Side Information-Assisted Symbolic Regression for Data Storage

Xiangwu Zuo
CSE Dept.
Texas A&M Univ.
dkflame@tamu.edu

Anxiao (Andrew) Jiang
CSE Dept.
Texas A&M Univ.
ajiang@cse.tamu.edu

Netanel Raviv
CSE Dept.
Washington Univ. in St. Louis
netanel.raviv@wustl.edu

Paul H. Siegel
ECE Dept.
Univ. of California, San Diego
psiegel@ucsd.edu

Abstract—**There are various ways to use machine learning to improve data storage techniques. In this paper, we study symbolic regression, a machine-learning method for recovering the symbolic form of a function from its samples. We present a new symbolic regression scheme that utilizes side information for higher accuracy and speed in function recovery. The scheme enhances latest results on symbolic regression that were based on recurrent neural networks and genetic programming. The scheme is tested on a new benchmark of functions for data storage.**

I. Introduction

There are various approaches of using machine learning to improve data storage techniques. This work studies symbolic regression, a machine-learning method for recovering the symbolic form of a function from its samples. Specifically, let $\mathbf{x} = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$ be n variables. Let $y = f(\mathbf{x}) \in \mathbb{R}$ be a function of $\mathbf{x}$. Let $S = \{(X_i, y_i) \mid i = 1, 2, \cdots, m\}$ be m samples of the function f, where each point $X_i \in \mathbb{R}^n$ is a sample of $\mathbf{x}$, and $y_i = f(X_i)$ is the corresponding value of y. Given such a dataset of samples S, the goal of *symbolic regression* is to recover the symbolic form of the function f (such as $y = -x_1 \log_2 x_1 - (1 - x_1)\log_2(1 - x_1)$ or $y = \sqrt{(x_1 - x_2)^2 + (x_3 - x_4)^2}$). The recovered function should fit the samples well, and also be as simple as possible.

Symbolic regression can have numerous applications to data storage. One application is modeling physical storage devices, such as non-volatile flash memories, where it can provide functional insights into evolving cell threshold voltage distributions, spatial inter-cell interference effects, and page-level bit error counts as the memory ages over read/write cycles or loses charge with infrequent use. Symbolic regression can bridge experimental data with theoretical models. Another application is aiding the analysis of storage schemes, such as modeling the performance of error-correcting codes, wear-leveling schemes, etc. Here it can bridge simulation data with theoretical models and provide insights for further analysis.

Symbolic regression is a challenging problem in AI. The number of functions in the search space grows exponentially as the lengths of the functions increase. Existing symbolic-regression methods mainly use evolutionary algorithms or deep learning [3]. In particular, a recent method [1] (which we shall call DSR-GP) improves the Deep Symbolic Regression (DSR) approach [2] and achieves state-of-the-art performance by combining deep neural networks with genetic programming. In this work, we introduce a new symbolic regression approach that further improves DSR-GP by combining it with *side information*. Broadly speaking, side information can refer to any information that correlates with the ground-truth function f, although in this paper, we shall focus on a narrower type: functions that resemble all or a part of f. We show that the new scheme, which we shall call DSR-GP-SI, can notably improve the performance of symbolic regression.

A longer version of this paper appeared in *Proc. IEEE Information Theory Workshop (ITW)*, Mumbai, India, November 2022. [4]

II. Symbolic Regression using Side Information

To represent symbolic functions, we need a set of operators S_{op}, a set of variables S_{var}, and a set of values for coefficients S_{coeff}. For example, if $S_{op} = \{+, -, \times, \div, \sin, \cos, \log, \exp\}$, $S_{var} = \{x_1, x_2\}$ and $S_{coeff} = \mathbb{R}$, then $\sin(x_1 - x_2) + 1.5\exp(x_1)$ is a valid symbolic function. If we wish to represent all coefficients using a placeholder "const", we can let $S_{coeff} = \{\text{const}\}$. The three sets S_{op}, S_{var} and S_{coeff} together form the *token library*. A symbolic function can be represented by its *Polish notation*, which is a depth-first traversal of the symbols in the computation tree (where leaves are variables or constants, and internal nodes are operators) of the symbolic function. For example, for $f(x_1, x_2) = \frac{-x_1}{x_1+x_2}\log x_2 - \sin(2x_1)\cos x_2$, its *Polish notation* is "$- \times \div \times \text{const}\, x_1 + x_1\, x_2\, \log\, x_2 \times \sin \times \text{const}\, x_1\, \cos\, x_2$."

