List of primitive functions

Description	In	N in	Out	N out	Comm	Param
Nominal to binary	nom	1	bin	1–4	-	Yes
Ordinal to binary	ord	1	bin	1–4	-	Yes
Linear to linear segments	lin	1	lin	1–4	-	Yes
Linear segments to binary	lin	1	bin	1–4	-	Yes
Get one column of n-matrix	bin	1–4	bin	1	-	Yes
Conjunction	bin	2–6	bin	1	Yes	-
Disjunction	bin	2–6	bin	1	Yes	-
Negate binary	bin	1	bin	1	-	-
Logarithm	lin	1	lin	1	-	-
Hyperbolic tangent sigmiod	lin	1	lin	1	-	-
Logistic sigmoid	lin	1	lin	1	-	-
Sum	lin	2–3	lin	1	Yes	-
Difference	lin	2	lin	1	No	-
Multiplication	lin,bin	2–3	lin	1	Yes	-
Division	lin	2	lin	1	No	-
Inverse	lin	1	lin	1	-	-
Polynomial transformation	lin	1	lin	1	-	Yes
Radial basis function	lin	1	lin	1	-	Yes
Monomials: $x\sqrt{x}$, etc.	lin	1	lin	1	-	-

There given

- the measured features $\Xi = \{\xi\}$,
- the expert-given primitive functions $G = \{g(\mathbf{b}, \xi)\}$,

$$g : \xi \mapsto x;$$

- the generation rules: $\mathcal{G} \supset G$, where the superposition $g_k \circ g_l \in \mathcal{G}$ w.r.t. numbers and types of the input and output arguments;
- the simplification rules: g_u is not in $\mathcal{G}$, if there exist a rule

$$r : g_u \mapsto g_v \in \mathcal{G}.$$

The result is

the set of the features $X = \{\mathbf{x}_1, \dots, \mathbf{x}_j, \dots, \mathbf{x}_n\}$.

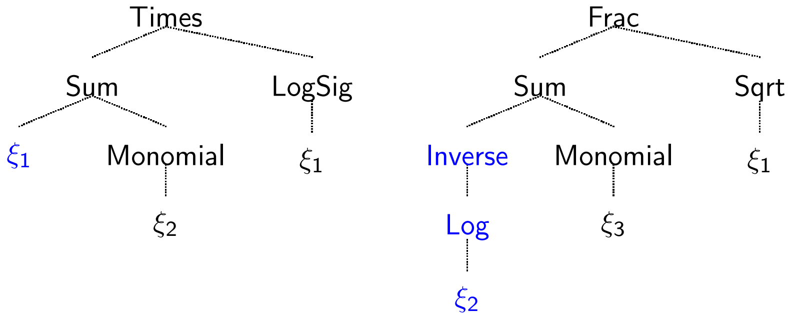
The number of features exceeds the number of clients!

Examples of generated features

- **Frac**(Period of residence, Undeclared income)
- **Frac**(**Seg**(Period of employment), Term of contract)
- **And**(Income confirmation, Bank account)
- **Times**(**Seg**(Score hour), **Frac**(**Seg**(Period of employment), Salary))

1. Select random nodes in two features,
2. exchange the corresponded subtrees,
3. modify the function at a random node for another one from the primitive set.

Any modification must result an admissible superposition.



1. Consider cartesian product $G \times \Xi$ of the set of non-generated variables Ξ the primitives G. Denote by a_ι the superpositions $g_v(\xi_u)$
2. Product superpositions a_ι no more than P times

$$a_\iota = g_v(\xi_u), \quad \text{where the index} \quad \iota = (v-1)U + u$$

and

$$x_j = \prod \underbrace{a_{\iota_1} \dots a_{\iota_p}}_{p \text{ times}}, \quad \text{where} \quad \iota \in \{1, \dots, UV\}, \quad p \in \{1, \dots, P\}.$$

In the other words

$$\xi_u \xrightarrow{g_v} g_v(\xi_u) \equiv a_\iota \xrightarrow{\prod^p} x_j, \qquad j \in \mathcal{J}.$$

