CCSI
Carbon Capture Simulation Initiative

Learning Models of Unspecified Functional Form through Symbolic Regression

Alison Cozad, Nick Sahinidis, Zachary Wilson

CMPD CENTER

Carnegie Mellon University

NETL

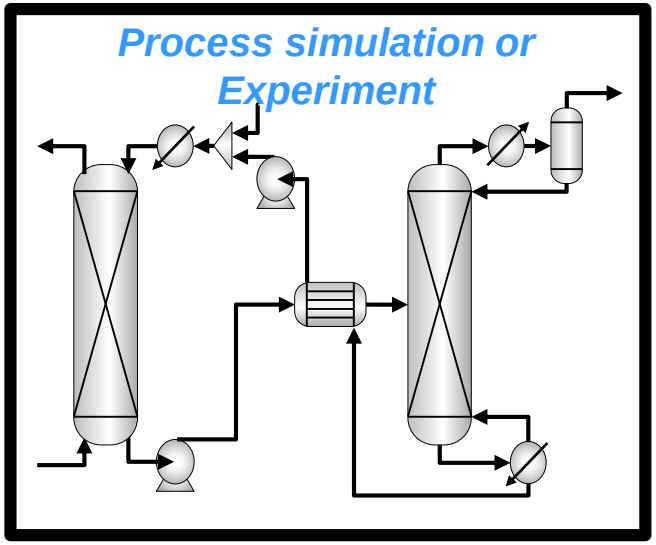
CCSI

LEARNING Problem

Build a model of output variables z as a function of input variables x over a specified interval

$$x \in \mathbb{R}^k$$
$$x^l \leq x \leq x^u$$

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_j \\ \vdots \\ x_k \end{pmatrix}$$



$$\begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_l \\ \vdots \\ z_m \end{pmatrix}$$

$$z \in \mathbb{R}^m$$
$$z = f(x)$$

Independent variables:
Operating conditions, inlet flow properties, unit geometry

Dependent variables:
Efficiency, outlet flow conditions, conversions, heat flow, etc.

PARAMETRIC REGRESSION

- Functional forms specified a priori
  - Multiple linear regression

    Model selection can compare functional forms, linear in coefficients

$$\hat{z}(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 e^{x_1} + \beta_5 e^{x_2} + \ldots$$

$$\hat{z}(x) = 2 + x_2 + 5\, e^{x_1}$$

  - Nonlinear regression

    Function is nonlinear in terms of regression variable

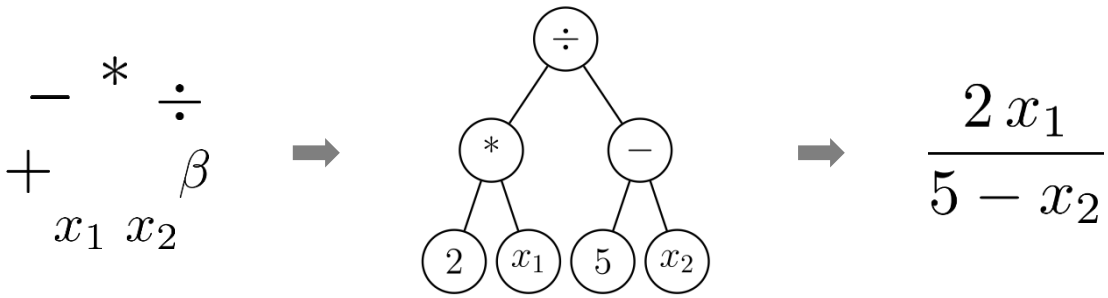
$$\hat{z}(x) = \beta_0 \exp^{\frac{\beta_1}{T}}$$

$$\hat{z}(x) = x_1^{\beta_1} + x_2^{\beta_2} + \beta_0$$

SYMBOLIC REGRESSION

- Flexible nonlinear regression
  - Symbolic regression

    Offers a source of nonlinear forms given only a set of operators addition, subtraction, multiplication, division, etc.