Let $y = f(\mathbf{x})$ be the ground-truth function we look for. Let G_f be its computation tree. This work considers a function f_{SI} to be *side information (SI)* for f if the computation tree of f_{SI}, $G_{f_{SI}}$, can be obtained from a subtree of G_f, G_{sub}, where some subtrees in G_{sub} can be replaced by the "const" tokens. (For discussions on how side information can be obtained in practice, and how to extend the above SI to more general cases, please see the full paper [4].) The *similarity* between f and f_{SI}, $\gamma(f, f_{SI})$, can be measured as follows. Let $L(f)$ and $L(f_{SI})$ be the number of symbols in f and f_{SI}, respectively. Then $\gamma(f, f_{SI}) \triangleq L(f_{SI})/L(f) \in [0, 1]$.

We present a new symbolic regression (SR) scheme DSR-GP-SI, which extends DSR [2] and DSR-GP [1] by incorporating side information. It searches for a function $\hat{f}$ that not only fits the samples in S well, but also is close to the provided side-information function f_{SI}. To define the "fitness" of $\hat{f}$, $\mathcal{F}_{S,f_{SI}}(\hat{f})$, let us first define several notations: (1) let $\mu_y = \frac{1}{|S|}\sum_{i=1}^{|S|} y_i$ and $\sigma_y = \sqrt{\frac{1}{|S|}\sum_{i=1}^{|S|}(y_i - \mu_y)^2}$, then the "normalized root-mean-square error" $\text{NRMSE}(\hat{f}) \triangleq \frac{1}{\sigma_y}\sqrt{\frac{1}{|S|}\sum_{i=1}^{|S|}(\hat{f}(X_i) - y_i)^2}$; (2) let $d_{Lev}(\hat{f}, f_{SI})$ be the Levenshtein distance between the Polish notations of $\hat{f}$ and f_{SI}, then the "pre-order representation distance" $\text{PORD}(\hat{f}, f_{SI}) \triangleq \frac{d_{Lev}(\hat{f}, f_{SI})}{L(\hat{f})+L(f_{SI})}$; (3) let $\delta > 0$ be a constant parameter, then the "normalized distance to side-information function" $\text{NDSIF}(\hat{f}, f_{SI}) \triangleq \frac{1}{|S|}\sum_{i=1}^{|S|}\frac{|\hat{f}(X_i)-f_{SI}(X_i)|}{\max\{|f_{SI}(X_i)|,\delta\}}$. Then $\mathcal{F}_{S,f_{SI}}(\hat{f}) \triangleq \frac{w_1}{1+\text{NRMSE}(\hat{f})} + \frac{w_2}{1+\text{PORD}(\hat{f},f_{SI})} + \frac{w_3}{1+\text{NDSIF}(\hat{f},f_{SI})}$, where w_1, w_2, w_3 are constant parameters. The greater $\mathcal{F}_{S,f_{SI}}(\hat{f})$ is, the better $\hat{f}$ is considered to be.

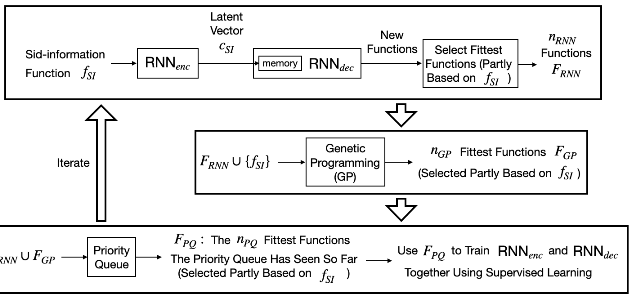
Sid-information Function f_{SI}
RNN$_{enc}$
Latent Vector c_{SI}
memory
RNN$_{dec}$
New Functions
Select Fittest Functions (Partly Based on f_{SI})
n_{RNN} Functions F_{RNN}
Iterate
$F_{RNN} \cup \{f_{SI}\}$
Genetic Programming (GP)
n_{GP} Fittest Functions F_{GP} (Selected Partly Based on f_{SI})
$F_{RNN} \cup F_{GP}$
Priority Queue
F_{PQ} : The n_{PQ} Fittest Functions The Priority Queue Has Seen So Far (Selected Partly Based on f_{SI})
Use F_{PQ} to Train RNN$_{enc}$ and RNN$_{dec}$ Together Using Supervised Learning

Fig. 1. The Symbolic Regression Scheme DSR-GP-SI using side information.

