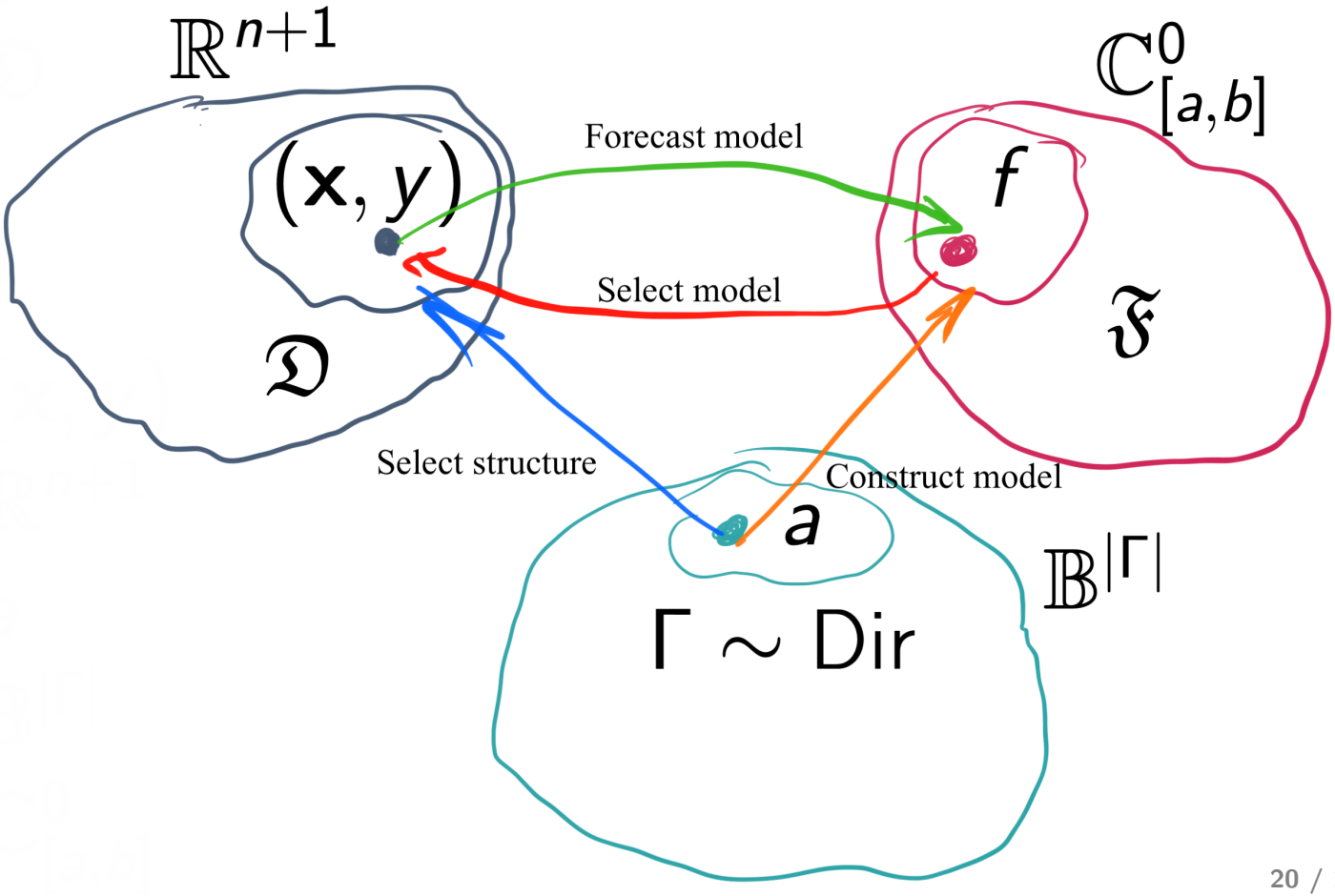
Multimodelling as a way to select and forecast models



Exhaustive search (linear models)

The superposition is

$$y = w_0 + \alpha_1 w_1 x_1 + \alpha_2 w_2 x_2 + \ldots + \alpha_W w_W x_W.$$

Here $\alpha \in \{0, 1\}$ is part of the model. Exhaustive search is

α_1	α_2	$\ldots$	α_W
1	0	$\ldots$	0
0	1	$\ldots$	0
$\ldots$	$\ldots$	$\ldots$	$\ldots$
1	1	$\ldots$	1

.

Discrete genetic algorithm for feature selection (simple ver.)

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

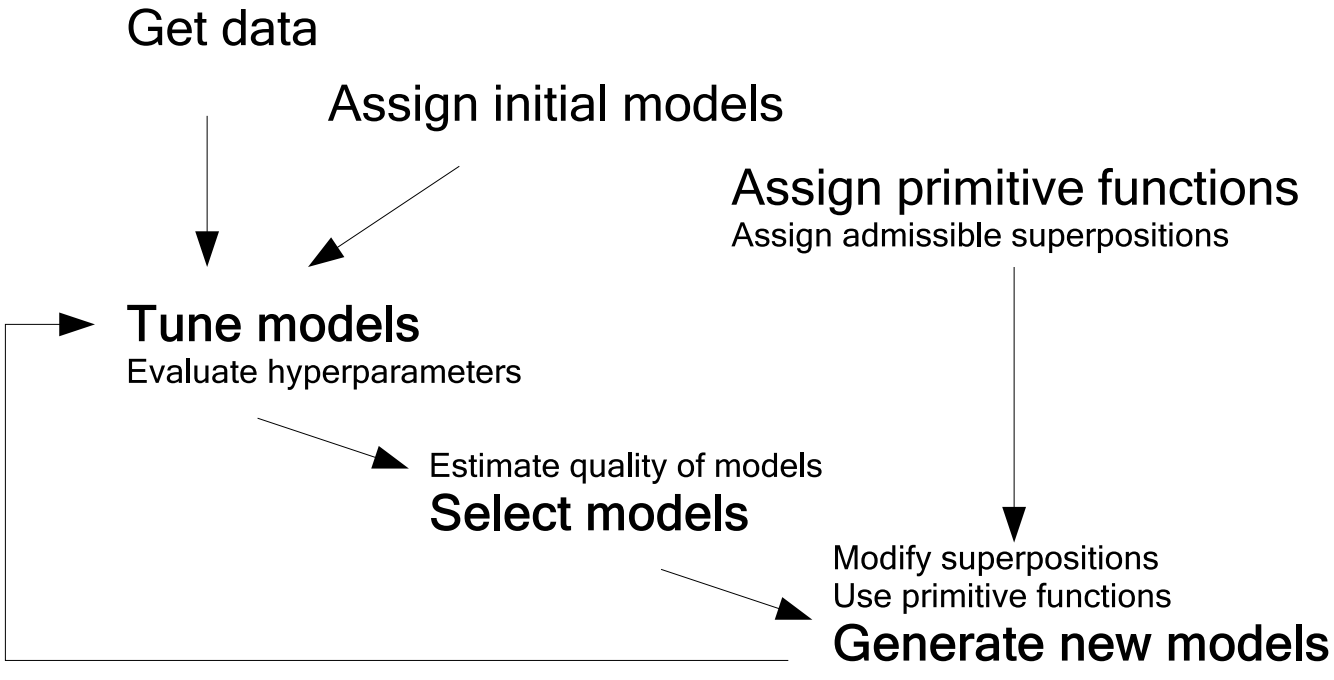
$$[a_{p,1}, \ldots, a_{p,\nu}, a_{q,\nu+1}, \ldots, a_{q,n}] \to \mathbf{a'}_p,$$

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

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

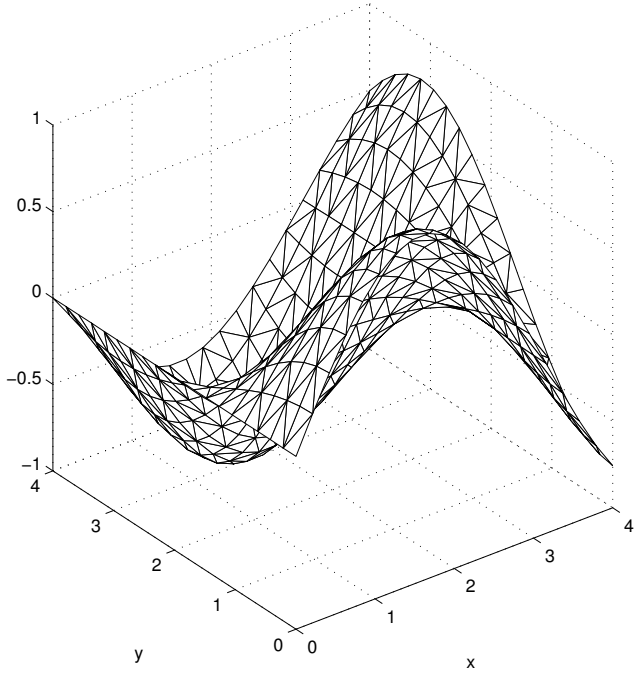
Repeat R times; here P, Q, R are the parameters of the algorithm and n is the number of the corresponding model features.

The process of the model construction



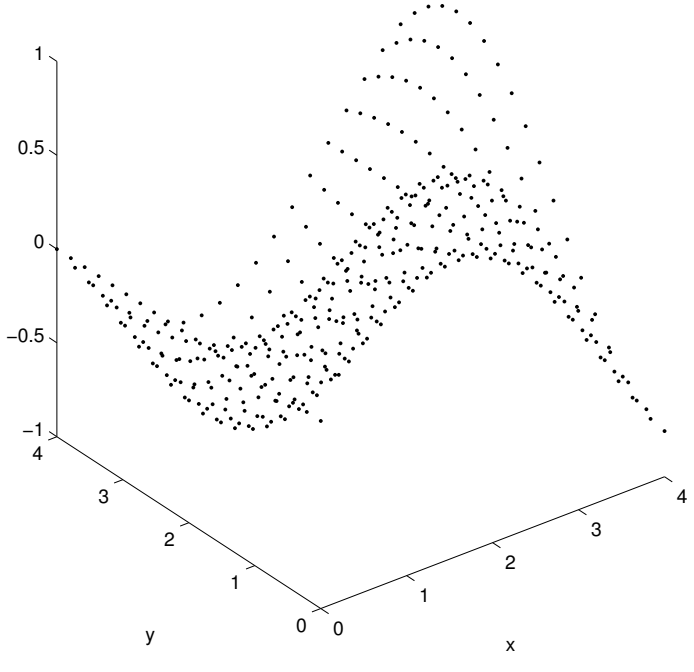
Think of a model

Let it be $y = f(\mathbf{w}, \mathbf{x}) = sin(x_1) * sin(w_1 x_2 + w_2)$.



Given data

The corresponded sample set is shown; it has 380 samples.



Given primitive functions

Function	Description	Parameters
	$g(\mathbf{b}, x_1, x_2)$	
plus	$y = x_1 + x_2$	–
times	$y = x_1 x_2$	–
	$g(\mathbf{b}, x_1)$	
divide	$y = 1/x$	–
multiply	$y = ax$	a
add	$y = x + a$	a
normal	$y = \frac{\lambda}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\xi)^2}{2\sigma^2}\right) + a$	λ, σ, ξ, a
linear	$y = ax + b$	a, b
parabolic	$y = ax^2 + bx + c$	a, b, c
sin	$y = sin(x)$	–
logsig	$y = \frac{\lambda}{1+\exp(-\sigma(x-\xi))} + a$	λ, σ, ξ, a

Set of the generated models

Let the generated models $\mathcal{F} = \{f_i\}$ be a set of admissible superpositions of the primitive functions $G = \{g\}$.