Consider the linear models as the polynomial with a monomial $a_\iota = g_v(\xi_u)$

$$f(\mathbf{w}, \mathbf{x}) = \sum_{\iota=1}^{UV} w_\iota a_\iota + \sum_{\iota=1}^{UV} \sum_{\kappa=1}^{UV} w_{\iota\kappa} a_\iota a_\kappa + \sum_{\iota=1}^{UV} \sum_{\kappa=1}^{UV} \sum_{\tau=1}^{UV} w_{\iota\kappa\tau} a_\iota a_\kappa a_\tau + \cdots .$$

Set of the primitive functions G

Let $G = \{g_1, \dots, g_l | g = g(\mathbf{b}, \cdot, \dots, \cdot)\}$ such that there are given

- the function $g : (\mathbf{b}, x) \mapsto x'$,
- its parameters $\mathbf{b}$ (the empty set is allowed),
- number of arguments $v(g)$ of the function g and the order of the arguments (zero arguments is allowed),
- domain $\text{dom}(g)$ and codomain $\text{cod}(g)$.

Consider the model $f(\mathbf{w}, \mathbf{x})$ as a superposition

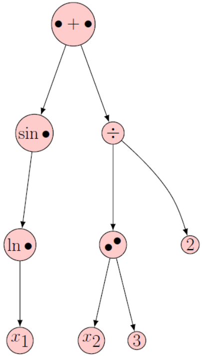
$$f(\mathbf{w}, \mathbf{x}) = (g_{i(1)} \circ \dots \circ g_{i(K)})(\mathbf{x}), \text{ where } \mathbf{w} = [\mathbf{b}_{i(1)}^{\mathsf{T}}, \dots, \mathbf{b}_{i(K)}^{\mathsf{T}}]^{\mathsf{T}}.$$

The admissible superposition f

is the superposition, which satisfies

$$\text{cod}(g_{i(k+1)}) \subseteq \text{dom}(g_{i(k)}), \text{ for any } k = 1, \dots, K-1.$$

- The vertex V_i corresponds to the primitive function $g_{s(i)}$.
- The number of outgoing nodes from the vertex V_i equal the number of arguments of $v(g_{s(i)})$.
- The order of the outgoing nodes from the vertex V_i equals the order of the arguments of $g_{s(i)}$.
- The leaves of the tree Γ_f corresponds to the independent variables x_i and constants; they are treated as the primitives $g(\emptyset)$.



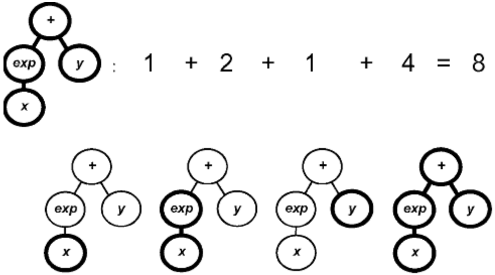
The tree for the superposition $\sin(\ln x_1) + \frac{x_2^3}{2}$

The superposition depth $d(f)$ is

maximum depth of the tree Γ_f, number of the nodes V from the root to the most distanced leaf.

The superposition complexity $C(f)$ is

the number of all admissible subtrees of the tree Γ_f.



Given: $G=\{g_u,h_v|u\in\mathcal{U},v\in\mathcal{V}\}$ is a set of the primitive functions of one and two arguments, $\mathbf{x}=\{x_j|j\in\mathcal{J}\}$ — independent variables.

Step 1: $\mathcal{F}_1=\left\{f_s^{(1)}\right\}=\{g_u(x_j)\}\cup\{h_v(x_j,x_k)\}$,
$k\in\mathcal{J}$, $s\in\left\{1,\ldots,|\mathcal{U}|\cdot|\mathcal{J}|+|\mathcal{V}|\cdot|\mathcal{J}|^2\right\}$.

