Adversarial Bandit Gene Expression Programming for Symbolic Regression

Congwen Xu
Beijing Key Laboratory of Petroleum Data Mining, China
University of Petroleum, Beijing, China
xucongwen-cup@foxmail.com

Qiang Lu*
Beijing Key Laboratory of Petroleum Data Mining, China
University of Petroleum, Beijing, China
luqiang@cup.edu.cn

Jake Luo
Department of Health Sciences and Administration,
University of Wisconsin Milwaukee, United States
jakeluo@uwm.edu

Zhiguang Wang
Beijing Key Laboratory of Petroleum Data Mining, China
University of Petroleum, Beijing, China
cwangzg@cup.edu.cn

ABSTRACT

Gene expression programming (GEP) is a commonly used approach in symbolic regression (SR). However, GEP often falls into a premature convergence and may only reach a local optimum. To solve the premature convergence problem, we propose a novel algorithm based on an adversarial bandit technique, named AB-GEP. AB-GEP segments the mathematical space into many subspaces. It leverages a new selection method, AvgExp3, to enhance the population jump between segmented subspaces while maintaining the population diversity. AvgExp3 dynamically estimates a subspace by rewards generated from AB-GEP without any assumption about the distribution of subspace rewards, making AB-GEP choose the appropriate subspace that contains the correct results. The experimental evaluation shows that the proposed AB-GEP method can maintain the population diversity and obtain better results than canonical GEPs.

CCS CONCEPTS

• **Computing methodologies** → **Genetic programming**; *Machine learning approaches*;

KEYWORDS

adversarial bandit, symbolic regression, gene expression programming

ACM Reference Format:
Congwen Xu, Qiang Lu*, Jake Luo, and Zhiguang Wang. 2021. Adversarial Bandit Gene Expression Programming for Symbolic Regression. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459499

1 INTRODUCTION

For a given dataset $\{X, Y\}$, the goal of symbolic regression (SR) is to find a symbolic function $f(X) = Y'$ that can minimize the distance between Y' and Y in the mathematical expression space. Although many genetic programming (GP) algorithms have shown to be efficient in SR, they tend to converge prematurely and fall into a local optimum owing to the decline in population diversity. For maintaining a healthy population diversity, [3] adopts a static space partition method (**SPJ-GEP**) to let population jump between segmented subspaces. However, SPJ-GEP implementing the UCB and ϵ-greedy strategy to make the population jump between subspaces will become invalid over time. To avoid the performance degradation, we propose a new adversarial bandit based gene expression programming (**AB-GEP**) algorithm. As shown in Figure 1, AB-GEP first partitions the mathematical space into k subspaces. Then, it leverages a new subspace selection method (called **AvgExp3**) which modifies the adversarial bandit Exp3 [2] to choose one subspace. AvgExp3 evaluates the subspace according to the rewards dynamically generated by exploiting subspaces, so it can get the evaluation result more accurately. Next, AB-GEP uses the crossover operator to make the population jump between subspaces. After the population jumps into a subspace, the population recombines with the historically best individual in the subspace to exploit this subspace quickly.

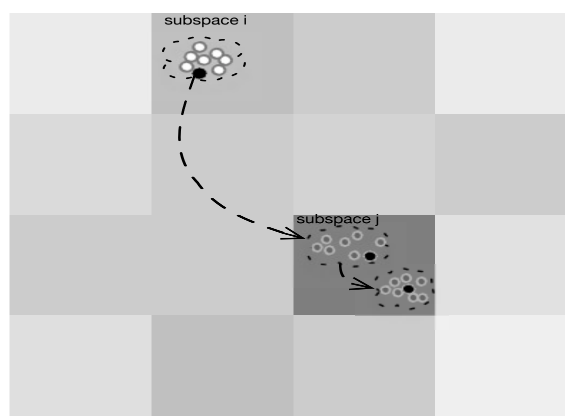


Figure 1: AB-GEP framework

2 ADVERSARIAL BANDIT GEP

The space-partition tree [3] divides the mathematical expression space Ω into k subspaces $\{\omega_1, \omega_2, \ldots \omega_k\}$. If each subspace ω_i is regarded as an arm in a k-armed bandit, the partitioned subspaces

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3459499

$\{\omega_1, \omega_2, \dots \omega_k\}$ are the k-armed bandit environment (adversary). Then SR is similar to the k-armed adversarial bandit problem, i.e., balancing between subspace exploration and subspace exploitation to discover the best mathematical expression that is fitted to the dataset.

2.1 Choosing Subspace

AB-GEP obtains the probability (P_{t,ω_i}) of choosing each subspace ω_i at round (generation) t according to the probability distribution $\mathcal{P}_t = \{P_{t,\omega_1}, ..., P_{t,\omega_k}\}$. P_{t,ω_i} is calculated by Equation 1.

$$P_{t,\omega_i} = \frac{e^{\alpha \hat{S}'_{t-1,\omega_i}}}{\sum_{j=1}^{k} e^{\alpha \hat{S}'_{t-1,\omega_i}}}. \tag{1}$$

where α is the learning rate and $\hat{S}'_{t-1,\omega_i}$ is the average estimated reward of subspace ω_i by the end of round $t-1$. $\hat{S}'_{t-1,\omega_i}$ is calculated by Equation 2.

$$\hat{S}'_{t,\omega_i} = \frac{(t-1)\hat{S}'_{t-1,\omega_i} + 1 - \frac{\mathbb{I}\{\omega_x=\omega_i\}\left(1-\frac{1}{1+f^*_{\omega_x}}\right)}{P_{t,\omega_i}}}{t} \tag{2}$$

where $f^*_{\omega_x}$ is the fitness of the best individual in subspace ω_x.

