Investigating the Use of Linear Programming to Solve Implicit Symbolic Regression Problems

Quang Nhat Huynh
University of New South Wales
Canberra, ACT, Australia
quang.huynh@student.adfa.edu.au

Hemant Kumar Singh
University of New South Wales
Canberra, ACT, Australia
h.singh@adfa.edu.au

Tapabrata Ray
University of New South Wales
Canberra, Australia, ACT
t.ray@adfa.edu.au

ABSTRACT

Approaches that attempt to combine feature selection and symbolic feature evolution processes have gained popularity in the recent years to solve symbolic regression problems. However, much of the existing research is applicable/has been applied to solve *explicit equations* ($y = f(\mathbf{x})$) only, which cannot be directly applied to discover *implicit equations* ($f(\mathbf{x}) = 0$). In this paper, we investigate a potential approach that uses Linear Programming to construct meaningful implicit equations out of a set of evolving features.

CCS CONCEPTS

• **Computing methodologies → Hybrid symbolic-numeric methods**;

KEYWORDS

Symbolic Regression, Linear Programming, Implicit Equation

ACM Reference Format:
Quang Nhat Huynh, Hemant Kumar Singh, and Tapabrata Ray. 2019. Investigating the Use of Linear Programming to Solve Implicit Symbolic Regression Problems. In *Proceedings of the Genetic and Evolutionary Computation Conference 2019 (GECCO '19 Companion)*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3321889

1 INTRODUCTION

Implicit functions are those with the form of $f(\mathbf{x}) = 0$ and usually represent complex surfaces [5], partial differential equations [3] and natural laws [4]. Though explicit equations ($y = f(\mathbf{x})$) are well studied in the literature, there exist very limited attempts to solve implicit equations symbolically. For implicit equations, Genetic Programming (GP) with Mean Square Error (MSE) fitness can yield trivial expressions which are zeros for all or most of the dataset, such as $sin^2(x) + cos^2(x) - 1$ or $1/(1000 + x^2)$ [5]. The work in [5] attempts to solve this issue using a new fitness function involving partial derivatives in lieu of MSE. However, the application of the approach is challenging if there are more than three variables in the dataset and/or the data points are not in spatial or time order.

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

ACM ISBN 978-1-4503-6748-6/19/07...$15.00
https://doi.org/10.1145/3319619.3321889

More recent methods for explicit functions combine feature selection methods such as Least Absolute Shrinkage and Selection Operator (LASSO) [1] or Mixed Integer Linear Programming (MILP) [2] with feature evolution processes in the form of GP. Unfortunately, these approaches are not suited to discover implicit equations as they always return zeros as the weights of the selected features. In this paper, we propose a novel approach of using Linear Programming (LP) to search for implicit equations efficiently.

2 PROPOSED APPROACH

2.1 Linear Programming

Our goal is to form an implicit expression which has the minimum Sum of Absolute Error (SAE), out of a subset of the feature set. When the output is zero, the SAE is calculated as in Eq. 1.

$$\underset{w}{\text{Minimize}} \quad \sum_i \quad \Big|\sum_j w_j s_{ij}\Big|, \tag{1}$$

where s_{ij}, $i = 1, 2, ..., n$, and $j = 1, 2, ..., m$, denote the output values of m features on n given fitness cases and w_j indicates the weight of feature j. At the core of our approach, this minimization problem is transformed to an equivalent linear formulation shown in Eq. 2 with an extra variable z_i, which indicates the difference between predicted output and the target for data point i.

$$\begin{aligned} &\underset{z}{\text{Minimize}} && \sum_i z_i, \qquad z_i \geq 0 \\ &\text{Subject to:} && -z_i \leq \sum_j w_j s_{ij} \leq z_i \end{aligned} \tag{2}$$

2.1.1 *Avoiding all zero weights.*

Eq. 2 is not sufficient to prevent the trivial solution where all the weights w are zeros. Therefore, an extra constraint is required to prevent this case. Let s_{rj} be the output of feature j based on a random fitness case where the output is known to be non-zero in advance. Eq. 3, where c is a preset constant, will enforce at least one of the weights to be non-zero. Only one constraint for the positive c is necessary since LP can simply negate the values of the weights if $\sum_j w_j s_{rj}$ turns out to be negative.

$$\sum_j w_j s_{rj} \geq c, \qquad c > 0 \tag{3}$$

2.1.2 *Linear programming with $L1$ regularization.*

Minimizing only the SAE suffers from many side effects, such as precision error, too many non-zero weights, etc. A common approach is adding a $L1$ norm penalty term to the objective as in Eq. 4 to achieve a sparse selection effect. λ is the weight of the penalty term. The new minimization problem can be transformed to its equivalent LP form in Eq. 5, where u_j denotes the absolute value of w_j.

$$\underset{w}{\text{Minimize}} \quad \sum_i \Big|\sum_j w_j s_{ij}\Big| \quad + \quad \lambda \|\mathbf{w}\|_1 \tag{4}$$

$$\begin{aligned} \underset{z}{\text{Minimize}} \quad & \sum_i z_i + \lambda \sum_j u_j, \qquad z_i \geq 0, \quad u_j \geq 0 \\ \text{Subject to:} \quad & -z_i \leq \sum_j w_j s_{ij} \leq z_i \\ & -u_j \leq w_j \leq u_j \end{aligned} \tag{5}$$