Step k:

(Gen) Append to $\mathcal{F}$ the set

$$\mathcal{F}^{(k)}=\left\{f_s^{(k)}\right\}=\left\{g_u\left(f_{s'}^{(k-1)}\right)\right\}\cup\left\{h_v\left(f_{s''}^{(k-1)},f_{s'''}^{(k-1)}\right)\right\},$$

(Rem) which does not contain the superpositions, isomorphic to $g_u\left(f_s^{(k)}\right)$ and $h_v\left(f_s^{(k)},f_{s'}^{(k)}\right)$ form the sets $\mathcal{F}^{(k)}\ldots\mathcal{F}^{(1)}$.

Exhaustive search in the set of the generalized linear models

$$\mu(y) = w_0 + \alpha_1 w_1 x_1 + \alpha_2 w_2 x_2 + \ldots + \alpha_R w_R x_R.$$

Here $\alpha \in \{0, 1\}$ is the structural parameter.

Find a model defined by the set $\mathcal{A} \subseteq \mathcal{J}$:

α_1	α_2	$\ldots$	$\alpha_{\lvert\mathcal{J}\rvert}$
1	0	...	0
0	1	...	0
...	...	...	...
1	1	...	1

Exhaustive search in the set of the generalized linear models

$$\mu(y) = w_0 + \alpha_1 w_1 x_1 + \alpha_2 w_2 x_2 + \ldots + \alpha_R w_R x_R.$$

Here $\alpha \in \{0, 1\}$ is the structural parameter.

Find a model defined by the set $\mathcal{A} \subseteq \mathcal{J}$:

α_1	α_2	...	$\alpha_{\lvert\mathcal{J}\rvert}$
1	0	...	0
0	1	...	0
...	...	...	...
1	1	...	1

Discrete genetic algorithm for feature selection (simple ver.)

1. There are set of binary vectors $\{\mathbf{a}_1, \dots, \mathbf{a}_P\}$, $\mathbf{a} \in \{0,1\}^n$;
2. get two vectors $\mathbf{a}_p, \mathbf{a}_q$, $p, q \in \{1, \dots, P\}$;
3. chose random number $\nu \in \{1, \dots, n-1\}$;
4. split both vectors and change their parts:

$$[a_{p,1}, \dots, a_{p,\nu}, a_{q,\nu+1}, \dots, a_{q,n}] \to \mathbf{a'}_p,$$

$$[a_{q,1}, \dots, a_{q,\nu}, a_{p,\nu+1}, \dots, a_{p,n}] \to \mathbf{a'}_q;$$

5. choose random numbers $\eta_1, \dots, \eta_Q \in \{1, \dots, n\}$;
6. invert positions $\eta_1, \dots, \eta_Q$ of the vectors $\mathbf{a'}_p, \mathbf{a'}_q$;
7. repeat items 2-6 $P/2$ times;
8. evaluate the obtained models.

Repeat R times; here P, Q, R are the parameters of the algorithm and n is the number of the corresponding model features.

1. There are set of binary vectors $\{\mathbf{a}_1, \dots, \mathbf{a}_P\}$, $\mathbf{a} \in \{1, \dots, k\}^n$;
2. get two vectors $\mathbf{a}_p, \mathbf{a}_q$, $p, q \in \{1, \dots, P\}$;
3. chose random number $\nu \in \{1, \dots, n-1\}$;
4. split both vectors and change their parts:

$$[a_{p,1}, \dots, a_{p,\nu}, a_{q,\nu+1}, \dots, a_{q,n}] \to \mathbf{a}'_p,$$

$$[a_{q,1}, \dots, a_{q,\nu}, a_{p,\nu+1}, \dots, a_{p,n}] \to \mathbf{a}'_q;$$

5. choose random numbers $\eta_1, \dots, \eta_Q \in \{1, \dots, n\}$;
6. replace values in positions $\eta_1, \dots, \eta_Q$ of the vectors $\mathbf{a}'_p, \mathbf{a}'_q$ for random values from $\{1, \dots, k\}$;
7. repeat items 2-6 $P/2$ times;
8. evaluate the obtained models.

Repeat R times; here P, Q, R are the parameters of the algorithm and k is desired number of categories.