search process. In order to achieve self-inspection, a prediction algorithm is used by examining the algorithm's own progress. This pattern search in the history of previous programs' performance, enables the algorithm to predict the most likely way to search for future hypotheses. In a way, this is a reinforcement problem and thus just another sequence prediction problem. An advantage of this self-inspection is that no additional theory or learning algorithm is needed. To achieve an accurate measure of the information content of a program a registry machine is used. Table 1 shows some examples of predictions[1] using the approach proposed in this paper, were the "least complex programs found" column presents combinations of different compression methods.

Table 1. Results of predicting data series using the algorithm

Data sequence	Least complex program found	Prediction
1. 5, 10, 15, 20, 25	Delta→SRP	30, 35, 40, 45,...
2. 2, 4, 8, 16, 32	LCB→Delta→SRP	64, 128, 256, 512,...
3. 0, 9, 3, 4, 13, 7	Delta→SRP	8, 17, 11, 12, 21,...
4. 1, 2, 2, 3, 3, 3, 4, 4, 4, 4	RL→Delta→SRP	5, 5, 5, 5, 5, 6,...
5. 0, 3, 12, 21, 48, 75, 102	Delta→LCB→RL→Delta→SRP	183, 264, 345,...
6. 2, 2, 3, 2, 2, 4, 2, 2, 5, 2, 2, 6	N-gram→PWC→Delta→SRP	2, 2, 7, 2, 2, 8, 2, 2,...
7. 0, 1, 3, 6, 10, 15, 21, 28	Delta→Delta→SRP	36, 45, 55, 66, 78,...
8. 1, 1, 2, 1, 2, 3, 1, 2, 3	LZSS→PWC→Delta→SRP	4, 1, 2, 3, 4, 5, 1, 2, 3,...
9. 0, 1, 2, 4, 5, 7, 10, 11, 13, 16	Delta→LZSS→PWC→Delta→SRP	20, 21, 23, 26, 30,...

In conclusion the algorithm is simple and requires little domain specific input which makes it able to predict a wide range of input sequences. Furthermore, no prior knowledge is necessary. In addition, the paper pioneers the implementation and verification of the theory [VL1] concerning the connection between compression and learning. Much work on theory and evaluation is still needed in order to transform the method into a practical tool with the same wide applicability as for instance genetic programming. However, the firm theoretical foundation holds promise for a very efficient method and advantageous scaling properties.

References

[Sh1] Shannon, C.E.: A Mathematical Theory of Communication The Bell System Technical Journal Vol. 27, (1948) pp. 13

[So1] Solomonoff, R., J.: A preliminary report on a general theory of inductive inference. (1960) pp. 11–14

[VL1] Vitányi, P., Li, M.: Simplicity, Information, Kolmogorov Complexity, and Prediction. (1998) pp. 3

[1] Computed within 4 minutes on an Intel PII/133MHz