RC25620 (WAT1608-056) August 15, 2016
Mathematics

IBM Research Report

Globally Optimal MINLP Formulation for Symbolic Regression

Lior Horesh
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598 USA

Leo Liberti
LIX, École Polytechnique
F-91128 Palaiseau
France

Haim Avron
Tel Aviv University
55 Haim Levanon
Tel-Aviv, 6997801
Israel

Research Division
Almaden – Austin – Beijing – Brazil – Cambridge – Dublin – Haifa – India – Kenya – Melbourne – T.J. Watson – Tokyo – Zurich

Globally Optimal MINLP Formulation for Symbolic Regression

Lior Horesh[1], Leo Liberti[2], Haim Avron[3]

[1] *IBM "T.J. Watson" Research Center, Yorktown Heights, 10598 NY, USA*
Email:rhoresh@us.ibm.com

[2] *LIX, École Polytechnique, F-91128 Palaiseau, France*
Email:liberti@lix.polytechnique.fr

[3] *Tel Aviv university, 55 Haim Levanon, Tel-Aviv, 6997801, Israel*
Email:haimav@post.tau.ac.il

August 9, 2016

Abstract

We describe a mathematical program which finds an expression tree of least description complexity that fits a set of observations as well as required.

1 Introduction

We are concerned with the problem of finding a closed-form expression of a function $f(x)$, where $x = (x_1, \ldots, x_D)$, for which we only know a set of approximate observations $P = \{(\chi^h, \phi^h) \mid h \in H\}$ such that $f(\chi^h) \approx \phi^h$ for each $h \in H$.

We use the approximation relation symbol $\approx$ in the following sense: if μ is a metric on $\mathbb{R}^{|H|}$, then there is some constant $\varepsilon > 0$ such that $\mu(\mathbf{f}, \boldsymbol{\phi}) \leq \varepsilon$, where $\mathbf{f} = (f(\chi^h) \mid h \in H)$ and $\boldsymbol{\phi} = (\phi^h \mid h \in H)$. For the rest of the paper we will assume that $\mu(\mathbf{f}, \boldsymbol{\phi}) = \|\mathbf{f} - \boldsymbol{\phi}\|_2^2$, but this need not necessarily be the case in general.

An *expression digraph* is a directed graph representation of a mathematical expression g (we assume g to be a sentence of a formal language $\mathcal{L}$). The expression digraph D_g of g can be obtained as the parsing tree T_g which is derived from parsing g with the formal grammar of the language [1][1]. Although T_g is a directed tree in the graph theoretical sense of the word, if some term t appears more than once in g, then one can contract the nodes of T_g representing t, and end up with a Directed Acyclic Graph (DAG) [1]. The representation T_g of g is useful because it makes the recursive evaluation of g very efficient. In this paper, we are going to focus on expression trees rather than actual digraphs, i.e. we do not contract same term nodes.

In very broad terms, our problem is as follows. We are given:

- the sets H and P and the constant ε described above;
- the alphabet of the formal language $\mathcal{L}$,

and we look for an expression tree $T = (N, A)$, with nodes labelled by symbols in the alphabet, such that:

1. $|N| + |A|$ is minimum;

[1] Get a textbook citation

2. the evaluation function $\mathsf{eval}_T(x)$ of the expression represented by G is such that:

$$\mu(\mathbf{e}(\boldsymbol{\chi}), \boldsymbol{\phi}) \leq \varepsilon, \tag{1}$$

where $\mathbf{e}(\boldsymbol{\chi}) = (\mathsf{eval}_G(\chi^h) \mid h \in H)$.

Every expression tree corresponds to a unique mathematical expression g of $\mathcal{L}$. Since $\mathcal{L}$ is a formal language, we assume it has no ambiguity, which means that every mathematical expression in $\mathcal{L}$ corresponds to a unique expression tree[2] We define $|N| + |A|$ to be the *description complexity* $\Delta(g)$ of g.

We call this the Model Finding Problem (MFP).

2 The mathematical program

The aforementioned problem can be formulated as an optimization problem subject to constraints. More specifically, we minimize the description complexity of T subject to Eq. (1), which is a constraint on the fidelity of the evaluation process of T with respect to the given observations.

