Differential Evolution for Instance based Transfer Learning in Genetic Programming for Symbolic Regression

Qi Chen, Bing Xue, and Mengjie Zhang
Victoria University of Wellington
Wellington, New Zealand
{Qi.Chen,Bing.Xue,MengjieZhang}@ecs.vuw.ac.nz

ABSTRACT

Transfer learning attracts increasing attention in many fields in recent years. However, studies on transfer learning for symbolic regression are still rare. This work proposes a new instance weighting framework for genetic programming (GP) based symbolic regression for transfer learning. The key idea is to use differential evolution to search for optimal weights during the evolutionary process of GP, which helps GP identify and learn from more useful source domain instances and eliminate the effort of less useful source domain instances. The results show that the proposed method achieves notably better cross-domain generalisation performance in a very stable way than GP without the instance weighting framework and support vector regression.

KEYWORDS

Transfer Learning, Genetic Programming, Symbolic Regression, Differential Evolution

1 INTRODUCTION

Transfer learning is a learning framework in machine learning with the task of improving the learning performance in the target domain by utilising the knowledge gained from different but related source domain(s). The rationale behind transfer learning is that human beings can utilise the acquired knowledge to solve new but similar problems effectively. Let $P_s(Y_s, X_s)$ and $P_t(Y_t, X_t)$ denote the true underlying distribution of the source and target domains, where Y_s and Y_t are the outputs in the source and target domain, respectively, and X_s and X_t are the corresponding input variables. The general idea of transfer learning is to utilise $P_s(Y_s, X_s)$ to better approximate $P_t(Y_t, X_t)$. In recent years, studies on transfer learning have been conducted in many fields, e.g. reinforcement learning and classification [2].

Genetic Programming (GP) based symbolic regression (GPSR) [1] learns the relationship between the independent variables and dependent output(s) and expresses this relationship in mathematical models. GP has the trend of generating overcomplex models that overfit the training data. This is known as the *generalisation* issue in GP. In the transfer learning scenario, it would be more difficult for GP to learn the regression models to generalise well across domains, where the distributions of the data in these domains are different. Moreover, compared with the many studies in some other fields, research on transfer learning in GPSR is still rare.

Goals: This work aims to develop an instance weighting framework for GPSR for transfer learning to improve the *cross-domain generalisation* ability of the evolved regression model. The new instance weighting framework searches for the optimal weights of the source domain instances considering the performance of GP individuals, and it is expected to effectively correcting the difference between the marginal distributions $P_t(X_t)$ and $P_s(X_s)$.

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

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3321941

2 THE PROPOSED METHOD

2.1 The Overall Framework of TLGP

This work proposes a new instance weighting framework for GPSR for transfer learning. GP equipped with the new framework is named *transfer learning GP* (TLGP). The overall structure of TLGP is shown in Figure 1.

Figure 1 shows that, at every generation, GP individuals in TLGP learn from m sets of weighted source domain instances, i.e. the source domain instances weighted by m weight vectors from an optimisation method. A state-of-the-art version of different evolution, Self-adaptive DE (SaDE) [3] is used in this work. Each individual in SaDE, i.e. a weight vector, has n elements, which are the corresponding weights of the n source domain instances and are used to weight the learning error of the GP models/individuals on these instances. The GP individual that has the smallest error under one weight vector will be selected and exposed to the target training instances. In totally, m top GP individuals are selected. The error values of these models on the target domain will be treated as the fitness values of the m weight vectors in SaDE. As the evolutionary process approaching, GP individuals in TLGP are evolved to extract more helpful knowledge from the better weighted source domain instances. Meanwhile, the DE individuals are evolved towards providing a set of better weighted source domain data to train GP individuals that can obtain a smaller training error on the target training data. From generation to generation, the population of weight vectors keeps on evolving along with the evolution of models in GP. In this way, the proposed weighting framework is expected to correct the marginal distribution difference between $P_s(X)$ and $P_t(X)$ more effectively.

2.2 The Evaluation Process in TLGP

TLGP uses two fitness functions for the evaluation and re-evaluation processes. In the evaluation process, the fitness of a model in TLGP is a vector having m dimensions, where m is the number of weight

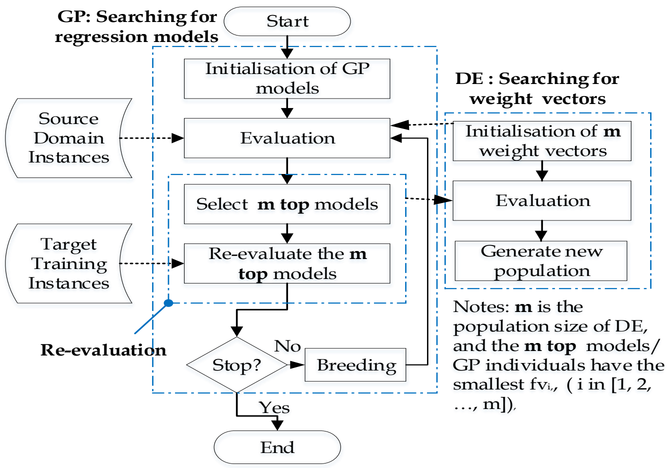

Figure 1: Transfer Learning Framework of TLGP.

vectors returned by SaDE. Each dimension is the weighted error of the individual on the source domain data under one specific weight vector. Weighted mean squared error (WMSE) in Equation (1), is employed as the fitness function for evaluation process.

$$fv_k = WMSE = \sum_{i=1}^{n} w_i \cdot (f_i - y_i)^2, (k \in \{1, 2, \dots, m\}) \qquad (1)$$

where n is the number of the source domain instances, w_i is the weight of the ith instance, f_i is the output of the candidate individual/model and y_i is the target output of the ith instance. The fitness vector of each GP individual consists of values from fv_1 to fv_m.