  - General linear models

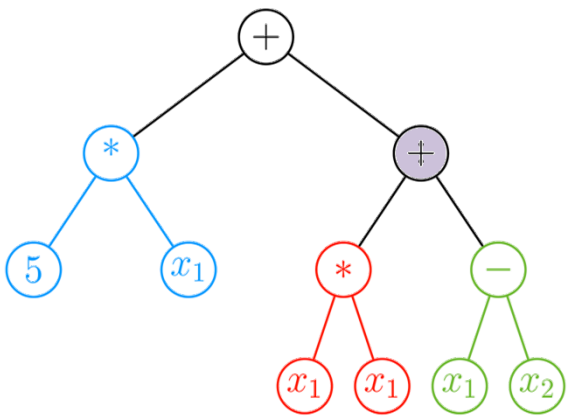
    Functional forms generated can be added to larger linear models

$$\hat{z}(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 \frac{2x_1}{5 - x_2}$$

EXPRESSION TREES

- Representation of nonlinear functional form
  - Can define any function using an appropriate tree

    Recursively defines the order of operations in a function using operands at leaf nodes and operators for all other nodes



$$5\,x_1 + \frac{(x_1)^2}{x_1 - x_2}$$

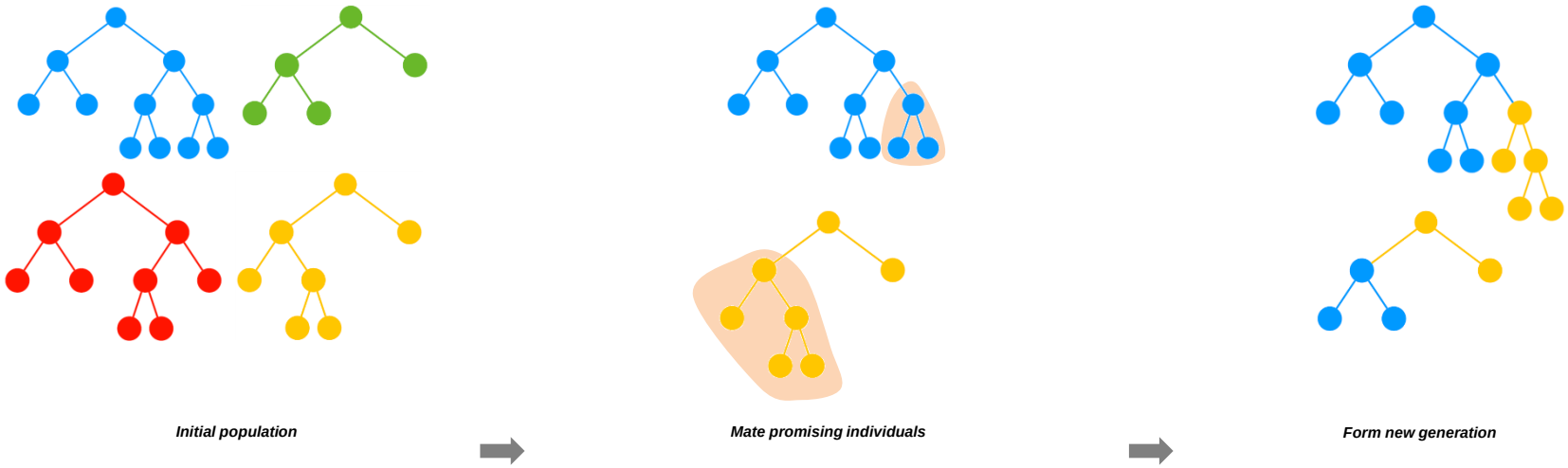
$$5\,x_1 + (x_1)^2 + x_1 - x_2$$

- Small changes drastically alter function

GENETIC PROGRAMMING

- ***Functional form learned without explicit instructions***
  - *Symbolic regression thought of as an application of genetic programming [Koza, 1992]*
  - *Stochastic nature offers no guarantees of an optimal model*

- ***Genetic algorithms are typically used to adjust and improve expression trees***



EXPRESSION TREE NOTATION

- Disjunctions modeled with binary variables $\{\boldsymbol{y_n^o}\}$

$$\{+\} \vee \{-\} \vee \{*\} \vee \{\div\} \vee \{\mathrm{cst}\} \vee \{x_d\} \vee \ldots$$

