Transfer Learning: A Building Block Selection Mechanism in Genetic Programming for Symbolic Regression

Brandon Muller, Harith Al-Sahaf, Bing Xue, and Mengjie Zhang
School of Engineering and Computer Science
Victoria University of Wellington, P.O. Box 600
Wellington 6140, New Zealand
{brandon.muller,harith.al-sahaf,bing.xue,mengjie.zhang}@ecs.vuw.ac.nz

ABSTRACT

In machine learning, transfer learning is concerned with utilising prior knowledge as a way to improve the process of training a new model in a different, but related, domain. Transfer learning has been shown to be beneficial across a large set of problems. One of the main questions any transfer learning approach must address is "What to transfer?". This paper proposes a new transfer learning method in genetic programming (GP) to improve solving symbolic regression problems by extracting all potentially good and unique building blocks from a source problem. The proposed method is compared against standard GP and a state-of-the-art GP method on ten regression datasets. The experimental results show that the proposed method has achieved significantly better or comparable performance to that of the competitive methods. Furthermore, the proposed method shows better initial population and convergence compared to the other methods.

CCS CONCEPTS

• **Computing methodologies** → **Genetic programming**; *Supervised learning by regression*; *Transfer learning*;

KEYWORDS

Genetic programming, building blocks, transfer learning

ACM Reference Format:
Brandon Muller, Harith Al-Sahaf, Bing Xue, and Mengjie Zhang. 2019. Transfer Learning: A Building Block Selection Mechanism in Genetic Programming for Symbolic Regression. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3322072

1 INTRODUCTION

With enough data and time, machine learning can be used to construct models that are capable of solving a broad range of problems. However, in practice, often there are limitations on the amount of available data and time. This could lead to the resulting models performing poorly. To address this problem, the knowledge of previously trained models can be used, which has an impact on reducing the amount of data and time required. The final accuracy of the new models could also be improved. This process of transferring existing knowledge in some manner from a source domain to a target domain is called transfer learning [5]. There are three main questions that all transfer learning techniques must address — "What to transfer?", "How to transfer?", and "When to transfer?" [5]. All of these questions are still topics of active research. Several transfer learning techniques have been proposed to address these questions, which have proven to be useful in a broad range of fields such as binary problems [1] and image classification [2].

Genetic programming (GP) is an evolutionary computation technique that uses genetic operators to generate and optimise programs to solve a specific task. Typically, the tree representation is used in GP to represent an individual (solution), which makes this technique very flexible and allows GP to be largely extendable since more functions can always be added. Furthermore, the behaviour of the resulting model from GP can be interpreted.

Most of the transfer learning techniques for GP involve transferring (sub-)trees from a subset of individuals. These transferrals can be used to initialise the initial population for the target problem [3], or as a constant influence by using them throughout the evolutionary process [4]. The three main questions of transfer learning can be made more specific in this case. The important questions become: "Which individuals to consider?", "Which (sub-)trees should be transferred?", and "How are the transferrals used?".

O'Neill *et al.* [4] proposed a transfer learning technique for GP, which considers the similarities of different solutions evolved on different problems. The algorithm selects the best solutions, extracts all of the common sub-trees between them, and uses them as building blocks in the target problem. The way these blocks are acquired and used was an improvement over the previous methods. However, despite the improvements demonstrated, there are still some limitations, such as the requirement of having two separate GP runs. Hence, such solution cn be very expensive. Furthermore, two separate source problems are needed to generate an effective set of building blocks, which might not always be available.

This research aims to further explore and proposes a novel building blocks selection method in GP. The effectiveness of the proposed method to tackle regression problems will be investigated and compared to standard GP and a state-of-the-art method.

2 THE PROPOSED METHOD

The much less restrictive selection process of the building blocks in the proposed method, *Primitives from Sub-trees* (PST), represents the main difference between this method and existing ones.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3322072

Table 1: The mean squared error on ten regression problems.

Problem	STDGP	CSRP (S0)	CSRP (S1)	PST (S0)		PST (S1)	
Trig-1	2.12E+18	3.69	1.87	0.9	(+++)	2.44	(++=)
Trig-2	1.03E+14	49.63	3.87	2.26	(+++)	4.01	(++=)
Trig-3	7.34E+18	7.19	5.52	2.68	(+++)	2.33	(+++)
Trig-4	1.81	0.93	0.93	1.99	(= - -)	1.64	(= - -)
Trig-5	4.11	1.19	1.03	2.39	(+ - -)	1.68	(+ - -)
Poly-1	1.41E+05	622	48.24	31.18	(++=)	4.14E+05	(- = -)
Poly-2	48.63	47.93	7.72	9.91	(++=)	8.72E+04	(- = -)
Poly-3	182.71	158	47.61	39.33	(++=)	8.31E+05	(= - -)
Poly-4	1.13E+06	1.15E+06	104.02	529.1	(=+=)	1.26E+04	(= - -)
Poly-5	1.52E+07	4.31E+10	3.34E+16	2.59E+03	(==+)	1.37E+07	(= - =)

Building blocks are collected from the best solution ot a source problem. Each sub-tree of the best solution is a potential building block. If the potential building block has a depth greater than some threshold (in this case 4), it will be discarded to prevent the transferred building block from being too large. The larger a building block is, the more expensive it is to be evaluated. The potential building blocks are then filtered for uniqueness (if two blocks output the same values for the same inputs, then the larger block is discarded). After this filtering, all of the remaining building blocks are converted into functions by replacing their leaf nodes with inputs, and then transferred to the function set of the target problem.

