Symbolic Regression with Interaction-Transformation

Prof. Fabricio Olivetti de França

Federal University of ABC
Center for Mathematics, Computation and Cognition (CMCC)
Heuristics, Analysis and Learning Laboratory (HAL)

12 de Julho de 2018

Topics

1. Goals
2. Interaction-Transformation
3. Experiments
4. Conclusions

Goals

Find the function

Find a function f that minimizes the approximation error:

$$\begin{aligned} \underset{\hat{f}(\mathbf{x})}{\text{minimize}} \quad & \|\epsilon\|^2 \\ \text{subject to} \quad & \hat{f}(\mathbf{x}) = f(\mathbf{x}) + \epsilon. \end{aligned}$$

Simple is better

- Ideally this function should be as simple as possible.
- Conflict of interests:
  - minimize approximation (use universal approximators)
  - maximize simplicity (walk away from generic approximators)

The good...

The Linear Regression:

$$\hat{f}(\mathbf{x}, \mathbf{w}) = \mathbf{w} \cdot \mathbf{x}.$$

- Very simple (and yet useful) model.
- Clear interpretation
- The variables may be non-linear transformation of the original variables.

. . . the bad. . .

The mean:

$$\hat{f} = \bar{f}(x)$$

- The average can lie!
- It's just for the sake of the pun

. . . the Deep Learning

A deep chaining of non-linear transformations that just works!

- Universal approximation
- Alien mathematical expression

The best?

Despite its success with error minimization, it raises some questions:

- What does the answer mean?
- What if the data is wrong?

Symbolic Regression

Searches for a function form and the correct parameters.

Hopefully a simple function

Symbolic Regression

Disclaimer: I have large experience with evolutionary algorithms, but limited with Symbolic Regression. I have start studying that last year.

Symbolic Regression

This was my first experience with GP:

$$\begin{aligned}
&6.379515826309025e-3+-0.00*id(x_1^{-}4.0*x_2^{3}.0*x_3^{1}.0)\\
&+-0.00*id(x_1^{-}4.0*x_2^{3}.0*x_3^{2}.0)-0.01*id(x_1^{-}4.0*x_2^{3}.0*x_3^{3}.0)\\
&-0.02*id(x_1^{-}4.0*x_2^{3}.0*x_3^{4}.0)+0.01*cos(x_1^{-}3.0*x_2^{-}1.0)+\\
&0.01*cos(x_1^{-}3.0)+0.01*cos(x_1^{-}3.0*x_3^{1}.0)+0.01*cos(x_1^{-}3.0*x_2^{1}.0)\\
&+0.01*cos(x_1^{-}2.0*x_2^{-}2.0)-0.06*log(x_1^{-}2.0*x_2^{-}2.0)\\
&+0.01*cos(x_1^{-}2.0*x_2^{-}1.0)+0.01*cos(x_1^{-}2.0*x_2^{-}1.0*x_3^{1}.0)\\
&+0.01*cos(x_1^{-}2.0)+0.01*cos(x_1^{-}2.0*x_3^{1}.0)\\
&+0.01*cos(x_1^{-}2.0*x_3^{2}.0)+0.01*cos(x_1^{-}2.0*x_2^{1}.0)\\
&+0.01*cos(x_1^{-}2.0*x_2^{1}.0*x_3^{1}.0)+-0.00*id(x_1^{-}2.0*x_2^{2}.0)\\
&-0.00*sin(x_1^{-}2.0*x_2^{2}.0)+0.01*cos(x_1^{-}2.0*x_2^{2}.0)+\dots
\end{aligned}$$

Why?

- Infinite search space
- Redundancy
- Rugged

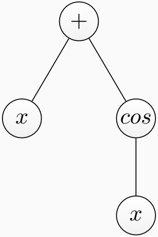
Redundancy

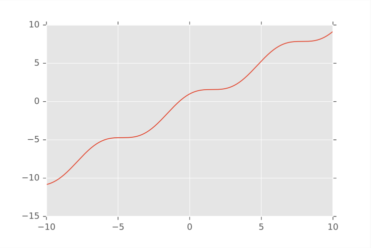
$$f(x) = \frac{x^3}{6} + \frac{x^5}{120} + \frac{x^7}{5040}$$