Expert information

Experts assign the initial models

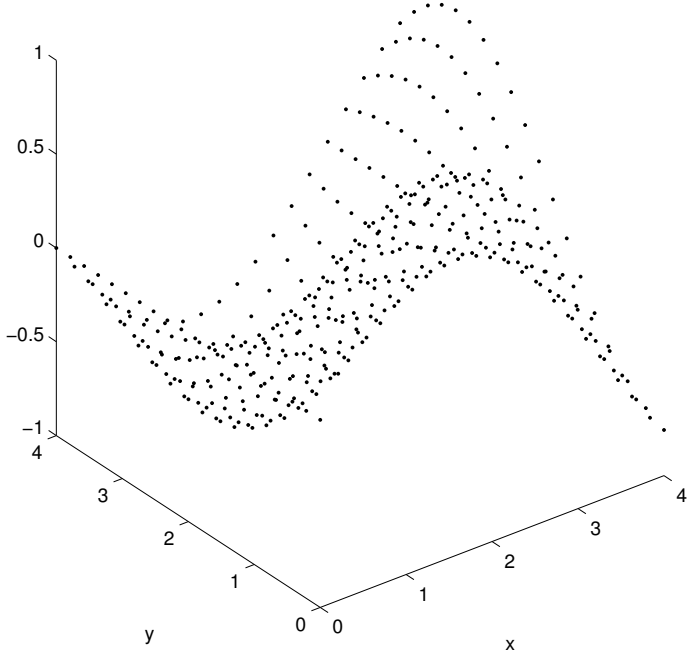
$$\begin{aligned} f_1: &\quad y = \text{linear}(x_1), \\ f_2: &\quad y = \text{normal}(x_2). \end{aligned}$$

And the initial conditions

1. the model complexity:
   $$\begin{cases} \text{number of primitives in a superposition } g \text{ no more than } 8, \\ \text{number of parameters } w \text{ no more than } 10; \end{cases}$$
2. the target function is sum of squared errors, SSE.

Competitive models

Given data

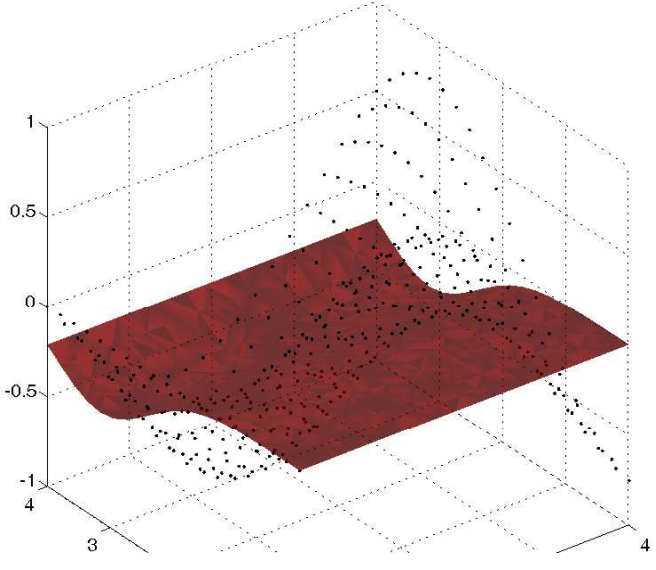


Competitive models

normal($w_{1:3}$,x_2)

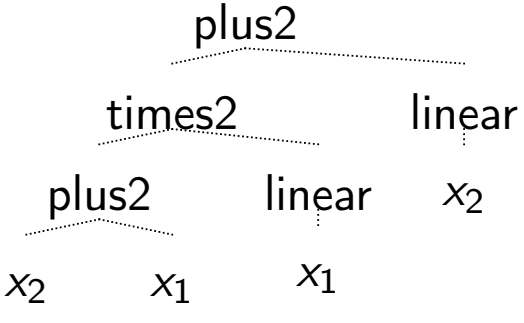
normal

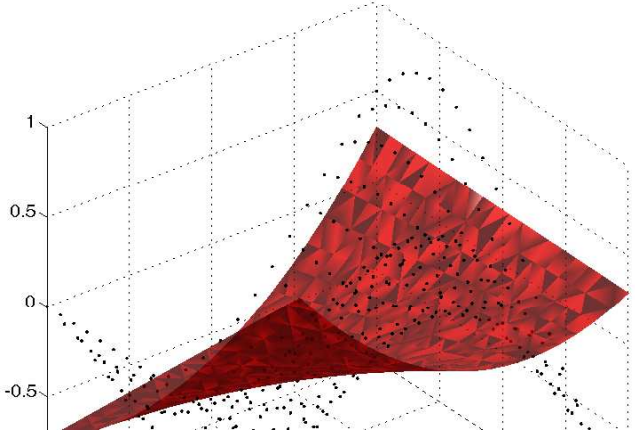
x_2



Competitive models

plus2(∅,times2(∅,plus2(∅,x_2,x_1),linear($w_{1:2}$,x_1)),linear($w_{3:4}$,x_2))

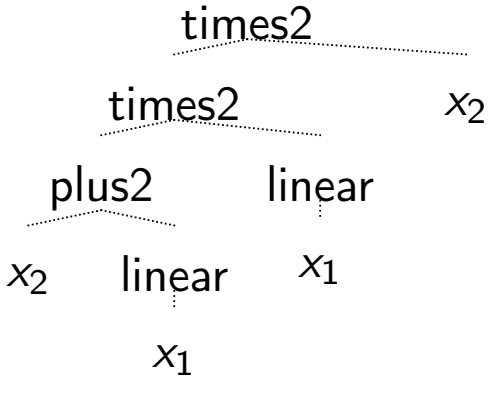


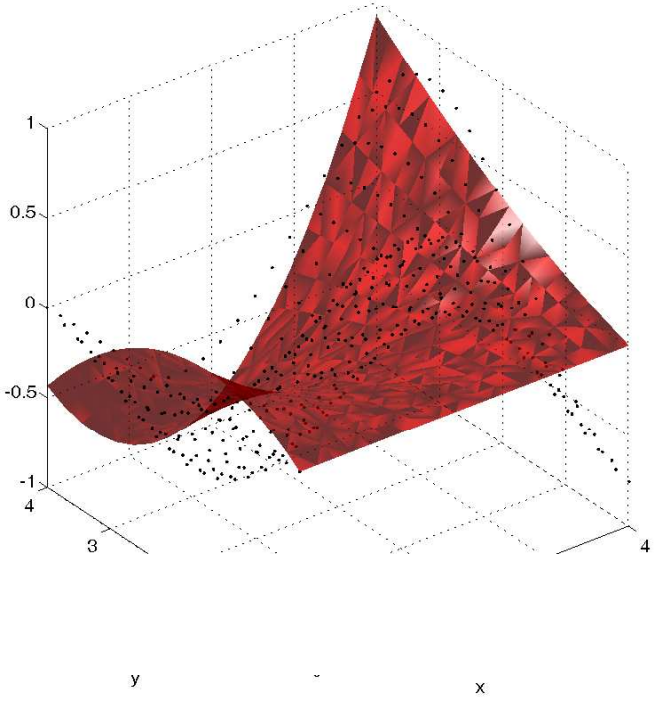


y x

Competitive models

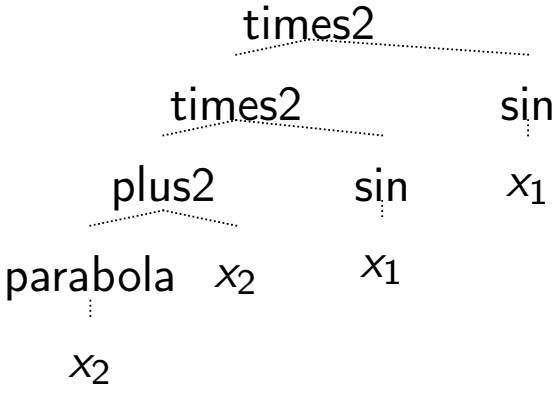
$$\text{times2}(\emptyset,\text{times2}(\emptyset,\text{plus2}(\emptyset,x_2,\text{linear}(w_{1:2},x_1)),\text{linear}(w_{3:4},x_1)),x_2)$$

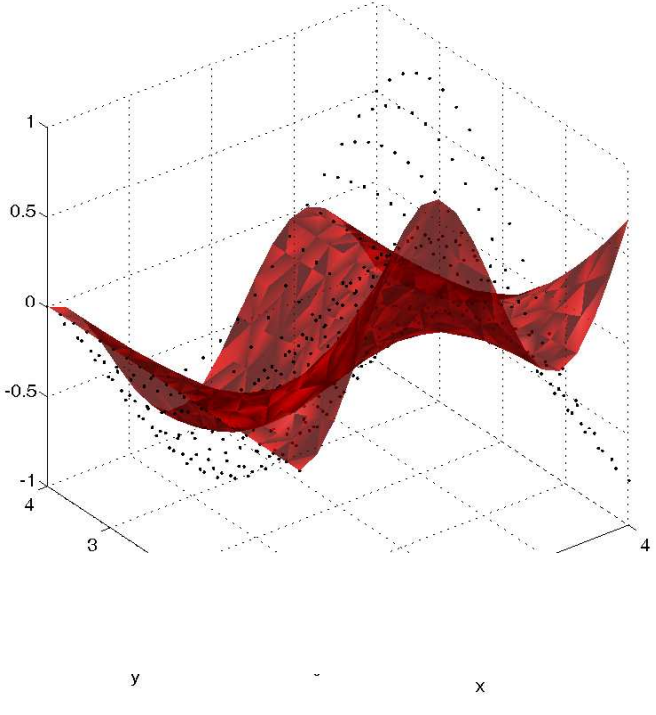




Competitive models

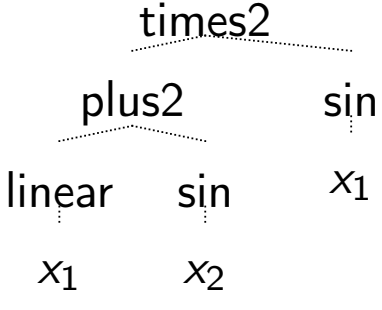
$$\text{times2}(\emptyset,\text{times2}(\emptyset,\text{plus2}(\emptyset,\text{parabola}(w_{1:3},x_2),x_2),\sin(\emptyset,x_1)),\sin(\emptyset,x_1))$$

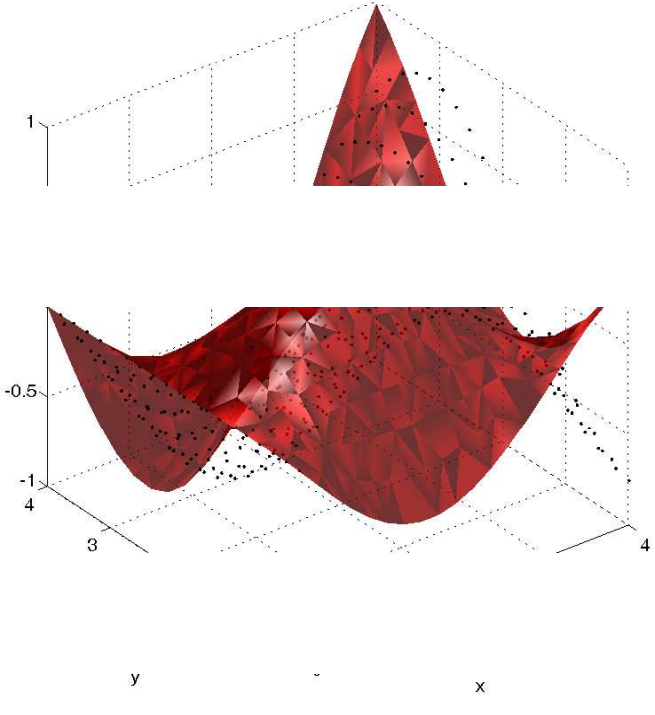




Competitive models

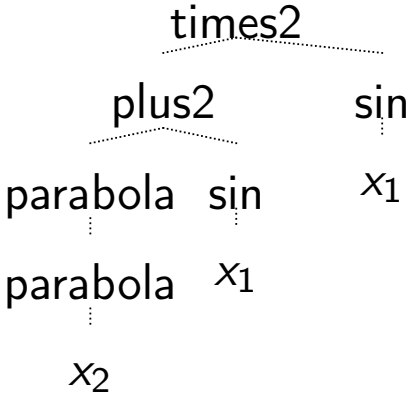
$\text{times2}(\emptyset,\text{plus2}(\emptyset,\text{linear}(w_{1:2},x_1),\sin(\emptyset,x_2)),\sin(\emptyset,x_1))$