The building blocks are used in the same way that other functions in the function set are. This means that when generating a new population, or creating a tree to be used in mutation, building blocks have the potential to be involved. Furthermore, the terminal set, function set, parameter settings, and fitness function (mean absolute error) are all the same as that in [4].

3 EXPERIMENT DESIGN AND RESULTS

Both standard GP (STDGP), and common sub-trees from related problems (CSRP) [4] will be used as benchmarks in this research. It should be noted that in this case both the source problems required by CSRP will be set to the same problem, i.e., only one problem, during the experiments. For each source problem, the building blocks are extracted and then applied to the target problem independently 30 times. The average performance of these 30 runs is calculated. The entire process is then repeated 10 times, and the average performance is reported and presented in Table 1.

Ten regression problems are used to assess the goodness of the proposed method that are five '*polynomial*' (Poly) and five '*trigonometric*' (Trig). The details of theses datasets can be found in [4].

For each problem, there are two different sources, which represent the source problem used for training. In Table 1, these two sources are indicated by 'S0' and 'S1', respectively.

The results show that the proposed method is comparable to CSRP in terms of the mean squared errors, and often outperforms standard GP. A Wilcoxon signed-rank test is performed in order to determine statistical significance. The symbols '+' and '-' are, respectively, used if PST has significantly better and worse performance than the other methods; otherwise, an '=' symbol is used. The three symbols indicate how the proposed method performed compared to STDGP, CSRP (S0) and CSRP (S1), respectively.

In many cases where there was a significant difference between CSRP (S0) and CSRP (S1), the equivalent deviation between PST (S0) and PST (S1) is not present. For example, the testing error on

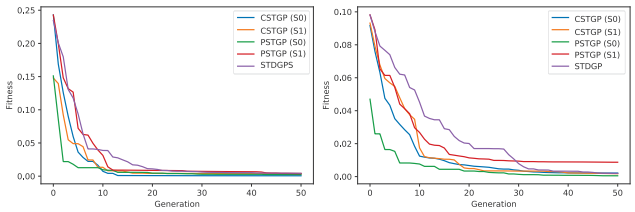

Figure 1: The PST convergence on (a) Poly-1, and (b) Trig-1.

Poly-2 had a relatively large difference between the two runs of the CSRP method, but the equivalent deviation cannot be seen for the PST method. However, the results show that both PST and CSRP are sensitive to the source problem, which can be observed by comparing the results obtained from the two different sources.

The proposed method often had the best starting performance, which shows that the transferred building blocks are immediately useful for the target problem. The training error convergence for Poly-1 and Trig-1 can be seen in Figure 1. Only two of the 10 figures are shown here, however, a similar pattern was noted for the other figures. In both the trigonometric and the polynomial datasets, the proposed method often had a better convergence. The convergence plots for the trigonometric dataset were slightly different. The starting points were much closer, but the convergence of PST still appeared to be better than the other methods in most cases.

4 CONCLUSIONS

This research aimed to develop a new transfer learning technique for GP. The proposed method has demonstrated to be capable of transferring useful information from a source problem to a target problem. It takes information in the form of building blocks, which are functions generated by all the potentially effective sub-trees of a solution, and then filters them. The experimental results on ten symbolic regression problems show that the information transferred, and the way they were used, by the proposed method can be useful compared to standard GP and a state-of-the-art method. Moreover, the proposed method shows better converge compared to the other methods.

In the future, the usage of several individuals for building block retrieval will be investigated.

REFERENCES

[1] Isidro M. Alvarez, Will N. Browne, and Mengjie Zhang. 2016. Human-inspired Scaling in Learning Classifier Systems: Case Study on the n-bit Multiplexer Problem Set. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*. ACM, 429–436.

[2] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. 2007. Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning*. ACM, 193–200.

[3] Thi Thu Huong Dinh, Thi Huong Chu, and Quang Uy Nguyen. 2015. Transfer learning in Genetic Programming. In *Proceedings of 2015 IEEE Congress on Evolutionary Computation*. IEEE, 1145–1151.

[4] Damien O'Neill, Harith Al-Sahaf, Bing Xue, and Mengjie Zhang. 2017. Common subtrees in related problems: A novel transfer learning approach for genetic programming. In *Proceedings of 2017 IEEE Congress on Evolutionary Computation*. IEEE, 1287–1294.

[5] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359.