Mathematical Programming (MP) is a formal language for describing optimization problems. The advantage of employing MP as a modelling tool for optimization problems is that very generic solvers are readily available to be deployed on many different classes of MPs. Specifically, our MP formulation turns out to belong to the very general class of Mixed-Integer Nonlinear Programs (MINLP). These describe optimization problems where some of the variables may be constrained to have integral values, and where some of the variables may appear nonlinearly in the problem.

An optimization problem has known input data and encodes the problem solution by means of variables. Accordingly, MP formulations involve different *entities*: *parameters*, i.e. the known input data, *decision variables*, which encode the problem solution, an *objective function*, and possibly some *constraints*. In MINLPs, objective functions have the form $\min_x g_0(x)$ and constraints have the form $g_i(x) \leq 0$ (for all $i \leq m$), where $g = (g_0, \ldots, g_m)$ is a sequence of mathematical expressions in $\mathcal{L}$. There are trivial reformulations which allow to express different optimization directions and different constraint sense by means of the above symbols [2].

2.1 Parameters

1. The dimension d of the Euclidean space embedding the observations.
2. The set H of observation indices.
3. The observation points $(\chi^h \mid h \in H)$ at which f is evaluated.
4. The observation values $(f^h \mid h \in H)$ corresponding to the evaluation of f at the observation points.
5. The set $\mathcal{A}$ of alphabet symbol indices.
6. The alphabet $\{\oplus_a \mid ; a \in \mathcal{A}\}$ of the formal language $\mathcal{L}$, which consists of mathematical operators of given arity K_a for each $a \in \mathcal{A}$.
7. Since many nodes can be labelled by the same operator symbol $\oplus_a$, we introduce a maximum quantity C_a of occurrences of operator $\oplus_a$ in the expression.
8. The set $V = \{(a, i) \mid a \in \mathcal{A} \wedge i \leq C_a\}$ of all potential nodes in the problem (N will turn out to be a subset of V).

[2]This bijection between expressions and trees ceases to hold if we consider expression digraphs, as there may be multiple ways to contract same term nodes.

For the sake of clarity, we assume the alphabet $\mathcal{A}$ to consist of very common arithmetic operators of arity two and one, such as + (binary sum), − (binary difference), × (binary product), ÷ (binary division), ˆ(binary power), $\cdot^2$ (unary square), $\sqrt{\cdot}$ (unary square root), log (unary logarithm), exp (unary exponential), sin (unary sine) and cos (unary cosine). In general, obviously, this need not be the case.

We also include two special operators `var` and `coef`, which stand for a variable and a coefficient. Although these operators obviously have arity zero, for technical reasons (which will be explained below) we define them as having arity 1.

2.2 Decision variables

Our model finder MINLP must take both topological decisions on the structure of the expression tree T, as well as numerical decisions about the evaluation of T at the given observation points. Accordingly, we introduce two main classes of decision variables.

2.2.1 Numerical variables

Every operator $\oplus_a$ for $a \in \mathcal{A}$, aside from the *leaf* operators in $L = \{\texttt{var}, \texttt{coef}\}$, has a given number K_a of operands. We associate two sets of variables, v and w, with nodes $s \in N$: specifically, v_{sh} holds the value of the evaluation at node $s \in N$ for observation point $h \in H$, and w_{skh} holds the corresponding value at the k-th subnode of s.

2.1 Example
For example, in the expression $x_4 + 10$ evaluated at $\chi_1^3 = 5$, we would have $v_{(+,1),3} = 15$, $w_{(+,1),1,3} = \chi_1^3 = 5$, and $w_{(+,1),2,3} = 10$ (see Fig. 1). Note that the index of x_4 does not appear in these variables — the matching of nodes to coordinate indices is made through the structural variable γ (see Sect. 2.2.2).

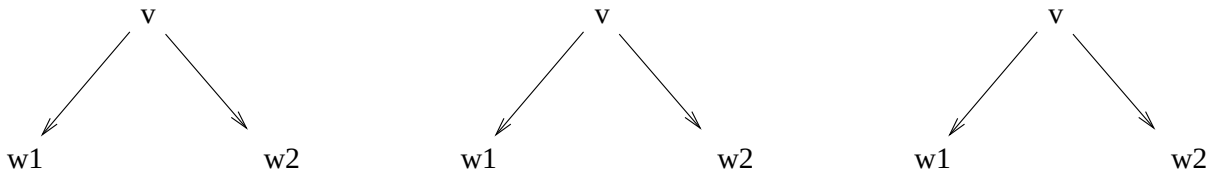

Figure 1: The expression appearing in Example 2.1.