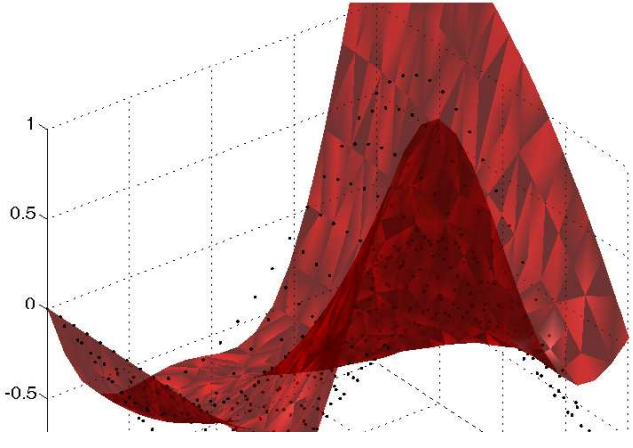




Competitive models

$$\text{times2}(\emptyset,\text{plus2}(\emptyset,\text{parabola}(w_{1:3},\text{parabola}(w_{4:6},x_2)),\sin(\emptyset,x_1)),\sin(\emptyset,x_1))$$

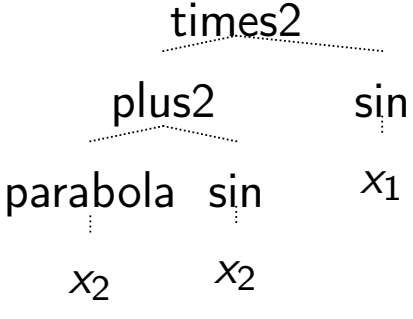


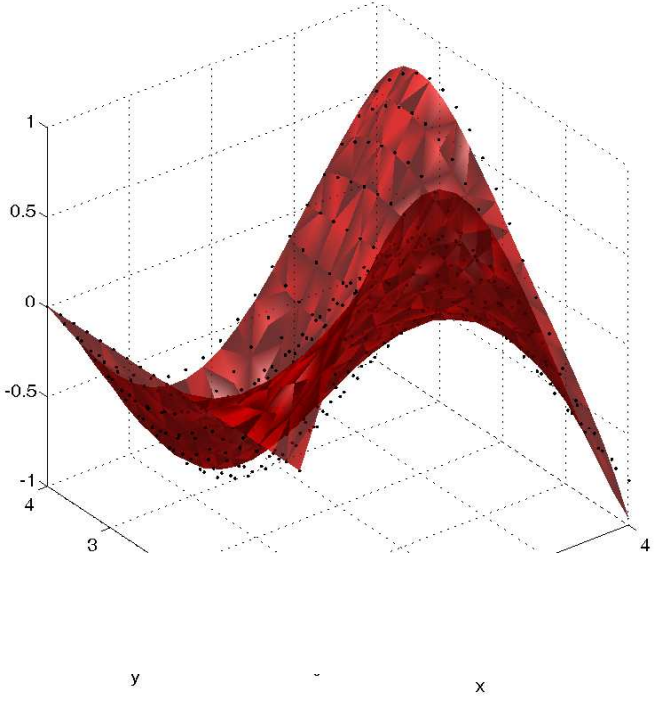


y x

Competitive models

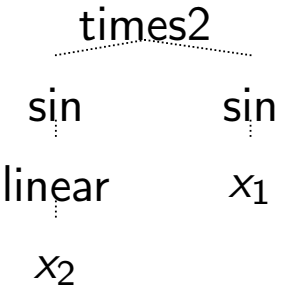
$$\text{times2}(\emptyset,\text{plus2}(\emptyset,\text{parabola}(w_{1:3},x_2),\sin(\emptyset,x_2)),\sin(\emptyset,x_1))$$

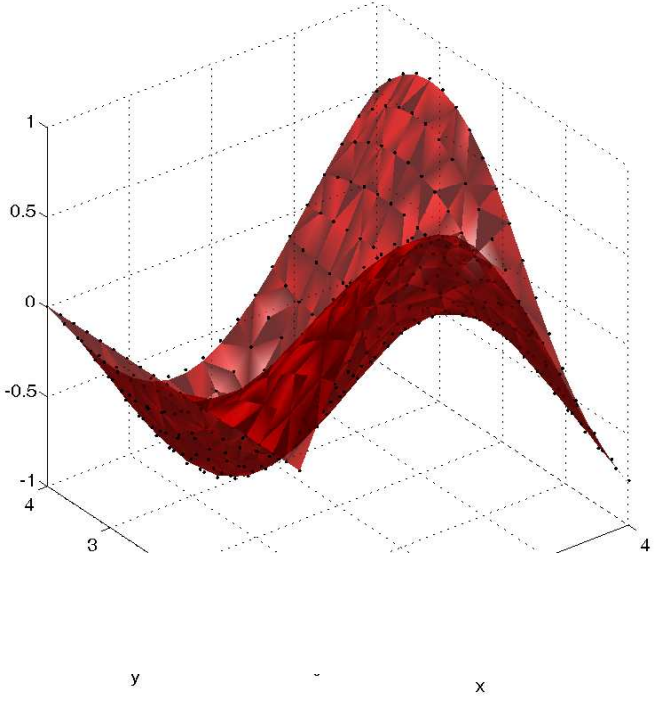


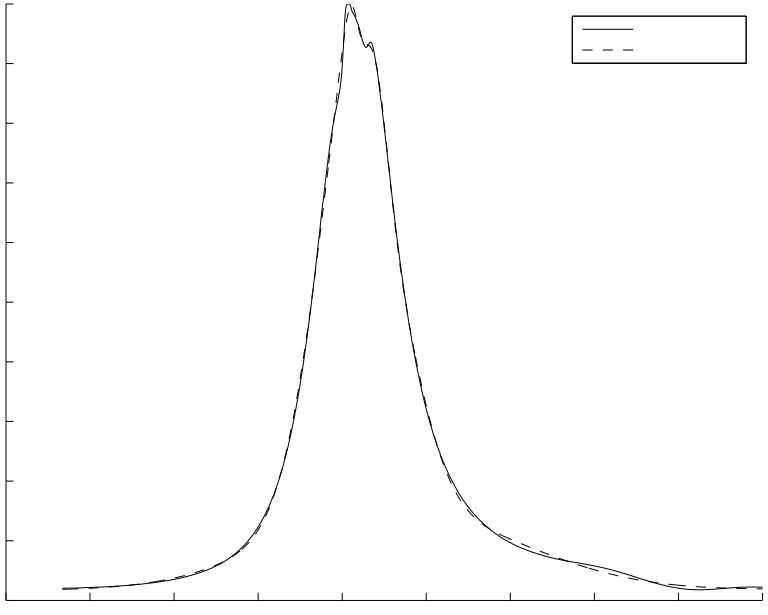


Competitive models

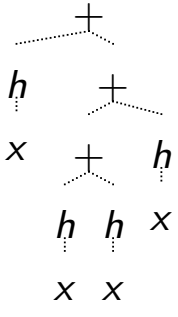
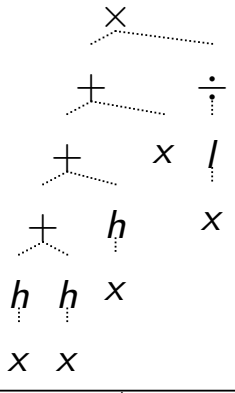
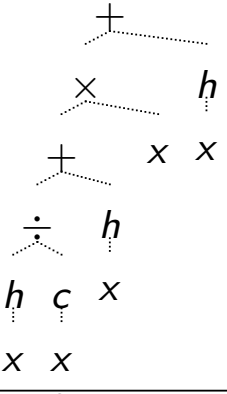
$\text{times2}(\emptyset,\sin(\emptyset,\text{linear}(w_{1:2},x_2)),\sin(\emptyset,x_1))$







The selected models

Model 1	Model 2	Model 3
		

Legend: h — gaussian $y = \lambda(2\pi\sigma^{-1/2})\exp(-(x-\xi)^2(2\sigma^{-2}) + a)$,
c — cubic $y = ax^3 + bx^2 + cx + d$, l — linear $y = ax + b$.

$$f_2 = g_1(g_2(g_3(g_4(g_5(x), g_6(x)), g_7(x)), x), g_8(x)).$$

The full representation of the Model 2 is

$$y = (ax + b)^{-1}\left(x + \sum_{i=1}^{3} \frac{\lambda_i}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(x-\xi_i)^2}{2\sigma_i^2}\right) + a_i\right).$$

Задача

Цель

Предложить автоматическую процедуру порождения ранжирующих функций произвольной структуры, имеющих высокий MAP.

Проблема

Переборный алгоритм порождает только очень простые функции. Алгоритмы генетического поиска дают сложные модели, быстро стагнируя в локальные минимумы.

Предлагаемое решение

Замедлять стагнацию с помощью регуляризации. Выводить из стагнации, определяя момент ее начала с помощью структурных метрик на множестве моделей.

Прикладная задача

Значимость

Предлагаемый подход предназначен для улучшения систем информационного поиска, основанных на экспертных оценках релевантности документа запросам.

Коллекции документов

Следуя традициям сообщества ИП, мы ставим своей целью построение ранжирующих функций, дающих высокий MAP на коллекциях TREC.

Актуальность

Постоянное развитие TREC-сообщества, программных пакетов, связанных в т.ч. с ранжирующими функциями (напр. Terrier) демонстрирует актуальность поставленной задачи.

Известные подходы к порождению функций

Goswami et al. Exploring the space of ir functions, 2014

Переборный алгоритм вычислительно сложен. Аналиризуются только функции с очень низкой структурной сложностью.

Fan et al. A generic ranking function discovery framework by genetic programming for information retrieval, 2004.

Генетический алгоритм порождения застревает в локальных минимумах, итоговые функции переусложнены.

Billhardt P. et al. Using genetic algorithms to find suboptimal retrieval expert combinations, 2002.

Генетический алгоритм поиска линейной комбинации ранжирующих функций возвращает параметры, существенно зависящие от коллекции.

Постановка задачи

Задача

Коллекция документов C экспертно отранжирована по релевантности запросам Q — $g : \bar{C} \times Q \to \{0, 1\}$. Необходимо аппроксимировать зависимость g некоторой явной функцией.

Постановка задачи

Оптимизационная задача:

$$f^* = \text{argmax}_f \big(\mathcal{F}(f) - \text{R}(f)\big),$$

где R — регуляризатор, функционал качества $\mathcal{F}$ — MAP, а максимизация проводится по всем суперпозициям над порождающими G.

Аргументы функции и критерии качества

Аргументы ранжирующей функции

Переменные в G — признаки пары документа и слова (d, w) :

$$x_w^d = t_d^w \log\left(1 + \frac{l_{avg}}{l_d}\right), \quad y_w = \frac{N_w}{N},$$

где N_w — # документов, содержащих w; t_d^w — число слов w в d, l_d — длина d, l_{avg} — средняя длина документа в коллекции. При этом значение функции f на паре (q, d) определяется:

$$f(q, d) = \sum_{w \in q} f(w, d) = \sum_{w \in q} f(x_w^d, y_w)$$

Критерии качества

Идеальная ранжирующая модель h такова, что для лучших моделей $\{f_i\}$ из (Goswami 2014) выполняется: $\mathcal{F}(h) > \mathcal{F}(f_i)$ — модель h равномерно лучше, чем все эталонные функции $\{f_i\}$.