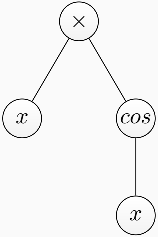
$$f(x) = \frac{16x(\pi - x)}{5\pi^2 - 4x(\pi - x)}$$

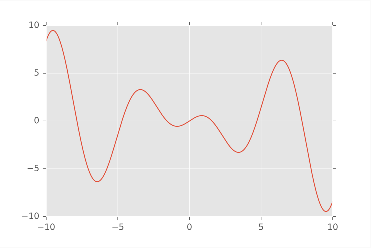
$$\mathbf{f}(\mathbf{x}) = \sin(\mathbf{x}).$$

Rugged space









What I wanted

- A few additive terms (linear regression of transformed variables)
- Each term with as an interaction of a couple of variables
- Maximum of one non-linear function applied to every interaction (no chaining)

Interaction-Transformation

Interaction-Transformation

Constrains the search space to what I want: a **linear combination** of the application of different **transformation functions** on **interactions** of the original variables.

Interaction-Transformation

Essentially, this pattern:

$$\hat{f}(x) = \sum_i w_i \cdot t_i(p_i(x))$$

$$p_i(x) = \prod_{j=1}^{d} x_j^{k_j}$$

$$t_i = \{id, \sin, \cos, \tan, \sqrt{}, \log, \ldots\}$$

Interaction-Transformation

Valid expressions:

- $5.1 \cdot x_1 + 0.2 \cdot x_2$
- $3.5 \sin\left(x_1^2 \cdot x_2\right) + 5 \log\left(x_2^3 / x_1\right)$

Interaction-Transformation

Invalid expressions:

- $\tanh\left(\tanh\left(\tanh\left(w \cdot x\right)\right)\right)$
- $\sin\left(x_1^2 + x_2\right)/x_3$

Interaction-Transformation

We can control the complexity of the expression by limiting the number of additive terms and the number of interactions:

$$\hat{f}(x) = \sum_{i=1}^{k} w_i \cdot t_i(p_i(x))$$

$$p_i(x) = \prod_{j=1}^{d} x_j^{k_j}$$

$$s.t. |\{k_j \mid k_j \neq 0\}| \leq n$$

Interaction-Transformation

Describing as an Algebraic Data Type can help us generalize to other tasks:

```
IT x       = 0 | Weight (Term x) `add` (IT x)

Term x     = Trans (Inter x)
Trans      = a -> a

Inter x:xs = 1 | x s `mul` Inter xs
```

The meaning of `add` and `mul` can lead us to boolean expressions, decision trees, program synthesis.

SymTree

Simple search heuristic:

```
symtree x leaves | stop      = best leaves
                 | otherwise = symtree x leaves'
  where
    leaves' = [expand leaf | leaf <- leaves]

symtree x [linearRegression x]
```

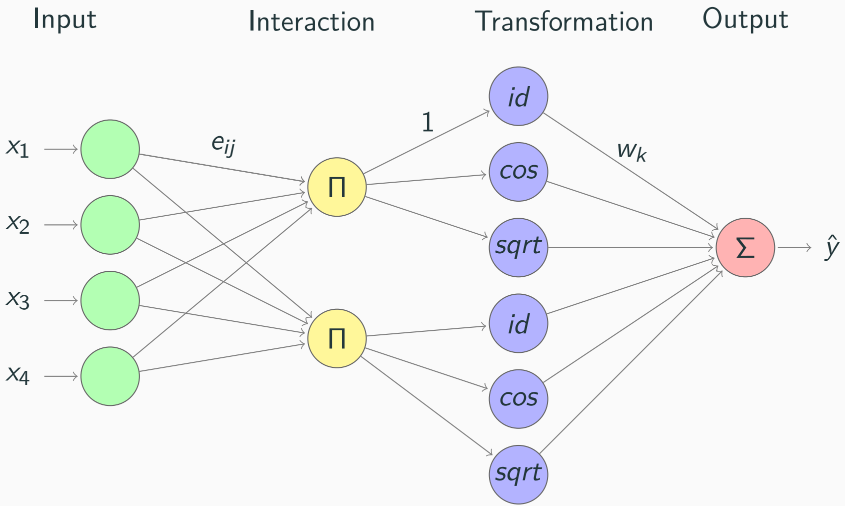
SymTree

```
expand leaf = expand' leaf terms
  where terms = interaction leaf U transformation leaf

expand' leaf terms = node : expand' leaf leftover
  where (node, leftover) = greedySearch leaf terms
```

IT-ELM

Interaction-Transformation Extreme Learning Machine, it generates lots of random interactions, enumerates the transformations for each interaction and then adjust the weight of the terms using l_0 or l_1 regularization.



