Optimal superposition trees restoration in symbolic regression

Radoslav Neychev, Innokenty Shibaev, Vadim Strijov

Moscow Institute of Physics and Technology

IDP-2020, Moscow, Russia

Optimal superposition trees restoration

Goal: Provide an approach to simple and easy to interpret regression models generation

Proposed solution:

- Predict the computational graph adjacency matrix with classifier
- Fix the predicted adjacency matrix to meet the arity constraints (if necessary)
- Restore the superposition tree from the predicted adjacency matrix

Method: The proposed approach to superposition tree restoration based on Prize-Collecting Steiner Tree algorithm

Literature overview

- Fast PCST implementation
  Hegde C., Indyk P., Schmidt L. A fast, adaptive variant of the Goemans-Williamson scheme for the prize-collecting Steiner tree problem, Workshop of the 11th DIMACS Implementation Challenge. Providence, Rhode Island, 2014

- Approximate k-MST algorithm
  Chudak F. A., Roughgarden T., Williamson D. P. Approximate k-MSTs and k-Steiner trees via the primal-dual method and Lagrangean relaxation, Mathematical Programming, 2004

- Generation of models easy to be interpreted
  A.M.Bochkarev, I.L.Sofronov, V.V.Strijov, Generation of expertly-interpreted models for prediction of core permeability, Systems and Means of Informatics, 2017.

- Weight Agnostic Neural Networks as computational graphs
  Adam Gaier and David Ha, Weight Agnostic Neural Networks, 2019

Problem

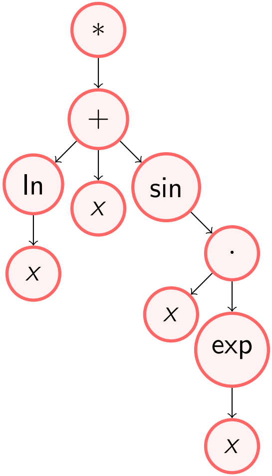
There given

- Set of basic functions $g_1, \dots, g_l$
- Family of generative superpositions $\mathcal{F} = \{f : f = \sup(g_1, \dots, g_l)\}$, $f_i \in \mathcal{F} \quad \forall i$
- Collection of datasets $\mathcal{A} = \{\mathbf{A}_1, \dots, \mathbf{A}_N\}$ *homogeneous* w.r.t. the family of generative superpositions $\mathcal{F}$

Superposition representation

Directed weighted graph $G = (V, E)$ with colored vertices v_i and special vertex r. Edge $e_i \in E$ e_i is assigned with a weight $w(e_i) = c_i \in [0, 1]$, vertex $v_i \in V$ is assigned with color $t(v_i) = t_i \in \mathbb{N}$. Graph is represented with adjacency matrix.

$f(x) = \ln(x) + x + \sin(x \cdot \exp(x))$



Computational graph and adjacency matrix

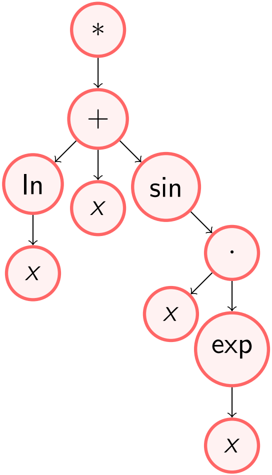
ar	$f(.)$	$*$	$+$	ln	sin	$\cdot$	exp	x
1	$*$	0	1	0	0	0	0	0
3	$+$	0	0	1	1	0	0	1
1	ln	0	0	0	0	0	0	1
1	sin	0	0	0	0	1	0	0
2	$\cdot$	0	0	0	0	0	1	1
1	exp	0	0	0	0	0	0	1

Adjacency matrix

ar	$f(.)$	$*$	$+$	ln	sin	$\cdot$	exp	x
1	$*$	0.2	0.7	0.5	0.4	0.5	0.3	0.2
3	$+$	0.3	0.2	1.	0.8	0.6	0.3	0.7
1	ln	0.3	0.2	0	0.	0.1	0.5	0.5
1	sin	0.1	0.4	0	0.5	0.9	0.2	0.5
2	$\cdot$	0.3	0.	0.3	0.5	0.	0.8	0.6
1	exp	0.3	0.3	0.4	0.1	0.5	0.4	0.4

Probability of adjacency matrix

$f(x) = \ln(x) + x + \sin(x \cdot \exp(x))$



Problem statement

Find

The optimal superposition f^* for every fixed pair $\mathbf{A} = (\mathbf{X}, \mathbf{y})$ minimizes the *loss function* S:

$$f^* = \arg\min_{f \in \mathcal{F}} S(f|\mathbf{X}, \mathbf{y}),$$

The squared loss is used hereinafter:

$$S(f|\mathbf{X}, \mathbf{y}) = \|f(\mathbf{X}) - \mathbf{y}\|_2^2.$$

Composition of basic functions satisfying constraints can be treated as f.