Алгоритм порождения ранжирующих функций

1. Создаётся начальное множество случайных моделей.
2. Часть моделей скрещиваются — обмениваются случайными подмоделями:
   $f = \sin(x) + \mathbf{\cos(xy)},\ g = \cos(x) + \mathbf{(x+y)} \to$
   $\to f = \sin(x) + \mathbf{(x+y)},\ g = \cos(x) + \mathbf{\cos(xy)}.$
3. Некоторые модели мутируют — часть модели заменяется на случайную функцию:
   $f(x, y) = \sin(x) + \mathbf{\cos(xy)} \to f = \sin(x) + \mathbf{\ln(y)}.$
4. Определяется, попал ли алгоритм в локальный минимум. В случае попадания удаляем часть моделей и добавляем множество случайных.
5. Выбирается некоторое число лучших моделей согласно функционалу качества $\big(\mathcal{F} - \mathrm{R}\big)(f)$. Если требуемая точность достигнута, алгоритм останавливается. Иначе, возврат на 2 шаг.

Определение попадания в локальный минимум

Имеется множество моделей $P = \{f_j\}_{j=1}^{|P|}$ и метрики $\mu(f_i, f_j)$. Значение μ на всем множестве P определяется как:

$$\mu(P) = \frac{\sum_{k<j} \mu_i(f_k, f_j)}{avlen(P)},$$

где средняя длина моделей в множестве P:

$$avlen(P) = \frac{1}{|P|} \sum_{j=1}^{|P|} |f_j|.$$

Попадание в локальный минимум определяется, как опускание значения $\mu(P)$ ниже порога Thresh, который определяется эмпирически.

Три метрики μ_1, μ_2, μ_3

Суперпозиции f_i, f_j представляются в виде помеченных деревьев T_i и T_j.

μ_1

$$\mu_1(f_i, f_j) = |T_i| + |T_j| - 2|T_{ij}|,$$

где T_{ij} — наибольший общий подграф деревьев T_i и T_j.

μ_2

Вторая метрика $\mu_2(f_i, f_j)$ — редакторское расстояние между строковыми представлениями деревьев T_i и T_j.

μ_3

Третья метрика $\mu_3(f_i, f_j)$ — редакторское расстояние между самими деревьями T_i и T_j.

Регуляризаторы

Для контроля структурной сложности модели в функционале качества присутствует регуляризатор. Исследуются три варианта (m — значение MAP(f)):

1. $R_1(f) = p \cdot m \cdot I(|f| < CT)$,
   где CT — пороговая сложность модели, $p \in (0, 1)$.
2. $R_2(f) = p \cdot m \cdot I(|f| \geq CT) \cdot (|f| - CT)$,
   где $C > 0$ — некоторая константа.
3. $R_3(f) = p \cdot m \cdot |f|^* \cdot \log(|f| + 1)$,
   где $|f|^*$ — количество переменных x, y в модели.

Вычислительный эксперимент

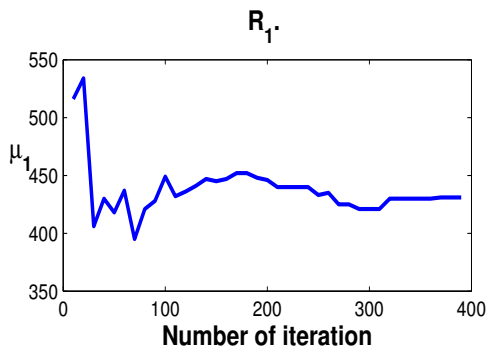
Выборки

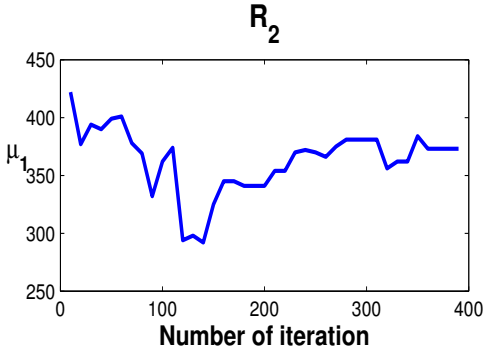
Используются выборки Trec5-8 для анализа моделей, генерируемых генетическим алгоритмом. При этом Trec7 — обучающая, а остальные — контрольные. Все выборки имеют примерно по 500 000 документов и 50 запросов к ним.

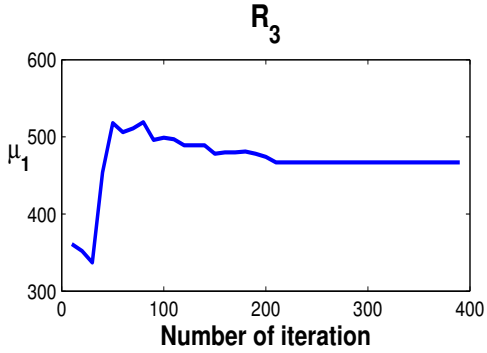
Вычислительный эксперимент

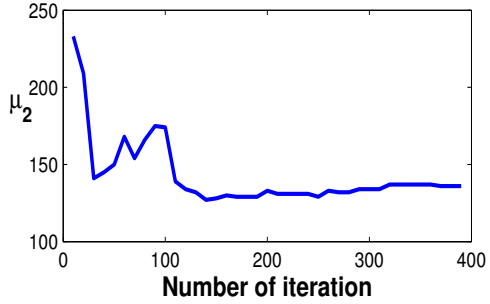
Запускается алгоритм при разных регуляризаторах на выборке Trec7. Выбирается регуляризатор, который наиболее отдаляет точку стагнации и при этом не позволяет моделям значительно переусложняться. Выберается метрика, которая наиболее точно определяет начало стагнации.
После этого запускается алгоритм снова и итоговые модели сравниваются с теми, что были отобраны в Goswami, 2014.

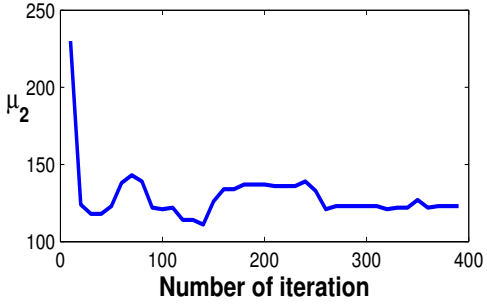
Анализ метрик и регуляризаторов

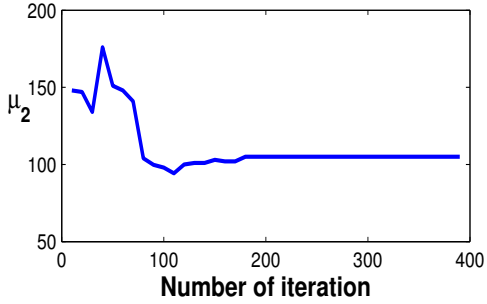


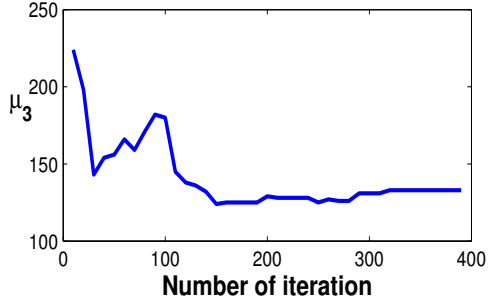


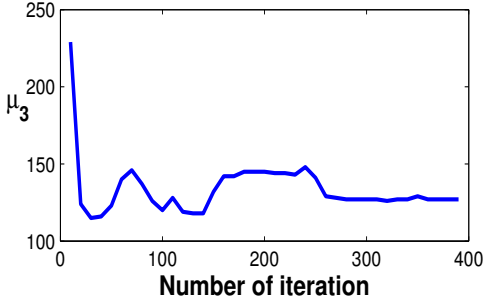


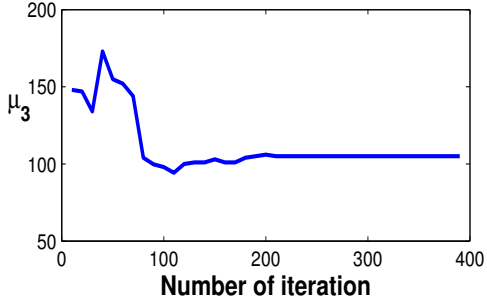












Анализ метрик и регуляризаторов

1. Метрика μ_1 не отражает момент стагнации — на графике нет заметных минимумов и последующих прямолинейных участков.
2. Метрики μ_2 и μ_3 фактически неотличимы по динамике. Выберается μ_2, как более эффективно вычислимое.
3. Регуляризатор R_1 слишком жесткий — алгоритм стагнирует уже после первых итераций.
4. Похожая ситуация наблюдается и для R_2 — метрики существенно уменьшаются уже на первых итерациях.
5. Напротив, с регуляризатором R_3 уменьшение более плавное. Далее используем **третий регуляризатор**.

Итоговые функции на Trec7(μ_2, R_3)

После 600 итераций отобраны функции $\{h_j\}$ ($\ln(x) \to \ln(x+1)$, а $g(x) = \ln\ln(x)$). Модели $\{h_j\}$ сравниваются с функциями $\{f_i\}$, признанными лучшими в работе Goswami,2014. Функции $\{f_i\}$ в среднем на множестве коллекций лучше, чем все модели сложностью не более 8 и известные ранжирующие модели $BM25, LGD, LM_{DIR}$:

#	Эталонные функции	#	Итоговые функции
f_1	$e^{\sqrt{\ln(x/y)}}$	h_1	$g\left(\frac{g(x)}{\sqrt{\ln(x)+x}}\right) - \ln(y)$
f_2	$\sqrt{\frac{\ln(x)}{\sqrt{y}}}$	h_2	$g\left(\frac{g(x)}{\sqrt{\frac{1}{2}\ln(x)+x}}\right) - \ln(y)$
f_3	$\sqrt[4]{\frac{x}{y}}$	h_3	$g\left(\ln\left(\frac{g(x)}{\sqrt{\frac{1}{2}\ln(x)+x}}\right) - \ln(y)\right)$
f_4	$\sqrt{y+\sqrt{\frac{x}{y}}}$	h_4	$g\left(\frac{g(x)}{\sqrt{g(\sqrt{x})+x}}\right) - \ln(y)$
f_5	$\sqrt{\sqrt{\frac{x}{y}}\cdot e^{-y}}$	h_5	$g\left(\frac{g(x)}{\sqrt{\ln(x)+\ln(y)}}\right) - \ln(y)$
f_6	$\sqrt{\sqrt{x}+\sqrt{\frac{x}{y}}}$	h_6	$g\left(\frac{g(\ln(x))}{\sqrt{\ln(x)+x}}\right) - \ln(y)$

Сравнение функций на разных выборках

Значение 50·MAP для функций f_i и h_j				
Функция	Trec5	Trec6	Trec7	Trec8
f_1	8.79	13.71	10.04	13.9
f_2	8.52	13.00	9.22	13.07
f_3	8.91	13.62	9.91	13.71
f_4	8.91	13.62	9.91	13.71
f_5	8.91	13.62	9.91	13.71
f_6	8.87	13.61	9.89	13.70
h_1	8.97	13.69	10.60	14.40
h_2	9.47	13.72	10.65	14.40
h_3	9.56	13.79	10.63	14.38
h_4	9.23	13.71	10.50	14.37
h_5	8.86	13.39	10.44	14.36
h_6	8.10	13.48	10.42	14.36

Выводы

Полученные функции заметно улучшают известные модели из Goswami, 2014. Модели $h_{1,2,3,4}$ лучше их даже на всех тестовых коллекциях.

Хотя структурная сложность моделей h_j примерно в два раза выше, чем у функций f_i, h_j имеют компактный вид.