Input
Interaction
Transformation
Output
x_1
x_2
x_3
x_4
e_{ij}
Π
Π
1
id
cos
$sqrt$
id
cos
$sqrt$
w_k
Σ
$\hat{y}$

Experiments

Data sets

Data set	Features	5-Fold / Train-Test
Airfoil	5	5-Fold
Concrete	8	5-Fold
CPU	7	5-Fold
energyCooling	8	5-Fold
energyHeating	8	5-Fold
TowerData	25	5-Fold
wineRed	11	5-Fold
wineWhite	11	5-Fold
yacht	6	5-Fold
Chemical-I	57	Train-Test
F05128-f1	3	Train-Test
F05128-f2	3	Train-Test
Tower	25	Train-Test

Methods

For the sets with folds:

- Each algorithm was run 6 times per fold and the median of the RMSE of the test set is reported
- SymTree was run 1 time per fold (deterministic)

For the sets with train-test split:

- Each algorithm was run 10 times and the median of the RMSE for the test set is reported
- SymTree was run 1 time per data set

Results

For a complete table:

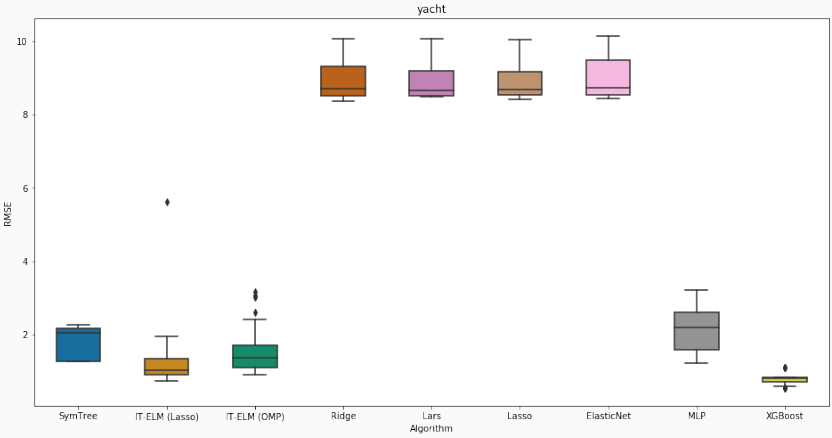
Binder

Cell –> Run All

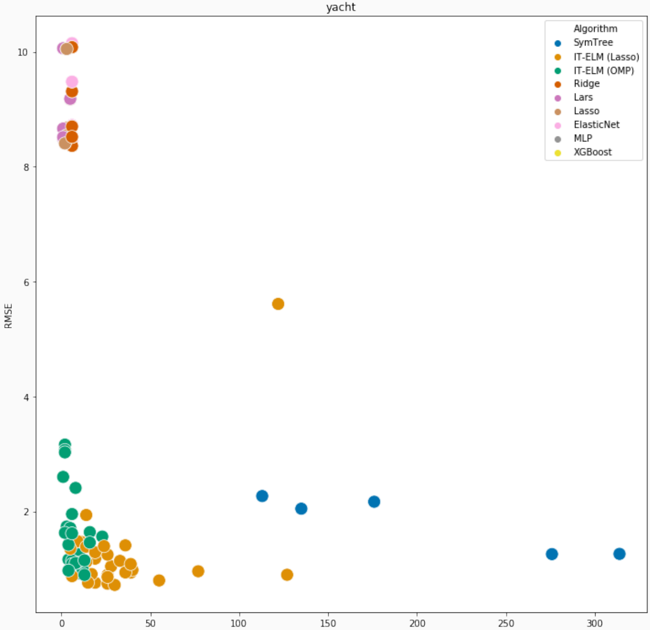
Results

Algorithm	Rank
XGBoost	2
SymTree	3.3
IT-ELM (Lasso)	3.53846
IT-ELM (OMP)	3.84615
MLP	5
Ridge	5.69231
Lars	5.76923
Lasso	6.76923
ElasticNet	7.76923