Solution

- Predict the edges probabilities in the adjacency matrix.
- Restore the computational graph.

k-MST PCST problem statement

Rooted k-MST (k-Minimum spanning tree)

Given weighted graph $G = (V, E)$ with root r, and edge weights $w(e_i) = w_i > 0,\ e_i \in E$. Construct a minimum-weigh directed tree with root vertex r covering at least k vertices.

Rooted PCST (Prize-Collecting Steiner Tree)

Given weighted graph $G = (V, E)$ with root r, and edge weights $w(e_i) = c_i \geqslant 0,\ e_i \in E$, every vertex $v_i \in V$ is assigned with a *prize* $\pi(v_i) = \pi_i \geqslant 0$. Construct a tree T with root r which minimizes the following functional:

$$\sum_{e \in E} c_e x_e + \sum_{S \subseteq V \setminus \{r\}} \pi(S) z_S.$$

k-MST PCST problem statement

Linear Programming PCST ($k - MST$) problem

With relaxed constraints

$$\begin{aligned} \underset{\substack{x_e, z_S \\ e \in E, S \subseteq V\setminus\{r\}}}{\text{minimize}} \quad & \min \sum_{e \in E} c_e x_e + \lambda \left(\sum_{S \subseteq V\setminus\{r\}} |S| z_S - (n-k) \right) \\ \text{s.t.} \quad & \sum_{e \in \delta(S)} x_e + \sum_{T : T \supseteq S} z_T \geq 1, \ \forall S \subseteq V\setminus\{r\}, \ v \in S, \\ & x_e \in [0,1], \ \forall e \in E \\ & z_S \in [0,1], \ S \subseteq V\setminus\{r\} \end{aligned}$$

In strict formulation $x_e \in \{0,1\}$, $x_e = 1$ denotes that the edge is included into the tree.
By analogy, $z_S \in \{0,1\}$, $z_S = 1$ for set $S = V \setminus T$

Computational experiment

Algorithms

The following algorithms are used for matrix restoration

- DFS
- BFS
- Prim's algorithm
- k-MST via PCST (directed and undirected)
- k-MST $+$ DFS (directed and undirected)
- k-MST $+$ BFS (directed and undirected)
- $k-MST$ $+$ Prim's algorithm (directed and undirected)

Test data

Synthetic data with following properties is used:

- All the functions are taking only one input.
- The arities of the function are generated by Binomial distribution (so there are many functions with small arity).
- 50 sets of arity values (with length from 5 to 20)
- 20 function for every set
- 5 copies of every function with noise from Uniform distribution
- Linear calibration to interval [0, 1]

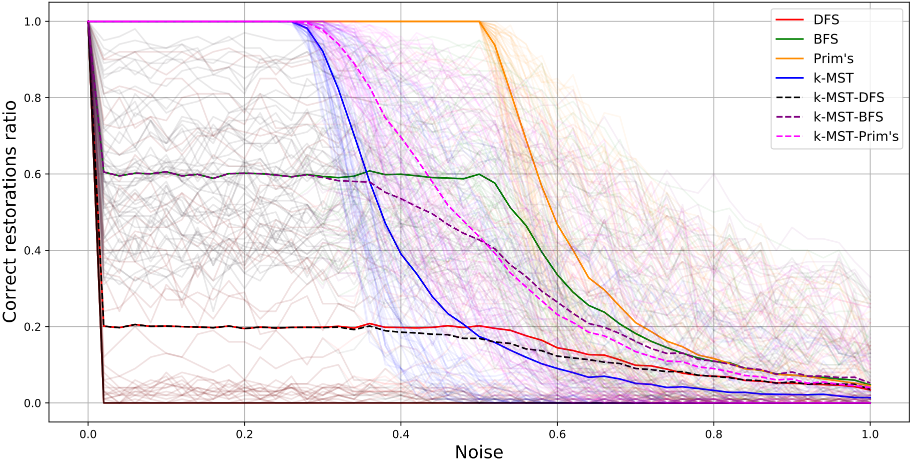
Quality measure

Quality is measured as correct restorations ratio.

$$\mathrm{Acc}(\mathrm{R}, \mathrm{N}, \mathcal{M}) = \frac{1}{|\mathcal{M}|} \sum_{M \in \mathcal{M}} [R(N(M)) = M],$$

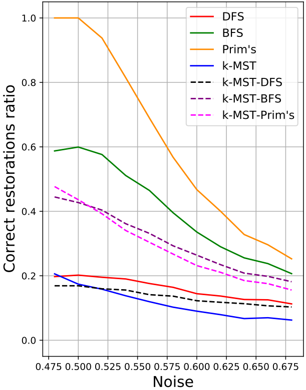
where N is the noise function and R is the restoration algorithm.

Ratio of correct restorations, no orientation



Performance is averaged over 100 runs with random initialization.
Arity of functions vary from 5 to 20, noise varies from 0 to 1.
Algorithms based on k-MST use symmetrized adjacency matrix.