Структурная сложность f_i и h_j			
Функция	Сложность	Функция	Сложность
f_1	6	h_1	14
f_2	6	h_2	15
f_3	5	h_3	16
f_4	7	h_4	16
f_5	8	h_5	15
f_6	8	h_6	15

Цель работы

Задача

Предложить метод прогнозирования структуры суперпозиции ранжирующей функции, описывающей предъявленную выборку оптимальным образом. Выборка состоит из пар “описание объекта – соответствующее ему оптимальная суперпозиция”.

Исследуемая проблема

Требуется построить ранжирующую модель для коллекции документов. Модель присваивает каждому документу ранг согласно поступающему запросу. Предлагается алгоритм прогнозирования структуры модели, как альтернатива алгоритму перебора суперпозиций элементарных функций.

Метод

Используется метод структурного обучения. Метод восстанавливает скрытую структуру, заданную на исходных данных.

Текущее состояние области

1. Jaakola T., Sontag D. Learning Bayesian Network Structure using LP Relaxations, 2010.
2. Koza, J. R. Genetic programming, 1998.
3. Г.И. Рудой, В.В. Стрижов. Алгоритмы индуктивного порождения суперпозиций для аппроксимации измеряемых данных, 2013.
4. Parantapa Goswami. Exploring the Space of IR Functions, 2014.
5. Clinchant Stephane, Gaussier Eric. Information-based Models for Ad Hoc IR, 2010.

Поиск глобальной модели

- коллекция документов $\mathbb{C} = \{d_i\}_{i=1}^{N}$;
- множество запросов к этой коллекции $\mathbb{Q} = \{q_j\}_{j=1}^{|Q|}$;
- покрытие коллекции $\mathbb{C}$ на подколлекции $C_k, \bigcup C_k = \mathbb{C}$;
- $f_k \in \mathcal{G}$ — оптимальная ранжирующая функция на подколлекции C_k для множества запросов $\mathbb{Q}$:

$$f_k : (C_k \times \mathbb{Q}, f_k) \mapsto \{0, 1\};$$

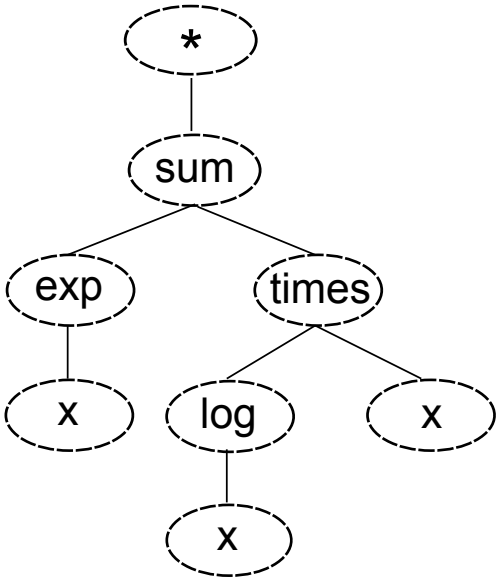
- множество порождающих функций $\mathcal{G}$;
- множество правил порождения суперпозиции $\mathcal{G} = \{g \to B(g, g)|U(g)|S\}$,где $B = \{+, -, *, /\}$, $U = \{sqrt, log, exp\}$, $S = \{x, y\}$

Требуется:

найти глобальную ранжирующую функцию

$$f^* = \arg\max_{f \in \mathcal{G}}(MAP(f, \mathbb{C}, \mathbb{Q})).$$

Правила построения дерева Γ_f суперпозиции f



$\mathcal{G} = \{\exp(\cdot), \log(\cdot), \text{sum}(\cdot,\cdot), \text{times}(\cdot,\cdot)\}$.
$f = \exp(x) + (\log x)x$

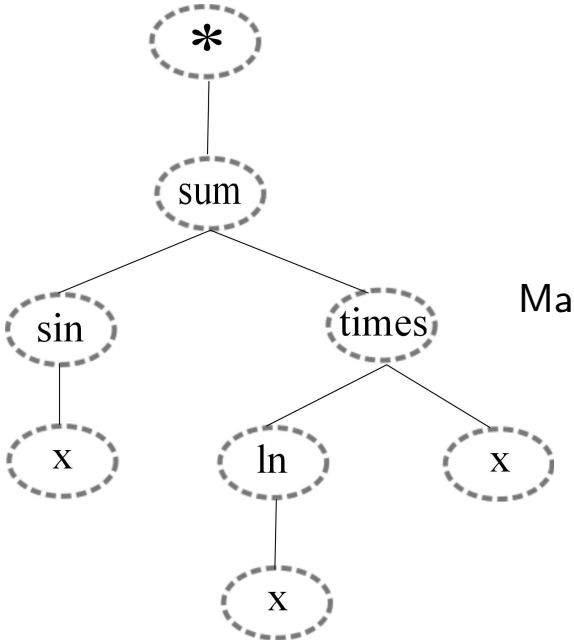
Дерево Γ_f

1. Корень дерева - $*$;
2. $V_i \mapsto g_r$;
3. $\text{val}(V_j) = v(g_{r(i)})$;
4. $\text{dom}(g_{r(i)}) \supset \text{cod}(g_{r(j)})$;
5. аргументы g_r упорядочены;
6. x_i — листья Γ_f.

Правила построения дерева Γ_f суперпозиции f

1. корнем дерева является специальный символ “ * ”, имеющий одну дочернюю вершину;
2. в остальных вершинах V_i дерева Γ_f находятся элементарные функции из набора $\mathcal{G}$;
3. число дочерних вершин V_j у некоторой вершины V_i равно арности соответствующей функции g_r: $v = v(g_r)$;
4. область определения функции дочерней вершины V_j содержит область значений функции родительской вершины V_i: $\mathrm{dom}(g_{r(i)}) \supset \mathrm{cod}(g_{r(j)})$;
5. порядок смежных некоторой вершине V_i вершин соответствует порядку аргументов соответствующей функции $g_r, r = r(i)$;
6. в листьях дерева Γ_f находятся свободные переменные x_i.

Ограничение на построение Z_f



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

Матрица связей Z_f дерева Γ_f

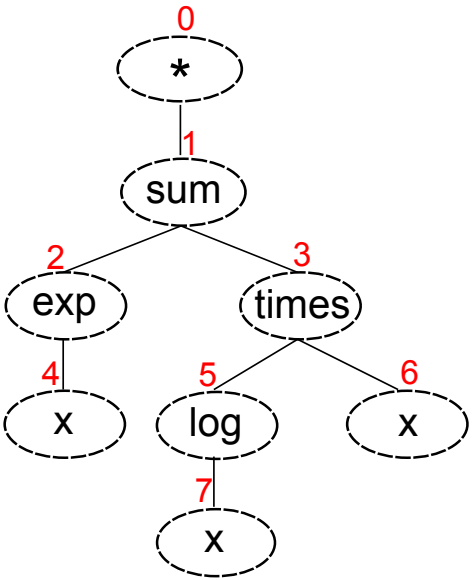
	sum	times	ln	sin	x
$*$	1	0	0	0	0
sum	0	1	1	0	0
times	0	0	0	1	1
ln	0	0	0	0	1
sin	0	0	0	0	1

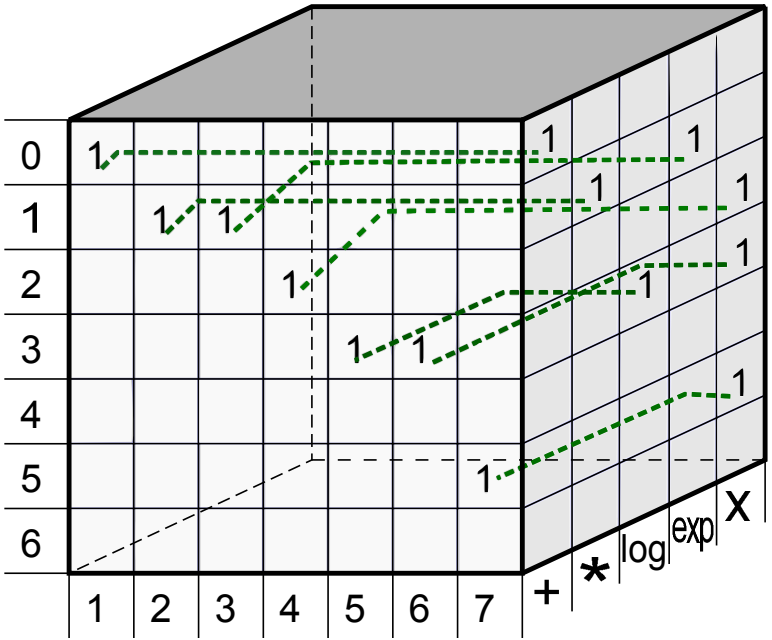
Матрица вероятностей связей P_f дерева Γ_f

	sum	times	ln	sin	x
$*$	0.7	0.1	0.1	0.1	0.2
sum	0.2	0.7	0.8	0.1	0.2
times	0.1	0.3	0	0.8	0.8
ln	0.2	0.1	0.3	0.1	0.9
sin	0.1	0.2	0.1	0	0.8

$\mathcal{M}$ — множество матриц, соответствующих суперпозициям из $\mathcal{F}$.

Ограничение на построение матрицы $\mathbf{Z}_f$ суперпозиции f





$$f = \exp(x) + (\log x)x$$

Трехиндексная матрица связей Z_f дерева Γ_f

- вершины дерева пронумерованы;
- первые два индекса — номера вершин в ребре;
- третий индекс — выбранная элементарная функция на конце ребра.

Прогнозирование структуры суперпозиции

Имеется обучающая выборка состоящая из документов C_k, множества запросов к ним $\mathbb{Q}$, и соответствующих ранжирующих функций f_k: $(C_k \times \mathbb{Q}, f_k)$.
Найти алгоритм прогнозирования

$$a : C_s \times \mathbb{Q} \mapsto f_s.$$

Этапы алгоритма прогнозирования:

1. найти матрицу вероятностей P_k;
2. найти $Z_{f_s} = \arg \max\limits_{Z \in \mathcal{M}} \sum_{i,j,k} P_{ijk} \times Z_{ijk}$.

Построение дерева $\hat{\Gamma}_f$

Задана матрица P, состоящая из блока $P'_{s\times s\times l}$:
P'_{ijk} — вероятность того, что на конце ребра (i,j) в дереве находится функция g_k
и блока $P''_{s\times s\times n}$:
P''_{ijk} — вероятность того, что на конце ребра (i,j) в дереве находится переменная x_k.
Назовем вершину i открытой, если ее арность не равна нулю, но у нее нет дочерних вершин.

Требуется:

построить матрицу Z_f дерева $\hat{\Gamma}_f$.

Способы построения:

1. жадный алгоритм с сохранением нескольких лучших вариантов;
2. метод динамического программирования.

Модификация жадного алгоритма для построения дерева $\hat{\Gamma}_f$

Задано K — максимально допустимая сложность суперпозиции.
Задано R — поддерживаемое число оптимальных суперпозиций.

- Объявляем корень дерева открытой вершиной.
- Пока количество единиц в матрице не превышает K, повторяем:
  1. сортируем по убыванию $P^r_{i^r jk}$ для каждой открытой i^r;
  2. выбираем R лучших $P^r_{i^r j_l k_l}$, $l = 1, \dots R$;
  3. достраиваем матрицы-кандидаты Z^r_l: $Z^r[i^r, j_l, k_l] = 1$;
  4. добавляем j_l к списку открытых вершин для матрицы Z^r, если $(i^r, j_l, k_l) \in P'$;
- если количество единиц превышает T, используем только независимые переменные: $(j^*, k^*) = \arg\max_k P''_{ijk}$, $(i, j^*, k^*) = 1$ для всех i-открытых.