2.2 Crossover

The crossover goal is to allow the population jump from the original subspace ω_i to a selected subspace ω_j. Meanwhile, the crossover exploits ω_j. The following two steps, jump and exploitation, are the same as the steps in [3]. For example, given an individual '$\{+-\} - \times xxxxxx$' in ω_{+-}, it jumps to ω_{++} after the head of its code is replaced with '$\{++\}$' by crossover. Then, it recombines with the best individual in ω_{++} to exploit ω_{++}. The jump of the population between subspaces avoids falling into a local optimum and maintains the population diversity.

3 EXPERIMENTS

For evaluating the proposed algorithm **AB-GEP**, we import the other three baseline GEP methods: **GEP**, **SL-GEP**, and **SPJ-GEP**. To obtain the performance metrics of the four algorithms, each of the algorithms runs ten times on twelve SR benchmark problems [1]. Moreover, the results are shown in Table 1. Observing the comparative results, we conclude that AB-GEP outperforms GEP, SPJ-GEP, and SL-GEP owing to the following two reasons. One reason is that, AB-GEP can find more correct results on Nguyen benchmark. The other reason is that AB-GEP can obtain smaller RMSEs than the other three algorithms on most benchmarks. In addition, Figure 2 shows the subspace selection method AvgExp3 in AB-GEP is more reasonable than that in SPJ-GEP, which lets AB-GEP obtain more accurate results than SPJ-GEP.

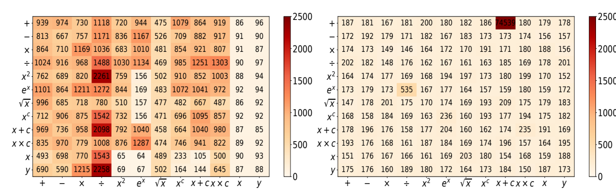
4 CONCLUSION

In this paper, we propose a novel algorithm AB-GEP to solve SR. AB-GEP has a novel subspace selection method AvgExp3 based on the adversarial bandit method. The subspace selection method can find and choose a high-value subspace that contains better results with high probability. Using Avg-Exp3 as the subspace selection method, AB-GEP can make the population jump between subspaces while maintaining the population diversity to avoid falling into a local optimum. The experimental results show that the performance of SPJ-GEP surpasses that of the other three GEPs.

Table 1: performance metrics

Function	GEP		SPJ-GEP		SL-GEP		AB-GEP	
	CR	RMSE	CR	RMSE	CR	RMSE	CR	RMSE
Koza-2	80%	.0059	**100%**	.0079	**100%**	**.0043**	**100%**	.0084
Nguyen-2	50%	.0142	70%	.0059	80%	.0071	**100%**	**.0024**
Nguyen-4	10%	.0256	20%	.0236	20%	.0238	**40%**	**.0206**
Nguyen-5	70%	.0098	**100%**	**.0065**	**100%**	.0073	**100%**	.0072
Nguyen-6	40%	.0123	**100%**	.0024	**100%**	.0025	**100%**	**.0014**
Nguyen-7	70%	.0091	60%	.0097	40%	.0111	**100%**	**.0082**
Nguyen-10	50%	.0137	80%	.0057	50%	.0102	**100%**	**.0018**
Vladis-1	0%	.0526	0%	.0531	0%	.0606	0%	**.0427**
Vladis-2	0%	.0913	0%	**.0637**	0%	.1132	0%	.0759
Vladis-3	0%	.4801	0%	**.4369**	0%	.6739	0%	.5753
Vladis-7	0%	1.3756	0%	1.3540	0%	1.4081	0%	**1.2765**
Vladis-8	0%	.3369	0%	**.2676**	0%	.6237	0%	.3487
best	0	0	3	4	3	1	**7**	**7**

CR represents the proportion of finding the correct result whose fitness computed by RMSE is less than 0.01.



(a) AB-GEP on Vladis-1 (b) SPJ-GEP on Vladis-1

Figure 2: comparison of AB-GEP and SPJ-GEP on selecting subspaces. Each small square represents a subspace. The darker the color, the more times the subspace is selected

ACKNOWLEDGMENTS

This work is supported by China National Key Research Project (No. 2019YFC0312003), National Natural Science Foundation of China (No. 61402532).

REFERENCES

[1] James McDermott, David R. White, Sean Luke andLuca Manzoni, et al. 2012. Genetic programming needs better benchmarks. ACM Press, 791. https://doi.org/10.1145/2330163.2330273

[2] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit algorithms*. Cambridge University Press.

[3] Qiang Lu, Shuo Zhou, Fan Tao, Jake Luo, and Zhiguang Wang. 2021. Enhancing gene expression programming based on space partition and jump for symbolic regression. *Information Sciences* 547 (2021), 553–567.