The DSR-GP-SI scheme is illustrated in Fig. 1. It searches for good functions in iterations, and it ends either when a found function matches the samples in S sufficiently well, or when it exceeds a time budget. In each iteration, the scheme works as follows: (1) Let f_{SI} be the given side-information function. We use a recurrent neural network (RNN) RNN$_{enc}$ to transform it to a latent vector c_{SI}, and use c_{SI} to initialize the memory of another RNN RNN$_{dec}$. We then use RNN$_{dec}$ to generate N_{RNN} new functions in an auto-regressive way (by generating the symbols in the Polish notation one by one), and select from them n_{RNN} functions of the highest "fitness". Let F_{RNN} denote those n_{RNN} functions. (2) Let $\mathcal{T}_{GP}^{0} = F_{RNN} \cup \{f_{SI}\}$ be the initial population for the genetic programming (GP) component. Use α generations (of GP operations including mutation, crossover and selection) to generate a new population $\mathcal{T}_{GP}^{\alpha}$, and select from it n_{GP} functions of the highest "fitness". Let F_{GP} denote those n_{GP} functions. (3) Feed the functions in $F_{RNN} \cup F_{GP}$ to a persistent maximum reward priority queue (MRPQ), which saves the n_{PQ} functions of the highest fitness it has seen. Then, the n_{PQ} functions in MRPQ (denoted by F_{PQ}) are used as sample functions to train the two RNNs RNN$_{enc}$ and RNN$_{dec}$ together using supervised learning, so that they can learn to generate new functions that are similarly good.

We evaluate the performance of DSR-GP-SI, and compare it to DSR [2] and DSR-GP [1], which the DSR-GP-SI scheme evolved from. We compare their performance on two existing benchmarks: Nguyen [3] and Livermore [1]. To better evaluate the potential of symbolic regression for data storage, we also present a new benchmark of 20 functions (called the *DS benchmark*), shown in [4]. The performance of the three schemes for the DS benchmark is shown in Fig. 2. (Here the side-information functions are generated randomly. For details, see [4].) It can be seen that having side information can help improve the recovery rate (the fraction of times of recovering functions correctly) substantially; and generally, the more side information (measured by similarity between the ground-truth function and the SI function), the better. Furthermore, even when "similarity" is small, there often still exist (short) SI functions that can improve the recovery rate significantly. That can be seen from those gray circles near the top of the figure (whose recovery rates are close or equal to 1). Similar performance improvement by side information can be observed for the Nguyen and Livermore benchmarks [4].

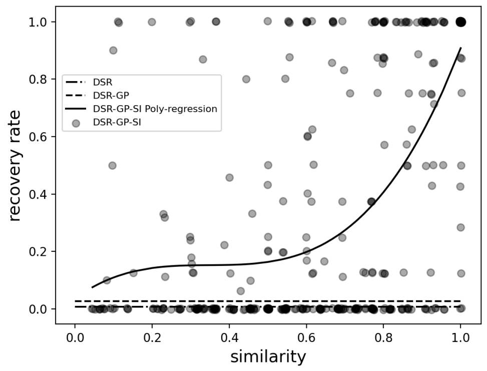

Fig. 2. Performance of DSR-GP-SI (solid curve and gray circles), DSR-GP (dashed line) and DSR (dash-dotted line) for the DS benchmark. Here the x-axis is the "similarity" $\gamma(f, f_{SI})$ between the ground-truth function and side-information function, and the y-axis is the "recovery rate", i.e., the fraction of times a ground-truth function is correctly recovered. (Note that "similarity" is relevant to DSR-GP-SI, but not to DSR or DSR-GP.) Each gray circle corresponds to a particular benchmark function and a particular SI function, where numerous experiments for DSR-GP-SI were performed and their recovery rate was shown. (Note that different sets of experiments may produce gray circles that overlap each other. The more overlapping, the darker the gray color becomes.) The solid curve shows the trend of the gray circles: for the average recovery rates corresponding to different "similarity" values (averaged over all functions in the DS benchmark and their experimented SI functions for each given "similarity"), the solid curve is their polynomial regression (a generalization of linear regression) with polynomial degree 3. The average recovery rates of DSR and DSR-GP (which use no side information) are 0.007 and 0.027, respectively, which are shown as two horizontal lines for easy comparison.

References

[1] T. N. Mundhenk, M. Landajuela, R. Glatt, C. P. Santiago, D. M. Faissol, and B. K. Petersen, "Symbolic Regression via Neural-Guided Genetic Programming Population Seeding," arXiv:2111.00053, 2021.

[2] B. K. Petersen, M. Landajuela, T. N. Mundhenk, C. P. Santiago, S. K. Kim, and J. T. Kim, "Deep Symbolic Regression: Recovering Mathematical Expressions from Data via Risk-Seeking Policy Gradients," in *Proc. International Conference on Learning Representations (ICLR)*, 2021.

[3] N. Q. Uy, N. X. Hoai, M. O'Neill, R. I. McKay, and E. Galvan-Lopez, "Semantically-based Crossover in Genetic Programming: Application to Real-valued Symbolic Regression," in *Genetic Programming and Evolvable Machines*, vol. 12, 91–119, 2011.

[4] X. Zuo, A. Jiang, N. Raviv and P. H. Siegel, "Symbolic Regression for Data Storage with Side Information," in *Proc. IEEE ITW*, 2022.