Метод динамического программирования для построения дерева $\hat{\Gamma}_f$

Дано недостроенное дерево Γ с матрицей Z, и процедура вызывается в одной из его открытых вершин $FindOptTree(j^*, k^*)$;

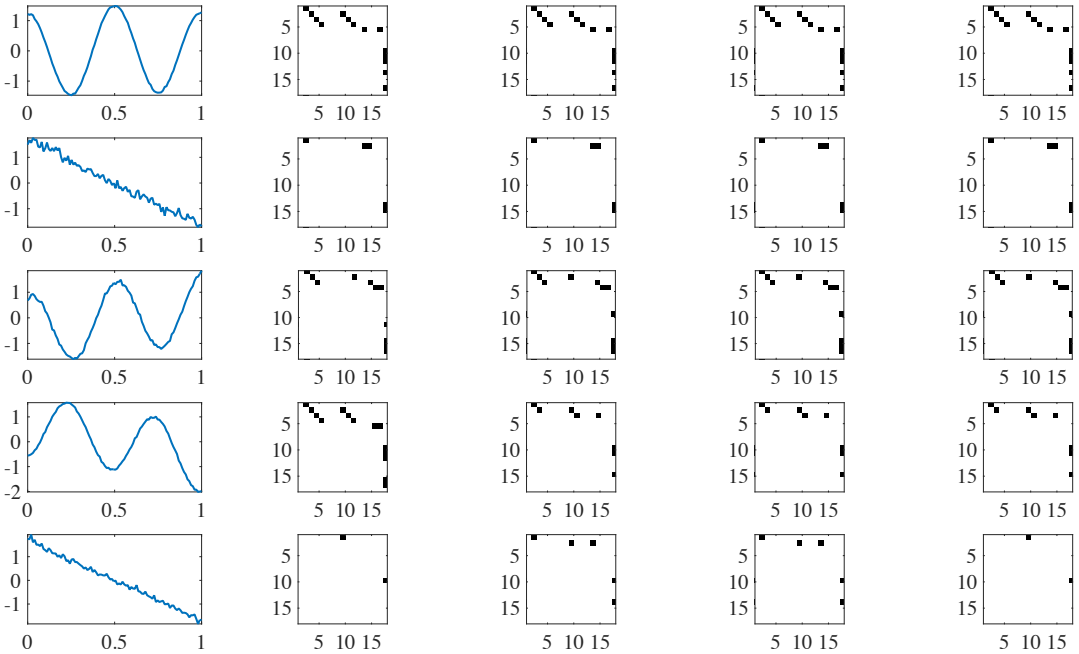
- если k^* — независимая переменная, то процедура завершена, возвращаем $(p_{j^*} = 1, Z = Z)$;
- если k^* — элементарная функция, то:
- для всех возможных переходов из j^* $(j^*, j, k), j = j^* + 1, \ldots, s, k = 1, \ldots, l + n$
  1. считаем $(p_j, Z_j) = FindOptTree(j, k)$;
  2. возвращаем новую вероятность поддерева

$$p_{j^*} = \max_{j=j^*+1,\ldots,s,k=1,\ldots,l+n} P_(j^*, j, k) * p_j$$

  3. достраиваем возвращаемую матрицу $Z = Z_j, \quad Z(j^*, j', k') = 1$, где

$$(j', k') = \arg\max_{j=j^*+1,\ldots,s,k=1,\ldots,l+n} P_(j^*, j, k) * p_j.$$

Тестирование методов прогнозирования Γ_f на синтетических данных



Метод динамического программирования показывает на

Прогнозирование ранжирующей функции на коллекции аннотаций EURO 2010

Цель эксперимента

Зная локальные ранжирующие функции для подколлекций, получить новую ранжирующую функцию для всей коллекции документов, проверить ее качество с помощью функционала MAP.

- 1663 документа в коллекции;
- 24 подколлекции документов;
- 1663 запроса;
- экспертно заданные ранги релевантности;
- множество порождающих функций $\mathcal{G} = \{g \to B(g,g)|U(g)|S\}$,где $B = \{+,-,*,/\}$, $U = \{sqrt, log, exp\}$, $S = \{x, y\}$;
- локальные ранжирующие функции: $\log \frac{x}{y} + \sqrt{y}$, $\sqrt{x} - x$, $\log \frac{x}{y*y}$, $\log \frac{x}{y} + \sqrt{x}$, $\log \sqrt{\sqrt{\frac{x}{y}}}$, $x + y - x^2$, $\log \frac{x}{y} + y$, ...

Прогнозирование ранжирующей функции на коллекции аннотаций EURO 2010

Номер	Символьная запись	MAP
1	$\frac{\log x + \log y}{\exp x}$	0.321
2	$\sqrt{x} - x$	0.308
3	$\sqrt{x} - \exp x + \sqrt{y} - \log(x)$	0.306
4	$\log \frac{x}{y}$	0.303
5	$\log \frac{x}{y*y}$	0.302
6	$x + y - x^2$	0.297
7	$\log \frac{x}{y}$	0.291

Спрогнозированные с помощью структурного обучения функции.

Цель исследования

Цель

Построение алгоритма восстановления матрицы смежности графа суперпозиции по матрице суперпозиции, предсказанной классификатором.

Задача

Восстановление суперпозиции, задающей модель по ее описанию, с помощью взвешенного графа с покрашенными вершинами.

Предложение

Использовать постановку задачи линейного программирования для k-MST, а затем восстановить отброшенные ограничения.

Постановка задачи

Формат описания суперпозиции

Для суперпозиции функций строится граф вычисления, матрица смежности которого и является описанием суперпозиции.

Задача восстановления матрицы суперпозиции

Дана матрица смежности неорентированного взвешенного графа $G = (V, E)$ с покрашенными вершинами и выделенной корневой вершиной r, при этом веса $w(e_i) = w_i \in [0, 1],\ e_i \in E$, а цвета вершин $c(v_i) = c_i \in \mathbb{N}$.

Построить максимальное остовное дерево накрывающее в этом графе как минимум k вершин так, чтобы порядок вершины v_i после накрытия был равен либо 0, либо $c_i + 1$ (для корневой вершины порядок равен 1).

Постановка задачи ($k - MST$, $PCST$)

Rooted $k - MST$ (k-Minimum spanning tree)

Дан неорентированный взвешенный граф $G = (V, E)$ с выделенной корневой вершиной r, при этом веса $w(e_i) = w_i > 0,\ e_i \in E$.
Построить минимальное остовное дерево накрывающее в этом графе как минимум k вершин.

Rooted $PCST$ (Prize-Collecting Steiner Tree)

Дан неорентированный взвешенный (и вершины и ребра имеют веса) граф $G = (V, E)$ с выделенной корневой вершиной r, при этом веса $w(e_i) = w_i > 0,\ e_i \in E$ (стоимости) и $c(v_i) = c_i > 0,\ v_i \in V$ (призы).
Построить дерево T максимизирующее функционал прибыли:
$P = \sum\limits_{v_i \in T} c(v_i) - \sum\limits_{e_i \in E(T)} w(e_i)$.

Задача ЛП для $PCST$ ($k - MST$) с релаксированными условиями целочисленности

$$\underset{\substack{x_e, z_S \\ e \in E, S \subseteq V \setminus \{r\}}}{\text{minimize}} \quad \min \sum_{e \in E} c_e x_e + \lambda \left(\sum_{S \subseteq V \setminus \{r\}} |S| z_S - (n - k) \right)$$

$$\text{s.t.} \quad \sum_{e \in \delta(S)} x_e + \sum_{T : T \supseteq S} z_T \geq 1, \ \forall S \subseteq V \setminus \{r\}, \ v \in S,$$

$$x_e \in [0, 1], \ \forall e \in E$$

$$z_S \in [0, 1], \ S \subseteq V \setminus \{r\}$$

В нерелаксированной постановке $x_e \in \{0, 1\}$, и $x_e = 1$ означает что соответствующее ребро взято в дерево F. Аналогично, $z_S \in \{0, 1\}$, $z_S = 1$ для множества $S = V \setminus F$

Варианты алгоритма восстановления

Базовые

- *DFS*
- *BFS*
- Алг. Прима

Основанные на *k-MST*

- $k - MST$ через *PCST*
- $k - MST + DFS$
- $k - MST + BFS$
- $k - MST$ + алг. Прима

Для каждого из них исследовались неориентированный и ориентированный варианты.

Вычислительный эксперимент

Предположения

- Наборы арностей функций имеют биномиальное распределение (много функций малой арности).
- Одна переменная (обобщение на несколько переменных — двумя способами).

Качество восстановления

Необходимо восстановить корректное дерево суперпозиции, так что восстановлением считаем лишь полное совпадение с исходной матрицей.

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

где N — функция зашумления, а R — алгоритм восстановления.

Вычислительный эксперимент

Тестовое множество

- 50 наборов арностей (длиной 5 —20)
- 20 функций для каждого набора
- 5 зашумлений каждой функции
- Шум — равномерно распределенный, в отрезке $[0, 1]$ с шагом 0.02
- Калибровка — линейная в отрезок $[0, 1]$

Цель эксперимента

Сравнить $k - MST$ подходы с остальными

Результаты экспериментов, неориентированный случай

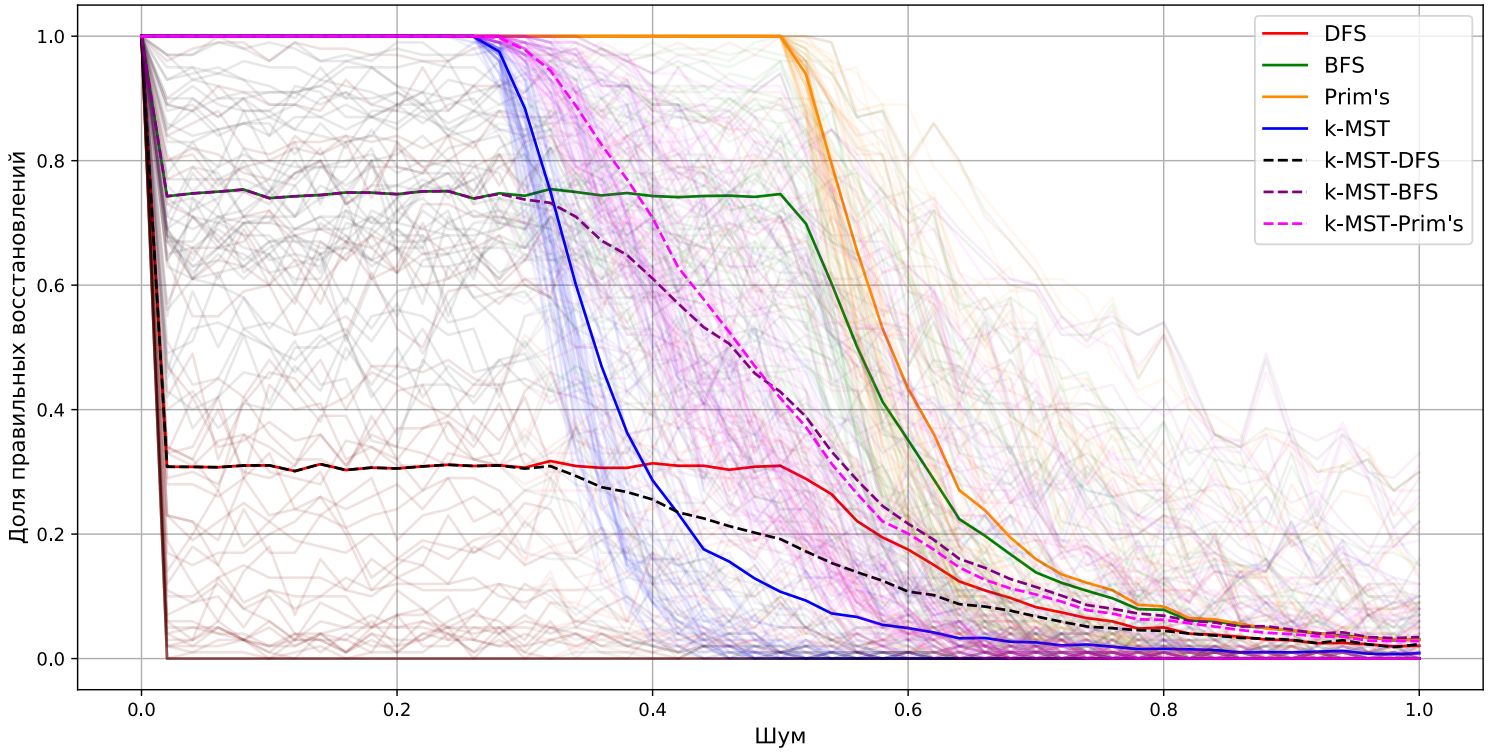


Рис. 1: Зависимость доли правильных восстановлений от силы шума, функции арности $1 - 2$. *k-MST* алгоритмы используют симметризованную матрицу смежности