2.1.3 *Mixed Integer Linear Programming with L0 regularization.*

Regularization with $L1$ norm usually results in the selection of too many features. In order to determine only a few features to construct an expression, MILP can be used in conjunction with a $L0$ norm penalty. Eq. 6 shows a SAE minimization problem with $L0$ regularization. It can be transformed to an MILP formulation as in Eq. 7, where t_j is a binary variable indicating if feature j is selected, w_{ub} and w_{lb} are the upper and lower bounds of w respectively.

$$\underset{w}{\text{Minimize}} \quad \sum_i \Big|\sum_j w_j s_{ij}\Big| \quad + \quad \lambda \|\mathbf{w}\|_0 \tag{6}$$

One drawback of using MILP is its heavy computation load and our recommendation is to limit the size of the input feature set.

$$\begin{aligned} \underset{z}{\text{Minimize}} \quad & \sum_i z_i + \lambda \sum_j t_j, \qquad z_i \geq 0, \quad t_j = \{0,1\} \\ \text{Subject to:} \quad & -z_i \leq \sum_j w_j s_{ij} \leq z_i \\ & t_j w_{lb} \leq w_j \leq t_j w_{ub} \end{aligned} \tag{7}$$

2.1.4 *Avoiding combination of previous grouped features.*

MILP with $L0$ regularization can be executed multiple times on one feature set to construct many expressions from different subsets. The constraint in Eq. 8 is required to avoid MILP choosing the same subset as previous executions. S_k is the subset of features selected in execution k and S contains all previous subsets.

$$\sum_{j \in S_k} t_j \leq |S_k| - 1 \quad \forall S_k \in S \tag{8}$$

The above constraint allows MILP to keep exploring a feature set when the previously constructed expressions did not yield a good validation error.

2.2 Main Algorithm

Algo. 1, inspired by EFS [1], is embedded with the formulations presented above. Note that even though LP with $L1$ regularization is mentioned above, it has not been integrated into our algorithm.

3 EXPERIMENTS

Table 1 lists the experimental set up for this study, while Table 2 shows the simulated benchmarks and their corresponding discovered forms by our proposed algorithm. One point to note is the scaling effect of c. Naturally the value of c should be set to as close to 0 as possible. However, because the right hand side is always 0, c can be of any magnitude, which in turn will affect the magnitude of w, w_{ub}, w_{lb} and even λ.

Algorithm 1 Proposed Algorithm

```
Require: Parameters listed in Table 1;
 1: $FS \leftarrow All\ original\ variables$; // FS: feature set
 2: $gen \leftarrow 0$;
 3: while $gen < G$ do
 4:     $S \leftarrow \emptyset$;
 5:     for $k \leftarrow 1$ to $K$ do
 6:         $S_k \leftarrow Generate\_Expression\_MILP(FS, S)$;
            // $S_k$: selected feature set in iteration $k$
 7:         $S \leftarrow \{S, S_k\}$;
 8:     end for;
 9:     $FS \leftarrow \{Unique\ features\ in\ S, all\ original\ variables\}$;
10:     $FS \leftarrow Generate\_Extra\_Features(FS, N, M)$;
11: end while;
```

Table 1: Parameters used for the numerical experiments

Parameter	Value
c, λ, w_{ub}, w_{lb}	0.5, 1e-4, 1, -1
G, K	10, 10
N: max size of feature set	40
M: max size of individual feature	5
Operators	+, -, *, /, sin, cos, exp, log

Table 2: Benchmark problems and discovered forms

Problem	Definition	Discovered Form
Circle	$x^2 + y^2 - 16$	$-0.0625078x^2 - 0.0625023y^2 + 1$
Sphere	$x^2 + y^2 + z^2 - 2$	$0.0499958x^2 + 0.0500005y^2 + 0.0499979z^2 - 0.0999904$
Pendulum	$\omega^2 - 19.6cos(\theta)$	$-0.805999cos(\theta) + 0.0411235\omega^2$
Pendulum	$\alpha + 9.8sin(\theta)$	$0.0458249\alpha + 0.449085sin(\theta)$

4 CONCLUSIONS

In this study, we propose a novel way to apply LP to solve implicit equations symbolically. In principle, this method can be also applied to uncover explicit equations and has the potential of discovering multiple relationships in one run. For future work, we would like to improve the feature evolution process and carry out in-depth investigation of the effect of using LP with $L1$ norm regularization to determine the input feature set for MILP; especially targeted towards scaling the performance for high-dimensional problems.

REFERENCES

[1] Ignacio Arnaldo, Una-May O'Reilly, and Kalyan Veeramachaneni. 2015. Building Predictive Models via Feature Synthesis. In *ACM Genetic and Evolutionary Computation Conference (GECCO)*. ACM, New York, NY, USA, 983–990.

[2] Q. N. Huynh, S. Chand, H. K. Singh, and T. Ray. 2018. Genetic Programming With Mixed-Integer Linear Programming-Based Library Search. *IEEE Transactions on Evolutionary Computation* 22, 5 (Oct 2018), 733–747.

[3] Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. 2017. Data-driven discovery of partial differential equations. *Science Advances* 3, 4 (2017).

[4] Michael Schmidt and Hod Lipson. 2009. Distilling Free-Form Natural Laws from Experimental Data. *Science* 324, 5923 (2009), 81–85.

[5] Michael Schmidt and Hod Lipson. 2010. *Symbolic Regression of Implicit Equations.* Springer US, Boston, MA, 73–85.