The reason why we need two sets of variables is that both v and w refer to the same operand s, but v indicates the parent node and w the child nodes; the unique relationship between parent and children warrants the distinction between v and w. The structural variables (see Sect. 2.2.2) match different occurrences of operators as operands, as shown in Example 2.2.

2.2 Example
The expression tree $3(x_4 + 5)$ consists of two operators: ×, which has operands 3 and +, and + itself, which has operands x_4 and 5. The two subtrees and the whole tree are shown in Fig. 2.

Example 2.2 shows that the + operator is both a parent and a child. This constraint will be enforced by setting $v_{th} = w_{skh}$ for each $h \in H$, whenever $t \in V$ is the k-th operand of $s \in V$.

Additionally, we also introduce a variable δ with the purpose to minimize the approximation error with respect to the observations: essentially, the error can never be more than ε, but if possible it will be driven to be lower.

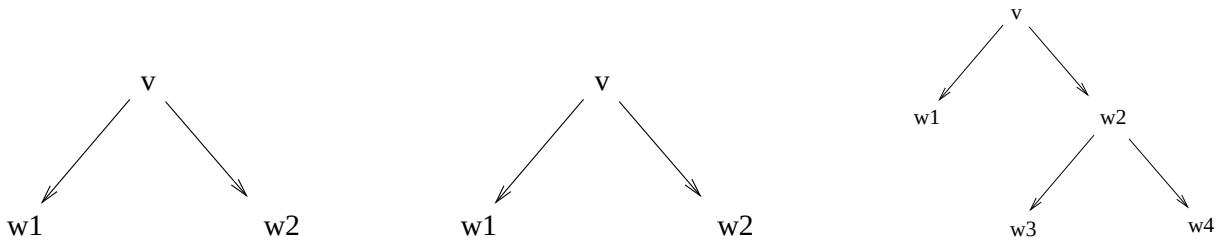

Figure 2: The expression appearing in Example 2.2.

2.2.2 Structural variables

All of the following structural variables are constrained to be binary (i.e. their values are in $\{0, 1\}$).

1. For each $s \in V$, $\alpha_s = 1$ if and only if $s \in N$, i.e. s is part of T.
2. For each $s \in V$, $\rho_s = 1$ if and only if s is the root node of T.
3. For each $s \in V$ and $d \leq D$, $\gamma_{sd} = 1$ if and only if: (a) s is a variable node (i.e. $s = (\mathtt{var}, i)$ for some $i \leq C_{\mathtt{var}}$) and (b) s is the variable representing the d-th coordinate x_d.
4. For each $s = (a, i), t = (b, j) \in V$ and $k \leq K_b$, $\beta_{stk} = 1$ if and only if s is the k-th operand of t.
5. For each $s = (a, i), t = (b, j) \in V$, $\sigma_{st} = 1$ if and only if s is an operand of t.

Note that σ are a projection of the β variables: essentially, $\sigma_{st} = 1$ if and only if there is a k such that $\beta_{stk} = 1$ (constraints to this effect are introduced below). However, we use β, σ for two different purposes: β control the matching of v and w variables (i.e. $v_{sh} = w_{tkh}$ if and only if $\beta_{stk} = 1$), whereas σ are used to ensure that the selected structure is a tree.

Additionally, we introduce variables to determine the rank of nodes in the evaluation tree: for each $s \in V$, r_s is the rank of node s. Although the rank is an integer value, the integrality of r_s will be enforced by linear constraints involving the binary structural variables σ. This means that r_s can be simply defined to be a continuous variable.