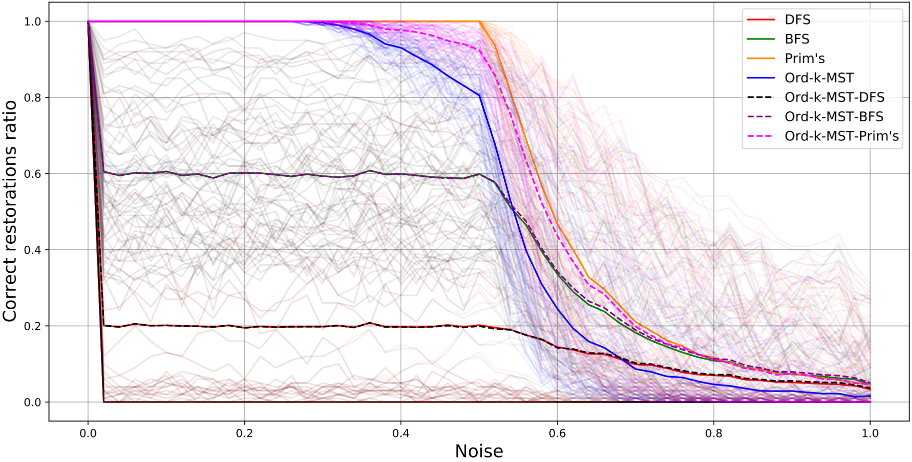
Ratio of correct restorations, oriented graph



Noise	.50	.52	.54	.56	.58
DFS	.2	.2	.19	.18	.16
BFS	.6	.58	.51	.46	.4
Prim's algorithm	**1.0**	**.94**	**.81**	**.69**	**.57**
k-MST	.17	.16	.14	.12	.1
k-MST-DFS	.17	.16	.16	.14	.14
k-MST-BFS	.43	.4	.36	.33	.29
k-MST-Prim's	.44	.39	.34	.33	.27

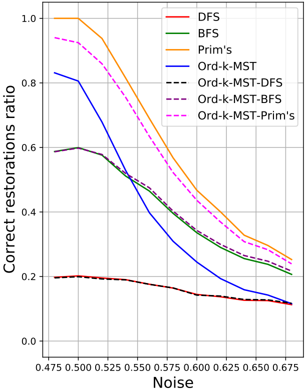
Illustration of algorithms behaviour near the 0.5 threshold of the noise value with more details. Prim's algorithm is the most persistent.

Ratio of correct restorations, orientated case



Performance is averaged over 100 runs with random initialization.
Arity of functions vary from 5 to 20, noise varies from 0 to 1.
Algorithms based on k-MST use original adjacency matrix.

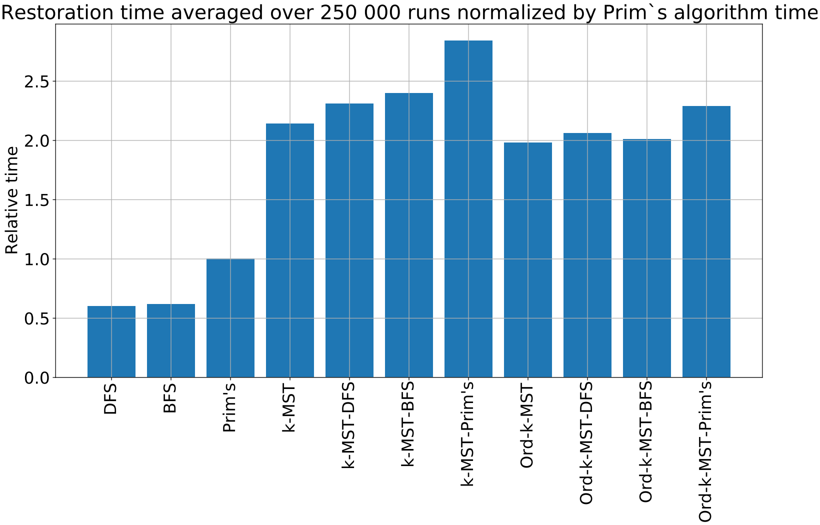
Ratio of correct restorations, orientated case



Noise	.50	.52	.54	.56	.58
DFS	.2	.2	.19	.18	.16
BFS	.6	.58	.51	.46	.4
Prim's algorithm	**1.0**	**.94**	**.81**	**.69**	**.57**
Ord-*k*-MST	.81	.68	.53	.4	.31
Ord-*k*-MST-DFS	.2	.19	.19	.18	.16
Ord-*k*-MST-BFS	.6	.58	.52	.47	.4
Ord-*k*-MST-Prim's	.92	.86	.76	.63	.52

Illustration of algorithms behaviour near the 0.5 threshold of the noise value with more details. Prim's algorithm is the most persistent, the Ord-*k*-MST-Prim's algorithm show much closer results.

Comparing the algorithms performance



Conclusion

- The proposed algorithm delivers accurate results, but is more prone to noise in the superposition matrix.
- The approach based on Prim's algorithm delivers the most accurate results and is the most resistant to small noise in data.
- Approaches based on BFS and DFS are unable to restore the original superposition if noise is present. PCST algorithm with BFS used for superposition matrix restoration shows mediocre results.

Backup