  - Nodes indexed by n
  - Sets of operators used as basis for logical constraints

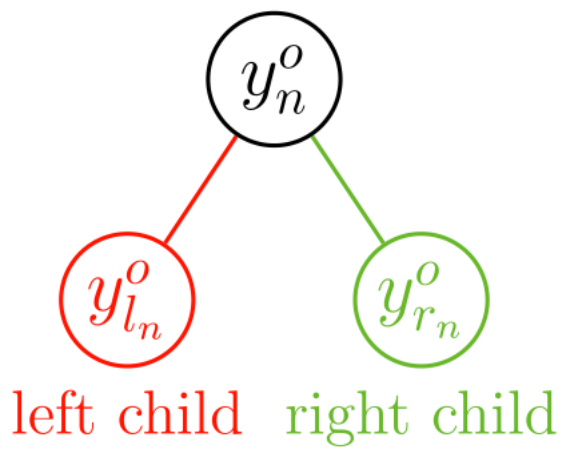
- Value at each node $\{\boldsymbol{v_{i,n}}\}$

  - Associated upper and lower bounds

- Function value at root node

$$f_i^o\left(v_{in}, v_{il_n}, v_{ir_n}, x_{id}\right)$$

  - Dependent on binary variables
  - Models constants and variables



$\mathcal{O}$	All operators and operands
$\mathcal{B}$	Binary operators
$\mathcal{U}$	Unary operators
$\mathcal{L}$	Operands or leaf nodes
$\mathcal{V}$	Variable node

RIGOROUS SYMBOLIC REGRESSION

- Benefits of the rigorous formulation include
  - Certification of the globally optimal functional form and parameter levels
  - Deterministic solutions, no run-to-run variation

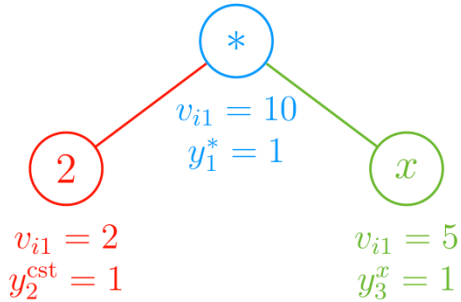
- Rigorous disjunctive mixed-integer nonlinear problem that
  - Minimizes model error
  - By relating node values using disjunctions over each node

$$\{+\} \vee \{-\} \vee \{*\} \vee \{\div\} \vee \{\text{cst}\} \vee \{x_d\} \vee \ldots$$

  - Example function

$$\text{Function: } \hat{z}(x) = 2\,x$$
$$\hat{z} = 10 \text{ at point } i,\ x_i = 5$$

$$\{*\} \text{ is active at Node 1} \Longrightarrow y_1^* = 1$$
$$\{\text{cst}\} \text{ is active at Node 2} \Longrightarrow y_2^{\text{cst}} = 1$$
$$\{x\} \text{ is active at Node 3} \Longrightarrow y_3^x = 1$$



OPTIMIZATION FORMULATION

$$
\begin{aligned}
\min \quad & [\text{error}] \\
\text{s.t.} \quad & \{+\} \vee \{-\} \vee \{*\} \vee \{\div\} \vee \{\text{cst}\} \vee \{x_d\} \vee \ldots \quad \forall \text{ nodes} \\
& [\text{logical constraints}] \\
& v_{in} \in [v_{in}^{\text{lo}}, v_{in}^{\text{up}}] \qquad \forall \text{ nodes } n \text{ and points } i
\end{aligned}
$$

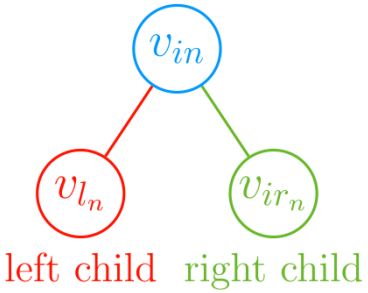
- Disjunctions formulated using big-M constraints
  - Defining operation at each non-terminal nodes, data point, and operator

$$
\begin{aligned}
f_i^o\left(v_{in}, v_{il_n}, v_{ir_n}, x_{id}\right) &\leq \overline{M}_{in}^o\left(1 - y_n^o\right) \\
f_i^o\left(v_{in}, v_{il_n}, v_{ir_n}, x_{id}\right) &\geq \underline{M}_{in}^o\left(1 - y_n^o\right)
\end{aligned}
$$