2.2.3 Boundedness

Most MINLP solvers are based on spatial Branch-and-Bound (sBB), which is a worst-case exponential-time exhaustive (albeit implicit) search for an approximate global optimum. Like all Branch-and-Bound (BB) algorithms, it computes upper and lower bounds for the optimal objective function value at a node by means of "subsolvers"; if these bounds differ by more than a given tolerance, the feasible space at the node is partitioned in some way, and the search is recursed on the subnodes. Because of the nature of the subsolvers, it is better, in general, if all variables are bounded. Although this will call for imposing arbitrary bounds, and will prevent the solver from detecting unboundedness, we are going to postulate that v, w both belong to a "large enough" hypercube $-M \leq v, w \leq M$. The rank variables r range between 0 and $|V|$, and δ ranges between 0 and ε. Naturally, all binary variables range by definition between 0 and 1.

2.3 Objective function

In Sect. 1, we defined the description complexity of the mathematical expression g, which we mean to determine by means of its expression tree T, as $\Delta(g) = |N| + |A|$. We can write Δ in function of the

decision variables as:

$$\Delta(g) = \sum_{s \in V} \alpha_s + \sum_{s,t \in V} \sigma_{ts}. \tag{2}$$

On the other hand, if we limit ourselves to the simple but practically useful arithmetic alphabet $\mathcal{A}$ defined in Sect. 2.1, we observe that all operators have arity at most two, which, since T is a tree, means that $\sum_{s,t \in V} \sigma_{ts} \leq 2 \sum_{s \in V} \alpha_s$. So it suffices to define the objective function $\min \sum_{s \in V} \alpha_s$.

We also want to minimize the error δ which measures how well g approximates the unknown function f. This results in a second objective function $\min \delta$, making this into a bi-objective MINLP. Since we also bound this error above by ε, however, we focus on defining the global optimum with respect to the description complexity, and merely focus on driving δ down as a second priority. Provided $\varepsilon < 1$, which we can always enforce by suitably modifying the error metric μ, and since $\Delta(g)$ can only change by at least one unit at a time (the value difference in any binary decision variable α_s), we can trivially scalarize the two objectives to obtain:

$$\min \sum_{s \in V} \alpha_s + \delta. \tag{3}$$

2.4 Constraints

The MP formulation for the MFP lists several constraint classes: approximation error, numerical relationships between parent and child nodes, structural constraints to determine a tree, and mixed constraints which relate numerical and structural variables.

2.4.1 Approximation error

As mentioned in Sect. 1, the approximation error of g with respect to f is given by $\mu(\mathbf{e}(\boldsymbol{\chi}), \boldsymbol{\phi})$, which we implement as $q\|\mathbf{e}(\boldsymbol{\chi}) - \boldsymbol{\phi}\|_2^2$, where q is an optional scaling constant. The vectors in the argument are both in $\mathbb{R}^{|H|}$, and the h-th component of $\mathbf{e}(\boldsymbol{\chi})$ is $\mathsf{eval}_G(\chi^h)$. Our formulation is set up so that the value of the mathematical expression evaluated at the h-th observation point is stored in the decision variable v_{sh} where s is the root node of the expression tree. Since a node is root if and only if $r_s = 1$, we have:

$$\sum_{h \in H} \left(\sum_{s \in V} \rho_s v_{sh} - \phi^h \right)^2 \leq \delta. \tag{4}$$

2.4.2 Numerical values of expression terms

These constraints assign correct values to the nodes of the tree, so that v_{sh} is the value of the term rooted at s when evaluated with the h-th observation point, for each $s \in V$ and $h \in H$. The general schema, for the i-th occurrence of the operator $\oplus_a$, is:

$$\forall h \in H \quad v_{(a,i),h} = \bigoplus_{k \leq K_a} w_{(a,i),k,h}. \tag{5}$$

We list different constraints for every non-leaf operator in $\mathcal{A}$:

$$\forall i \leq C_{+}, h \in H \quad v_{(+,i),h} = w_{(+,i),1,h} + w_{(+,i),2,h} \tag{6}$$
$$\forall i \leq C_{-}, h \in H \quad v_{(-,i),h} = w_{(-,i),1,h} - w_{(-,i),2,h} \tag{7}$$
$$\forall i \leq C_{\times}, h \in H \quad v_{(\times,i),h} = w_{(\times,i),1,h}\, w_{(\times,i),2,h} \tag{8}$$
$$\forall i \leq C_{\div}, h \in H \quad v_{(\div,i),h} = w_{(\div,i),1,h} / w_{(\div,i),2,h} \tag{9}$$
$$\forall i \leq C_{\hat{\cdot}}, h \in H \quad v_{(\hat{\cdot},i),h} = w_{(\hat{\cdot},i),1,h}^{w_{(\hat{\cdot},i),2,h}} \tag{10}$$
$$\forall i \leq C_{\cdot^2}, h \in H \quad v_{(\cdot^2,i),h} = w_{(\cdot^2,i),1,h}^2 \tag{11}$$
$$\forall i \leq C_{\sqrt{\cdot}}, h \in H \quad v_{(\sqrt{\cdot},i),h} = \sqrt{w_{(\sqrt{\cdot},i),1,h}} \tag{12}$$
$$\forall i \leq C_{\log}, h \in H \quad v_{(\log,i),h} = \log(w_{(\log,i),1,h}) \tag{13}$$
$$\forall i \leq C_{\exp}, h \in H \quad v_{(\exp,i),h} = e^{w_{(\exp,i),1,h}} \tag{14}$$
$$\forall i \leq C_{\sin}, h \in H \quad v_{(\sin,i),h} = \sin(w_{(\sin,i),1,h}) \tag{15}$$
$$\forall i \leq C_{\cos}, h \in H \quad v_{(\cos,i),h} = \cos(w_{(\cos,i),1,h}). \tag{16}$$

We remark that enforcing all of the nonlinear constraints above concurrently makes it very difficult for a MINLP solver to find the any solution. In practice, if one knows that certain operators will never occur in g, it is best to remove the operator from $\mathcal{A}$ (and hence the corresponding constraints) altogether.

Next, if an operator is a coefficient, its value must be the same over all the observation points.

$$\forall i \leq C_{\texttt{coef}}\ ,\ h < \ell \in H \quad v_{(\texttt{coef},i),h} = v_{(\texttt{coef},i),\ell}. \tag{17}$$

2.4.3 Structural operator relationships

The tree T has exactly one root node:

$$\sum_{s \in V} \rho_s = 1. \tag{18}$$

Root nodes are used:

$$\forall s \in V \quad \rho_s \leq \alpha_s. \tag{19}$$

If the root is a leaf, then $g = x_d$ for some $d \leq D$ or g is a constant. If this is the case, then no arc exists between nodes:

$$\forall q, s, t = (b, j) \in V, k \leq K_b \quad \beta_{stk} \leq 1 - \rho_q. \tag{20}$$

Note that in practice one would never look for such a simple expression to fit a set of observation data, so these constraints can usually be removed.

The w variables are indexed as w_{skh}, and denote the numerical value (evaluated at the h-th observation point) at the k-th child node of the parent s. Accordingly, for leaf operators which have no children, the corresponding β variables should be zero:

$$\forall s \in V, t = (b, j) \in L, k \leq K_b \quad \beta_{stk} = 0. \tag{21}$$

The parent-child relationship is irreflexive (no loops in the DAG):

$$\forall s = (a, i) \in V, k \leq K_a \quad \beta_{ssk} = 0. \tag{22}$$

We do not match the same operand node to two different child nodes:

$$\forall s, t = (b, j) \in V \quad \sum_{k \leq K_b} \beta_{stk} \leq 1. \tag{23}$$

Note that Eq. (23) is only valid if T is an expression tree. If we consider contractions of similar nodes, as in expression DAGs, then these constraints should be removed.

Only match used nodes:

$$\forall s, t = (b, j) \in V, k \leq K_b \quad \beta_{stk} \leq \alpha_s \tag{24}$$
$$\forall s, t = (b, j) \in V, k \leq K_b \quad \beta_{stk} \leq \alpha_t. \tag{25}$$

A used, non-root node must be matched to some parent (non-leaf) node other than itself:

$$\forall s \in V \quad \sum_{\substack{t=(b,j)\in V \\ t\notin L\cup\{s\}}} \sum_{k\leq K_b} \beta_{stk} \geq \alpha_s - \rho_s. \tag{26}$$

Only match `var` operator nodes to data points:

$$\forall s \in V \setminus \{\texttt{var}\}, d \leq D \quad \gamma_{sd} = 0. \tag{27}$$

Each `var` operator node must be matched to at least one data component if used.