Given m weight vectors, TLGP selects m models for re-evaluation where the ith models has the smallest fv_i. The errors (MSE) of these m models on the target training domain are measured. This value is also used as the fitness of the the corresponding DE individual.

3 EXPERIMENTS AND RESULTS

The proposed method is compared with three benchmark methods, i.e. support vector regression (*SVR*) and two GP methods,*GP-Tar* and *GP-Comb*, which are GP methods learning from the target training instances only and a combination of the source and the target domain (training) instances, respectively. For an easy comparison, the relative square error ($RSE = \sum_{i=1}^{n} \frac{(f_i - y_i)^2}{(\bar{y} - y_i)^2}$) on the target training data and the target test data are reported and compared. The methods are tested on one synthetic dataset, i.e. the Friedman #1 (Friedman) and two real-world datasets Kin and Student. The values of coefficients in Friedman are drawn from normal distribution with different ranges for the source and the target domains. The Kin dataset is taken from the Delve dataset repositories where "nm" data and "nh" data is used as the source and the target data, respectively. For the student dataset, the Mathematical performance and the Portuguese language data is used as the source and the target data, respectively. 50 independent runs have been conducted for each GP method. The Wilcoxon test, with a significance level of 0.05 and Z-test, are used to for the comparison between GP and SVR.

Results: Table 1 shows that TLGP has worse training performance than GP-Tar on all the target training data, while it has better training performance than GP-Comb and SVR on Student and Friedman.

Table 1: The Training and Test Performance, Program Size and Computational Time

	Method	Training RSE Median(SD)	Test RSE Median(SD)	Pro-Size #Node Median(SD)	Time Seconds Median(SD)
Kin	SVR	0.435	0.747	N/A	N/A
	GP-Tar	0.62(0.09)	0.81(0.05)	108.47(53)	16.98(8.66)
	GP-Comb	0.75(0.08)	0.79(0.06)	103.73(50)	190.19(8.44)
	TLGP	2.44(0.16)	0.71(0.07)	81.07(10)	710.65(417.94)
Student	SVR	0.11	0.44	N/A	N/A
	GP-Tar	0.03(0.02)	0.47(0.14)	144.73(79)	1.68(1.06)
	GP-Comb	0.38(0.01)	0.57(0.01)	135.47(53)	23.56(11.28)
	TLGP	0.05(4.21E-3)	0.2(0.04)	99.07(33)	71.43(53.31)
Friedman	SVR	176.71	1.053	N/A	N/A
	GP-Tar	0.14(0.03)	0.65(0.06)	185.27(85)	2.54(1.09)
	GP-Comb	0.96(3.24E-3)	1.01(4.95E-3)	173.33(107)	45.61(27.12
	TLGP	1.95(0.11)	0.29(0.04)	99.27(41)	124.13(95.05)

All the differences between the training performance are significant. The results on the target training data are not unexpected. In TLGP, the target training instances are involved in the training process to select models and evaluate the weights of the source domain instances. Compared with GP-Tar where these instances are directly involved in the training regression models, TLGP could lead to less effective training fitting.

Considering the generalisation performance on the target test sets, which is a more important criterion to measure the transfer learning performance, the proposed method TLGP is definitely the winner. It has much smaller test errors than the other three method on all the test sets. TLGP has a smaller RSE value than GP-Tar, which indicates that TLGP is able to transfer useful knowledge to improve the generalisation performance of GP. But it is not the case for GP-Comb and SVR. On Friedman where the distribution on the source and target domain is more different, GP-Comb and SVR have much worse test performance than GP-Tar, but TLGP can still outperform GP-Tar, which confirms the effective transfer learning ability of the new instance weighting framework.

It is clear that TLGP generally evolves the simplest regression models, which are much smaller than the models evolved by the other two GP methods in all the cases. The simpler models in TLGP are due to the implicit instance sampling performed by DE to search for the optimal weights. In this way, the evolutionary process generally learns from dynamic while simpler training sets. Considering the computational cost, TLGP is the most expensive method. The additional effort on searching for the weight vector and the more complex evaluations and selection processes all lead to an increase in the computational cost. However, the computational time of TLGP is generally only 2-3 times more than that of GP-Comb and it is still affordable since a GP run takes only several minutes. Clearly, since TLGP can significantly improve the generalisation performance and find/evolve much simpler models, such an increase in training time is a small price to pay.

REFERENCES

[1] John R Koza. 1992. *Genetic Programming II, Automatic Discovery of Reusable Subprograms.* MIT Press, Cambridge, MA.

[2] Haijun Lei, Tao Han, Feng Zhou, Zhen Yu, Jing Qin, Ahmed Elazab, and Baiying Lei. 2018. A deeply supervised residual network for HEp-2 cell classification via cross-modal transfer learning. *Pattern Recognition* 79 (2018), 290–302.

[3] A Kai Qin, Vicky Ling Huang, and Ponnuthurai N Suganthan. 2009. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE transactions on Evolutionary Computation* 13, 2 (2009), 398–417.