  - Example: Addition operator

$$
\begin{aligned}
y_n^+ &= 1 \\
v_{il_n} + v_{ir_n} - v_{in} &\leq \overline{M}_{in}^+\left(1 - y_n^+\right) \\
v_{il_n} + v_{ir_n} - v_{in} &\geq \underline{M}_{in}^+\left(1 - y_n^+\right)
\end{aligned}
$$

$$v_{in} = v_{il_n} + v_{ir_n}$$



OPTIMIZATION FORMULATION

$$
\begin{aligned}
\min \quad & [\text{error}] \\
\text{s.t.} \quad & \{+\} \vee \{-\} \vee \{*\} \vee \{\div\} \vee \{\text{cst}\} \vee \{x_d\} \vee \ldots \quad \forall \text{ nodes} \\
& [\text{logical constraints}] \\
& v_{in} \in [v_{in}^{\text{lo}}, v_{in}^{\text{up}}] \qquad \forall \text{ nodes } n \text{ and points } i
\end{aligned}
$$

- Disjunctions formulated using big-M constraints
  - Defining operation at each terminal nodes, data point, and operator

$$
\begin{aligned}
f_i^o\left(v_{in}, x_{id}\right) &\leq \overline{M}_{in}^o \left(1 - y_n^o\right) \\
f_i^o\left(v_{in}, x_{id}\right) &\geq \underline{M}_{in}^o \left(1 - y_n^o\right)
\end{aligned}
$$

  - Set constant term

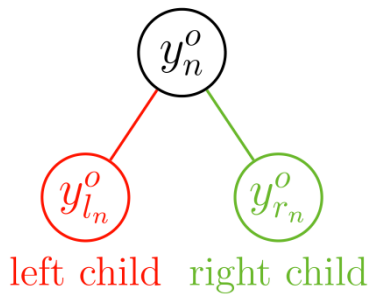
$$
\begin{aligned}
& v_{in} = v_{i'n} \\
& \text{for all data points } i \neq i'
\end{aligned}
\qquad
\begin{aligned}
& y_n^{\text{cst}} = 1 \\
& v_{in} - v_{i'n} \leq \overline{M}_{in}^{\text{cst}} \left(1 - y_n^{\text{cst}}\right) \quad \forall i \neq i' \\
& v_{in} - v_{i'n} \geq \underline{M}_{in}^{\text{cst}} \left(1 - y_n^{\text{cst}}\right) \quad \forall i \neq i'
\end{aligned}
$$

  - Set variable

$$
\begin{aligned}
& v_{in} = x_i \\
& \text{for data point } i
\end{aligned}
\qquad
\begin{aligned}
& y_n^x = 1 \\
& v_{in} - x_i \leq \overline{M}_{in}^x \left(1 - y_n^x\right) \\
& v_{in} - x_i \geq \underline{M}_{in}^x \left(1 - y_n^x\right)
\end{aligned}
$$

  - If no operator or operation is active, the nodes values are set to zero

OPTIMIZATION FORMULATION



$\mathcal{O}$	All operators and operands
$\mathcal{B}$	Binary operators
$\mathcal{U}$	Unary operators

- Logical constraints
  - Ensure only one operator per node

$$\sum_{o \in \mathcal{O}} y_n^o \leq 1 \qquad n = 1, 2, \ldots, N$$

  - If the node is a binary operator both children are active

$$\sum_{o \in \mathcal{B}} y_n^o \leq y_{l_n}^o \qquad n = 1, 2, \ldots, N$$

$$\sum_{o \in \mathcal{B}} y_n^o \leq y_{r_n}^o \qquad n = 1, 2, \ldots, N$$

  - If the node is a unary operator the left child is active

$$\sum_{o \in \mathcal{U}} y_n^o \leq 1 - y_{r_n}^o \qquad n = 1, 2, \ldots, N$$

EXAMPLE – COMPLEXITY CONTROL

Find a model for actual function:

$$z(x) = \frac{2\,x_1}{5 - x_2}$$

given 10 randomly sampled points $x_1, x_2 \in [0, 1]$

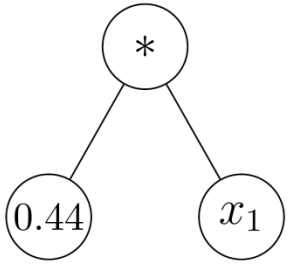
$$\sum_{i=1}^{N} \sum_{o \in O} y_n^o \leq T$$