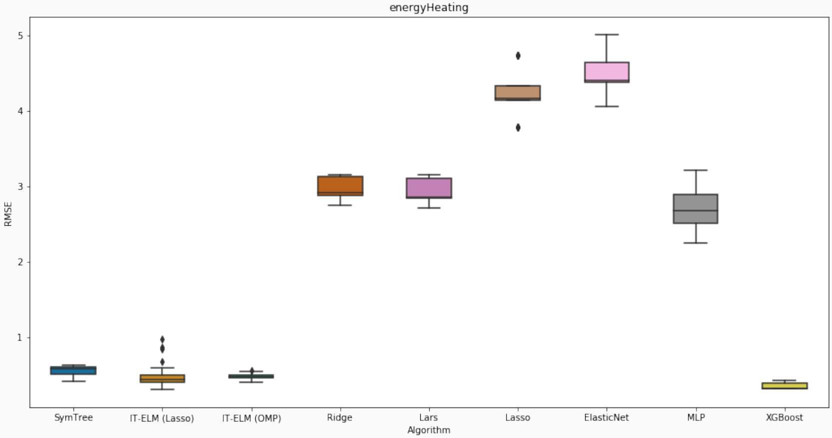
Results


yacht
RMSE
10
8
6
4
2
SymTree
IT-ELM (Lasso)
IT-ELM (OMP)
Ridge
Lars
Lasso
ElasticNet
MLP
XGBoost
Algorithm

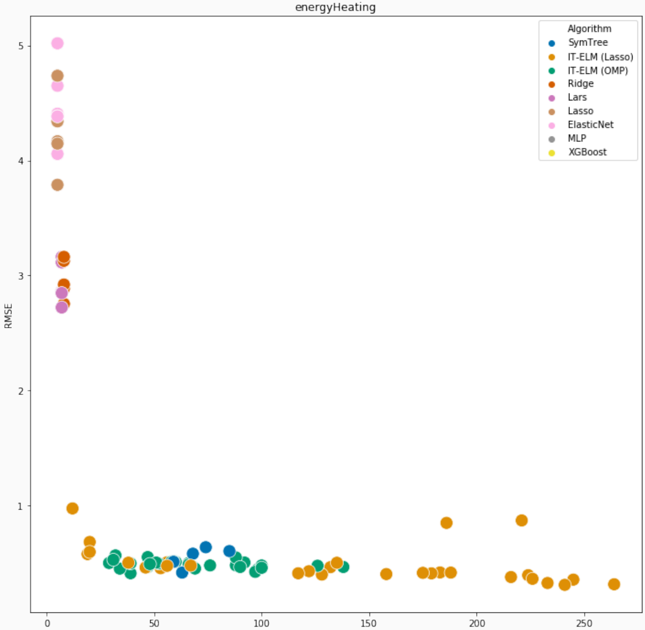
Results


yacht
RMSE
10
8
6
4
2
0
50
100
150
200
250
300
Algorithm
SymTree
IT-ELM (Lasso)
IT-ELM (OMP)
Ridge
Lars
Lasso
ElasticNet
MLP
XGBoost

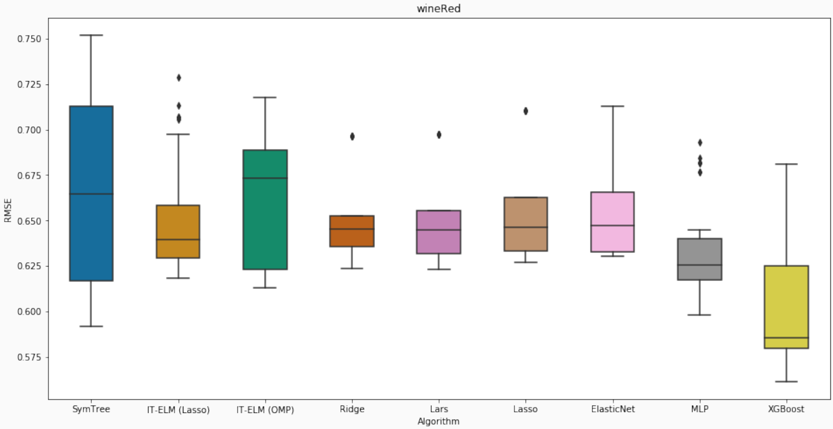
Results


energyHeating
RMSE
5
4
3
2
1
SymTree
IT-ELM (Lasso)
IT-ELM (OMP)
Ridge
Lars
Lasso
ElasticNet
MLP
XGBoost
Algorithm