$$\forall i \leq C_{\texttt{var}} \quad \sum_{d\leq D} \gamma_{(\texttt{var},i),d} \geq \alpha_{(\texttt{var},i)}. \tag{28}$$

Each observation point must be matched to at least one `var` node.

$$\forall d \leq D \quad \sum_{i\leq C_{\texttt{var}}} \gamma_{(\texttt{var},i),d} \geq 1. \tag{29}$$

Root nodes cannot be child nodes:

$$\forall s, t \in V \quad \sigma_{st} \leq 1 - \alpha_s. \tag{30}$$

Leaf nodes cannot be in a parent-child relationship with each other:

$$\forall s, t \in inL \quad \sigma_{st} = 0. \tag{31}$$

The root node is a parent:

$$\forall s \in V \quad \sum_{\substack{t\in V \\ t\neq s}} \sigma_{ts} \geq \rho_s. \tag{32}$$

Note that these constraints yield a contradiction with Eq. (20). In practice, Eq. (20) will be removed, whereas Eq. (32) will be active. These two constraint sets can be made compatible by defining additional binary variables to enforce that not both should be active at the same time.

Every used `var` node is a child of a non-leaf node:

$$\forall i \leq C_{\texttt{var}} \quad \sum_{t\in V\setminus L} \sigma_{(\texttt{var},i),t} = \alpha_{(\texttt{var},i)}. \tag{33}$$

Projection of β variables on σ variables:

$$\forall s \in V \setminus L, t = (b, j) \in V, k \leq K_b \quad \beta_{stk} \leq \sigma_{st} \tag{34}$$
$$\forall s \in V \setminus L, t = (b, j) \in V \quad \sum_{k\leq K_b} \beta_{stk} \geq \sigma_{st}. \tag{35}$$

The rank of the root node is zero:

$$\forall s \in V \quad r_s \leq (1 - \rho_s)|V|. \tag{36}$$

The rank of unused nodes is zero:

$$\forall s \in V \quad r_s \leq \alpha_s|V|. \tag{37}$$

If t is the parent of s, then its rank is the rank of s plus 1:

$$\forall s \in V, t \in V \quad r_s + 1 - (1 - \sigma_{st})(|V| + 1) \leq r_t \leq r_s + 1 + (1 - \sigma_{st})(|V| + 1). \tag{38}$$

2.4.4 Mixed constraints

Numerical values of nodes stored in v and w variables are equal if there is a parent-child relationship in the respective indices.

$$\forall s \in V, t = (b, j) \in V, k \leq K_b, h \in H \quad \beta_{stk}(v_{sh} - w_{tkh}) = 0. \tag{39}$$

Numerical values of `var` nodes stored in v are equal to observation points if the `var` node is assigned to the corresponding coordinate.

$$\forall d \leq D, h \in H, i \leq C_{\texttt{var}} \quad \gamma_{(\texttt{var},i),d}(v_{(\texttt{var},i),h} - \chi^h_d) = 0. \tag{40}$$

2.4.5 Cuts

The following are implied constraints. Although they do not modify the set of feasible or optimal solutions, they may improve the feasible/optimal set of the relaxation used by the sBB algorithm to compute lower bounds to the optimal objective function value at a node.

No node is its own parent (this is implied by Eq. (22) and Eq. (34)-(35)):

$$\forall s \in V \quad \sigma_{ss} = 0. \tag{41}$$

No mathematical sub-expression of the form $x_d - x_d$ can ever appear in g (this is implied by minimality of description complexity, Eq. (3)):

$$\forall i \leq K_- \tag{42}$$

References

[1] A. Costa, P. Hansen, and L. Liberti. Formulation symmetries in circle packing. In R. Mahjoub, editor, *Proceedings of the International Symposium on Combinatorial Optimization*, volume 36 of *Electronic Notes in Discrete Mathematics*, pages 1303–1310, Amsterdam, 2010. Elsevier.

[2] L. Liberti, S. Cafieri, and F. Tarissan. Reformulations in mathematical programming: A computational approach. In A. Abraham, A.-E. Hassanien, P. Siarry, and A. Engelbrecht, editors, *Foundations of Computational Intelligence Vol. 3*, number 203 in Studies in Computational Intelligence, pages 153–234. Springer, Berlin, 2009.