Результаты экспериментов, неориентированный случай

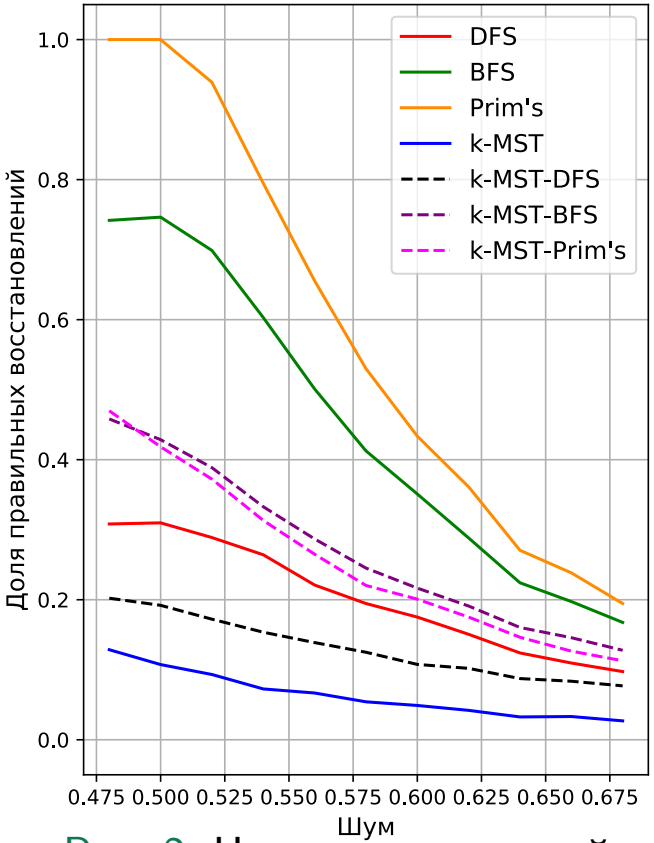


Рис. 2: Неориентированный вариант, качество при шуме ~ 0.5

Качество при шуме ~ 0.5

Шум	.50	.52	.54	.56	.58
DFS	.31	.29	.26	.22	.19
BFS	.75	.70	.60	.50	.41
Прим	**1.0**	**.94**	**.79**	**.66**	**.53**

и для неориентированных вариантов алгоритмов на основе *k-MST*:

Шум	.50	.52	.54	.56	.58
k-MST	.11	.09	.07	.07	.05
$+DFS$	.19	.17	.15	.14	.12
$+BFS$	.43	.39	.33	.29	.25
$+Prim's$	.42	.37	.31	.26	.22

Результаты экспериментов, ориентированный случай

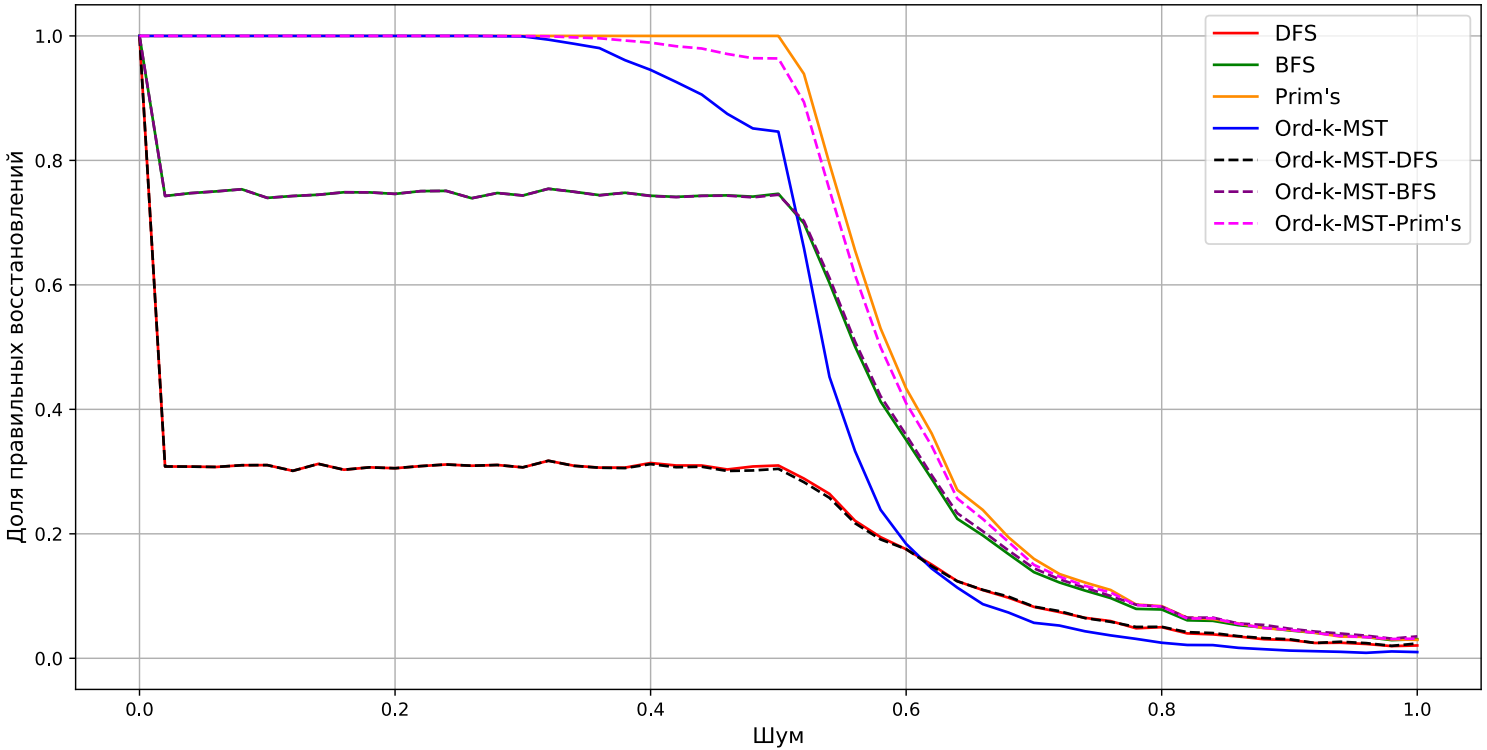


Рис. 3: Зависимость доли правильных восстановлений от силы шума, функции арности $1 - 2$. *k-MST* алгоритмы используют исходную матрицу смежности

Результаты экспериментов, ориентированный случай

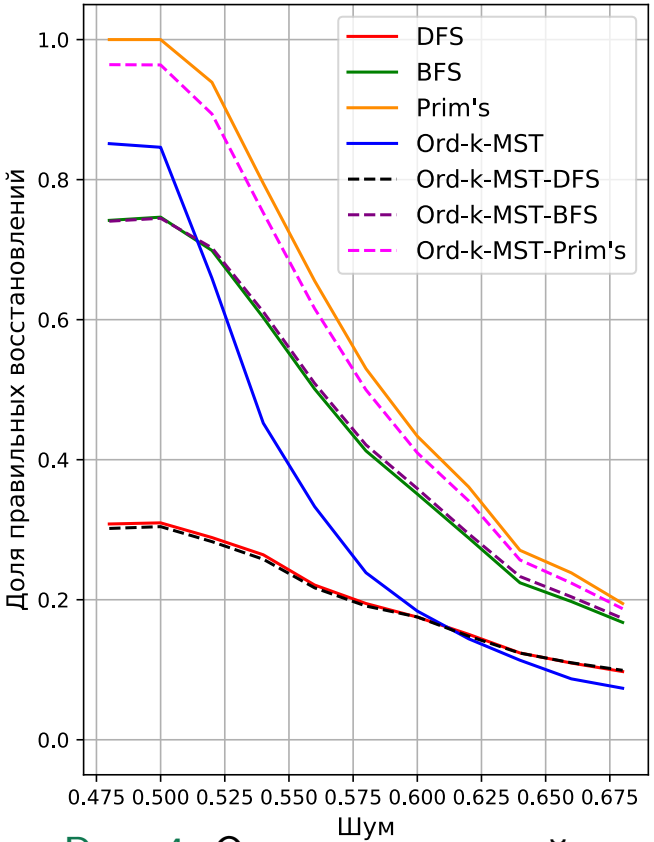


Рис. 4: Ориентированный вариант, качество при шуме ~ 0.5

Качество при шуме ~ 0.5

Шум	.50	.52	.54	.56	.58
DFS	.31	.29	.26	.22	.19
BFS	.75	.70	.60	.50	.41
Прим	**1.0**	**.94**	**.79**	**.66**	**.53**

и для ориентированных вариантов алгоритмов на основе *k-MST*:

Шум	.50	.52	.54	.56	.58
k-MST	.85	.66	.45	.33	.24
$+DFS$	.30	.28	.26	.22	.19
$+BFS$	.74	.70	.61	.51	.42
$+Prim's$	.96	.89	.75	.62	.50

Результаты экспериментов, время работы

Алгоритм	Малые арности время, с.	Большие арности время, с.
DFS	51	48
BFS	53	51
Prim's	85	67
k-MST	182	151
k-MST-DFS	196	162
k-MST-BFS	204	166
k-MST-Prim's	241	187
Ord-k-MST	168	127
Ord-k-MST-DFS	175	131
Ord-k-MST-BFS	171	133
Ord-k-MST-Prim's	195	149

Таблица 1: Полное время работы на датасете (250000 запусков)

Заключение

- Алгоритм Прима работает лучше всех остальных, и полностью восстанавливает при шуме до 0.5
- Ориентированный вариант алгоритма на основе k-MST работает существенно лучше неориентированного
- $k - MST$ препроцессинг существенно повышает качество работы алгоритма зависящего от порядка обхода (BFS)
- Для $k - MST$ параметр λ в $PCST$ надо положить равным 0.5
- При наличии даже малого кол-ва базовых функций больших арностей качество восстановления суперпозиций для любого из алгоритмов снижается.

Цель работы

Исследовать способы автоматического порождения и трансформации регрессионных моделей как суперпозиций экспертно-заданных функций.

Задачи

1. Предложить метод индуктивного порождения суперпозиций, исследовать его свойства.
2. Предложить метод поиска изоморфных структур суперпозиций.
3. Предложить методы трансформации, модификации и упрощения суперпозиций.
4. Применить предложенные методы для автоматического построения экспертно-интерпретируемых моделей.