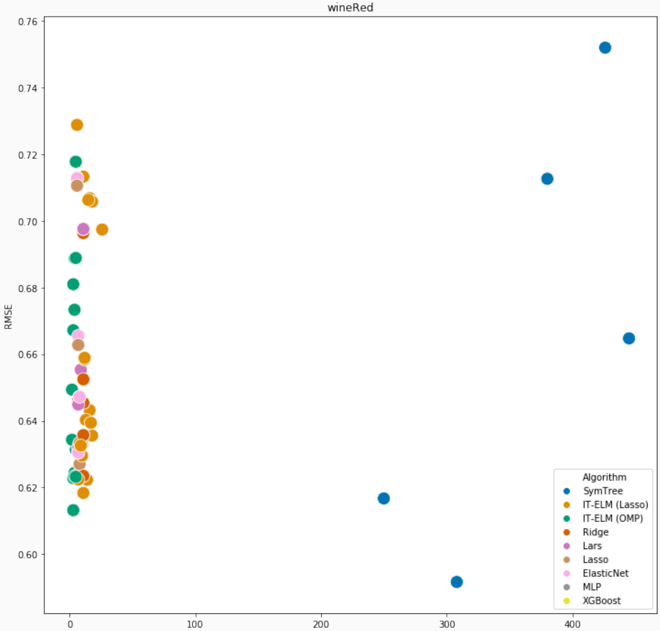
Results


energyHeating
RMSE
5
4
3
2
1
0
50
100
150
200
250
Algorithm
SymTree
IT-ELM (Lasso)
IT-ELM (OMP)
Ridge
Lars
Lasso
ElasticNet
MLP
XGBoost

Results


wineRed
0.750
0.725
0.700
0.675
0.650
0.625
0.600
0.575
RMSE
SymTree
IT-ELM (Lasso)
IT-ELM (OMP)
Ridge
Lars
Lasso
ElasticNet
MLP
XGBoost
Algorithm

Results


wineRed
RMSE
0.76
0.74
0.72
0.70
0.68
0.66
0.64
0.62
0.60
0
100
200
300
400
Algorithm
SymTree
IT-ELM (Lasso)
IT-ELM (OMP)
Ridge
Lars
Lasso
ElasticNet
MLP
XGBoost

Sample equation

CPU:

$$\approx 0.86 \cdot \mathit{cache} + 0.12 \cdot 10^{-6} \cdot \mathit{maxMem}\sqrt{\mathit{maxChan} \cdot \mathit{minMem}}$$

Sample equation

CPU:

$$\approx 0.86 \cdot \mathit{cache} + 0.12 \cdot \mathit{maxMem}(MB)\sqrt{\mathit{maxChan} \cdot \mathit{minMem}(MB)}$$

Sample equation

CPU:

$$\approx 0.86 \cdot cache + 0.12 \cdot maxMem(MB)\sqrt{maxChan}\sqrt{minMem(MB)}$$

More cache = more performance! (sounds about right)

Sample equation

CPU:

$$\approx 0.86 \cdot \mathit{cache} + 0.12 \cdot \mathit{maxMem}(MB)\sqrt{\mathit{maxChan}}\sqrt{\mathit{minMem}(MB)}$$

The original paper isn't clear about it, but it seems that max/min Mem refers to the range of machine tests with a given CPU. So the second term may represent the existence of not measured variables proportional to memory and channels of the experimented machines.

Conclusions

Summary

- The Interaction-Transformation representation can help to eliminate *complicated* expressions from the symbolic regression search space.
- Two algorithms created so far: SymTree, a search-based heuristic, and IT-ELM, based on extreme learning machines.
- The results show a good compromise between model accuracy and simplicity.

Future research

- Generalization as a Algebraic Data Type
- Use this representation for classification, program synthesis, etc.
- Broaden the search space a little bit
- Explore other search heuristics (evolutionary based)

Try it!

You can try a lightweight version of SymTree at:

https://galdeia.github.io/

It works even on midrange Smartphones!

Some problems with the provided data sets

The *folds* data sets were provided by some authors that extensively used for GP performance comparison, but:

Forest Fire contains many samples with target $= 0$, because most of the time the forests did not caught fire.

CPU should use the last but one column as the target variable, the last column is just the predicted values from the original paper.