One node

$\hat{z}(x) = 0.2405$

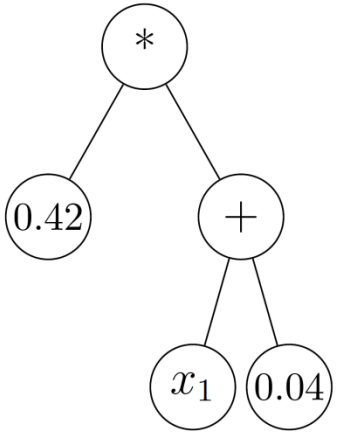
Three node

$0.4434\,x_1$



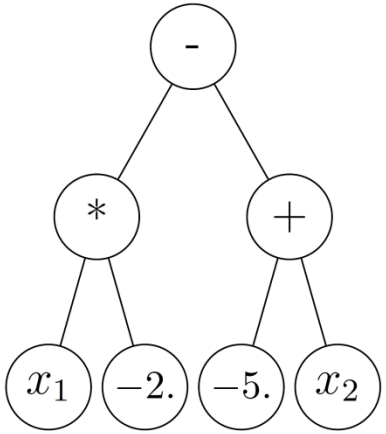
Five node

$0.42\,(\,x_1 + 0.039)$



Seven node

$-2.00\,x_1/\,(-5.00 + x_2)$



RMSE: 0.129 | 0.00360 | 0.00306 | 0.000

ILLUSTRATIVE EXAMPLE

Find a model for actual function:

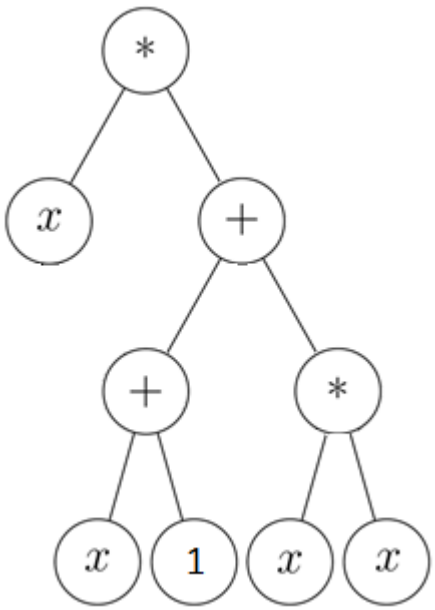
$$z(x) = x^3 + x^2 + x$$

given 20 randomly sampled points $x \in [-1, 1]$

- ***"Difficult synthetic problem" [McDermott 2014]***
  - ***50% success rate for an error tolerance of 0.01 using state-of-the-art implementations [Uy 2011]***
- ***Starts with tree depth of 4***
  - $\boldsymbol{O} \in \{+, -, *, \div, \exp, \log, \boldsymbol{x}, c\}$
  - ***MINLP formulation contains 300 continuous variables, 120 binary variables, and 9,047 constraints***
  - ***Optimal solution found in 1200s***
  - ***7 node function chosen in favor of 11 node function***

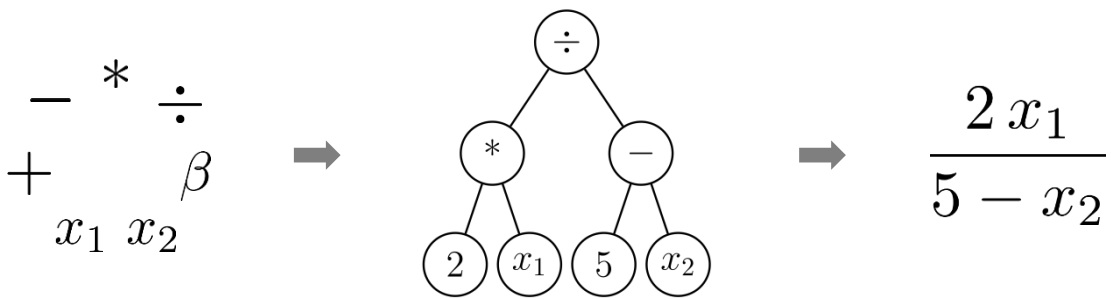
Seven node

$$z(x) = x * (x + 1 + x * x)$$



RMSE: 0.000

CONCLUSIONS



- We propose the first deterministic optimization formulation for the rigorous solution of symbolic regression problems
  - provides a certified optimal solution
  - No run-to-run variation
- We have shown exact function matches for literature problems that routinely show less that 50% success using current state-of-the-art symbolic regression methods

Acknowledgements

- Nick Sahinidis
- Alison Cozad
- David Miller & CCSI

Disclaimer This presentation was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.