История проблемы

1. Теорема схем (Дж. Холланд, 1975)
2. Метод группового учета аргументов (А.Г. Ивахненко, 1982)
3. Генетическое программирование (Дж. Коза, 1992)
4. Аналитическое программирование (И. Зелинка, 1994)
5. Усечение моделей (Т. Соул, 2002)
6. Теорема схем для генетического программирования (Р. Поли, Дж. Лангдон, 2003)
7. Использование Байесовского выбора (В.В. Стрижов, 2006)
8. Сети глубокого обучения (Дж. Бенджио, 2009)

Постановка задачи регрессии

Задана выборка

$$\mathfrak{D} = \{x \in \mathbb{X} \subseteq \mathbb{R}^n, y \in \mathbb{Y} \subseteq \mathbb{R}^1\}.$$

Экспертно задано множество G порождающих функций $g(\mathbf{w}, \mathbf{x})$.
Для g определены области аргументов $\mathbb{W} \in \mathbb{R}^m, \mathbb{X} \in \mathbb{R}^n$ и значений $g(w, x) \in \mathbb{R}^1$.
Множество G пополняется функциями $\mathrm{id}(\mathbf{x})$ и const.
Искомая модель $f = (g_1 \circ \ldots \circ g_V)(\mathbf{x})$ является суперпозицией порождающих функций $g_k \in G$,

$$f : \mathbb{X} \times (\mathbb{W}_1 \times \ldots \times \mathbb{W}_v) \to \mathbb{Y}.$$

Требуется породить модель f, минимизирующую значение функции ошибки $S(f)$ при ограничении на структурную сложность $C(f)$:

$$S(f_0) \to \min,$$

$$C(f_0) < C_0.$$

Допустимые суперпозиции

$$f = (g_1 \circ \ldots \circ g_V)(\mathbf{x})$$

Определение

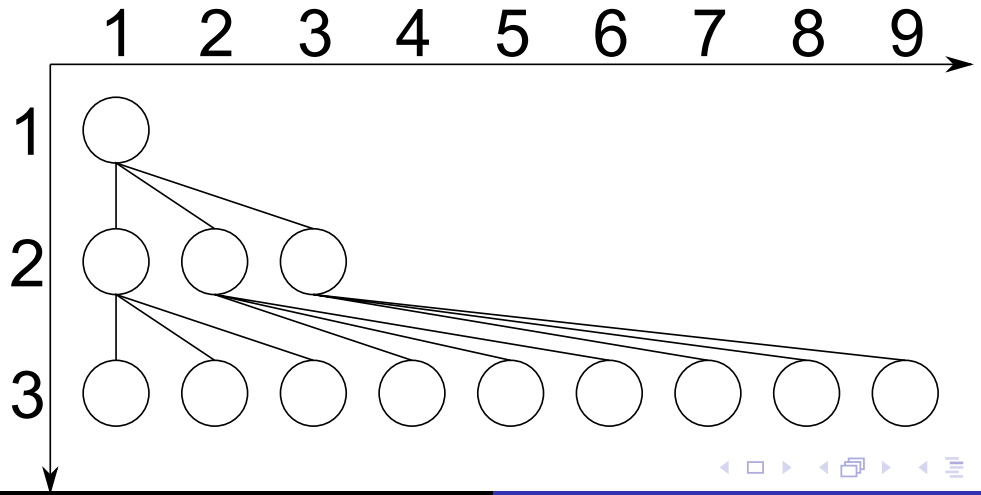
Допустимыми называют суперпозиции, удовлетворяющие следующим требованиям.

- *Элементами s_i суперпозици f могут являться только порождающие функции g_j и свободные переменные x_i.*
- *Количество аргументов элемента суперпозиции s_i равно арности соответствующей ему функции g_j.*
- *Порядок аргументов элемента s_i суперпозиции f соотвествует порядку аргументов соответствующей функции g_j.*
- *Для элемента s_i, аргументом которого является элемент s_j, для соответствующих порождающих функций $dom(g_i) \supseteq cod(g_j)$.*

Представление суперпозиций в виде деревьев

Каждой суперпозиции f однозначно соответствует дерево Γ_f.

- В вершинах v_i дерева Γ_f находятся соответствующие функции g_j.
- Число дочерних вершин у v_i равно арности соответствующей функции g_j.
- Порядок дочерних вершин v_i соотвествует порядку аргументов функции g_j.
- Листьями дерева Γ_f являются свободные переменные x или const.



Расстояние между моделями

Определение

Общее поддерево двух деревьев называется наибольшим, если в нем содержится наибольшее число вершин среди других общих поддеревьев. Наибольшее поддерево деревьев $\Gamma_i(V_i)$ *и* $\Gamma_j(V_j)$ *обозначается как* $\Gamma_{ij}(V_{ij})$.

Рассмотрим абсолютную функцию расстояния r между Γ_i и Γ_j, зависящую от их размеров и наибольшего общего подграфа,

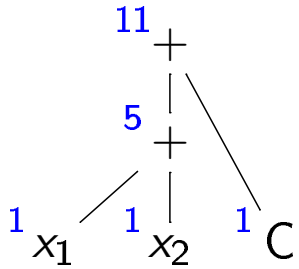
$$r_{ij} = p_i + p_j - 2p_{ij}.$$

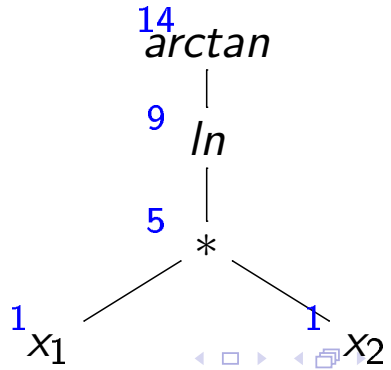
Символом p обозначается количество элементов в множестве V: $|V_i| = p_i$, $|V_j| = p_j$, $|V_{ij}| = p_{ij}$.

Сложность суперпозиции

Сложность суперпозиции f равна сложности корня v_0 и определяется индуктивно с помощью следующей процедуры.

1. Сложность C суперпозиции из одного элемента, соответствующего константной порождающей функции или свободной переменной x_i, равна 1:
$$C(\text{const}) = 1, \qquad C(x_i) = 1.$$
2. Сложность суперпозиции $g_i(f)$, где $C(f) = K$, а g_i — порождающая функция, равна сумме количества элементов в $g_i(f)$ и сложности f:
$$C(g_i(f)) = C(f) + |g_i(f)|.$$





Порождение всех допустимых суперпозиций

Перед первым шагом строится начальное значения множества моделей $\mathcal{F}_0$:

$$\mathcal{F}_0 = \{X, C\}.$$

На каждом шаге для $\mathcal{F}_i$ строится множество U_i из суперпозиций, полученных при применении функций $g_i \in G$ к элементам $f_j \in \mathcal{F}_{i-1}$:

$$U_i = \{g_i(f_{j_1}, \dots f_{j_k}), \mid g_i \in G_u, f \in \mathcal{F}_{i-1}\}.$$

Тогда множество

$$\mathcal{F}_i = \mathcal{F}_{i-1} \cup U_i.$$

Теорема, Сологуб

При данном способе порождения будут построены все допустимые суперпозиции с конечным числом элементов.

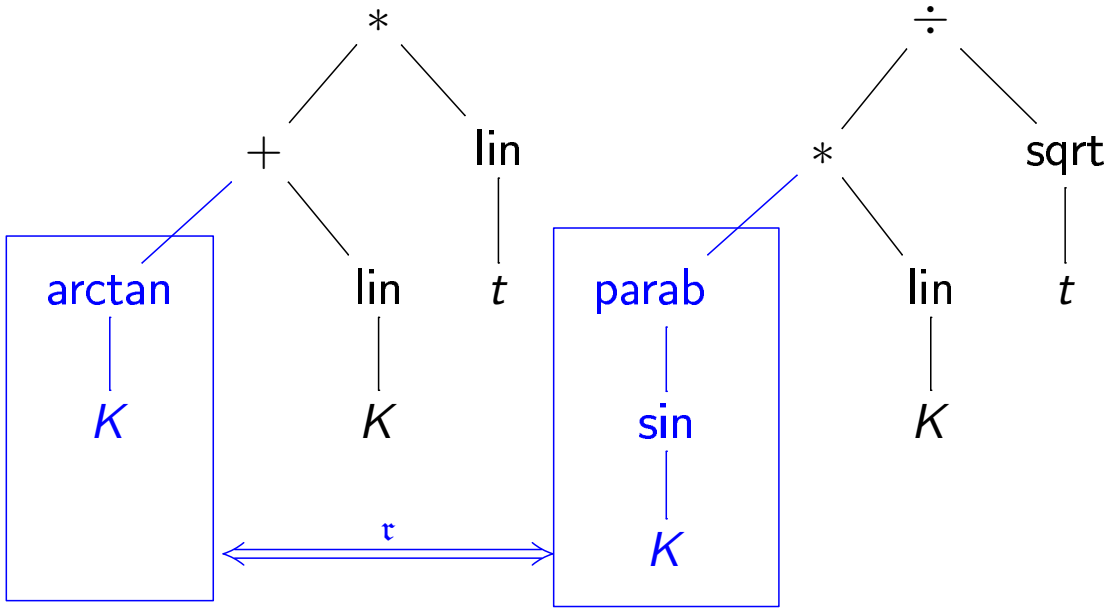
Теорема, Сологуб

Количество суперпозиций, порожденных после k-ой итерации:

$$|\mathcal{F}_k| = \mathcal{O}(l_p^{(p^k-1)/(p-1)} n^{p^k}).$$

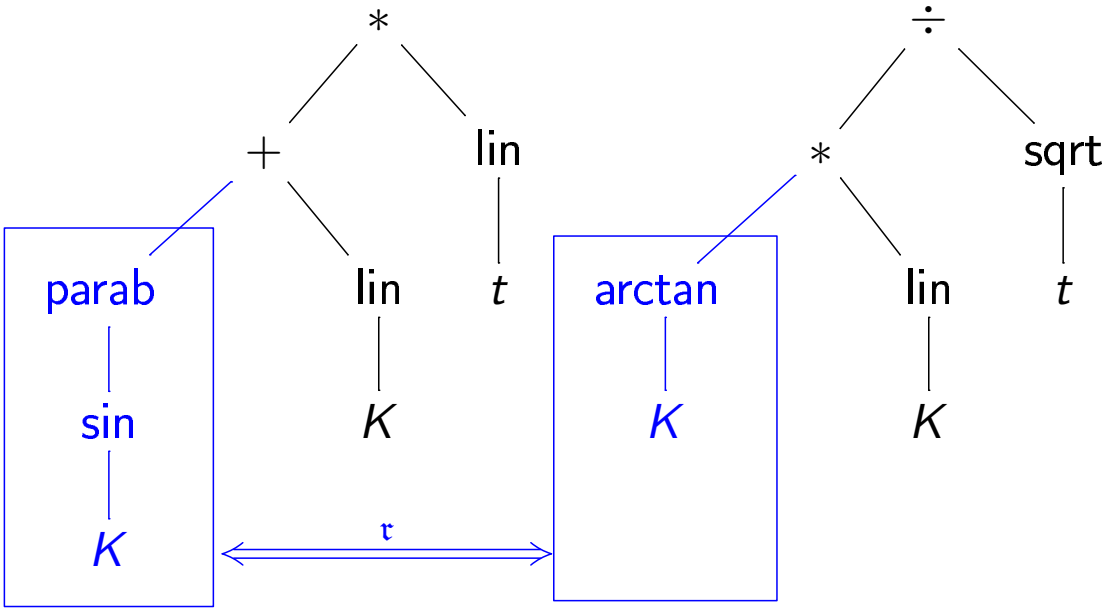
Последовательное порождение суперпозиций

1. Выполняется замена поддерева:
   1. случайно выбирается пара суперпозиций f_i и f_j,
   2. в деревьях Γ_i и Γ_j выбираются вершины v_i и v_j,
   3. порождаются новые модели f_i' и f_j' путем обмена поддеревьями с корнями в выбранных вершинах.



Последовательное порождение суперпозиций

1. Выполняется замена поддерева:
   1. случайно выбирается пара суперпозиций f_i и f_j,
   2. в деревьях Γ_i и Γ_j выбираются вершины v_i и v_j,
   3. порождаются новые модели f_i' и f_j' путем обмена поддеревьями с корнями в выбранных вершинах.



Последовательное порождение суперпозиций

2. Выполняется модификация вершины суперпозиции:
   1. в дереве Γ_j выбирается вершина v_k,
   2. из элементов G, имеющих число аргументов, как у g_k, выбирается g_s,
   3. порождающая функция g_k заменяется функцией g_s.

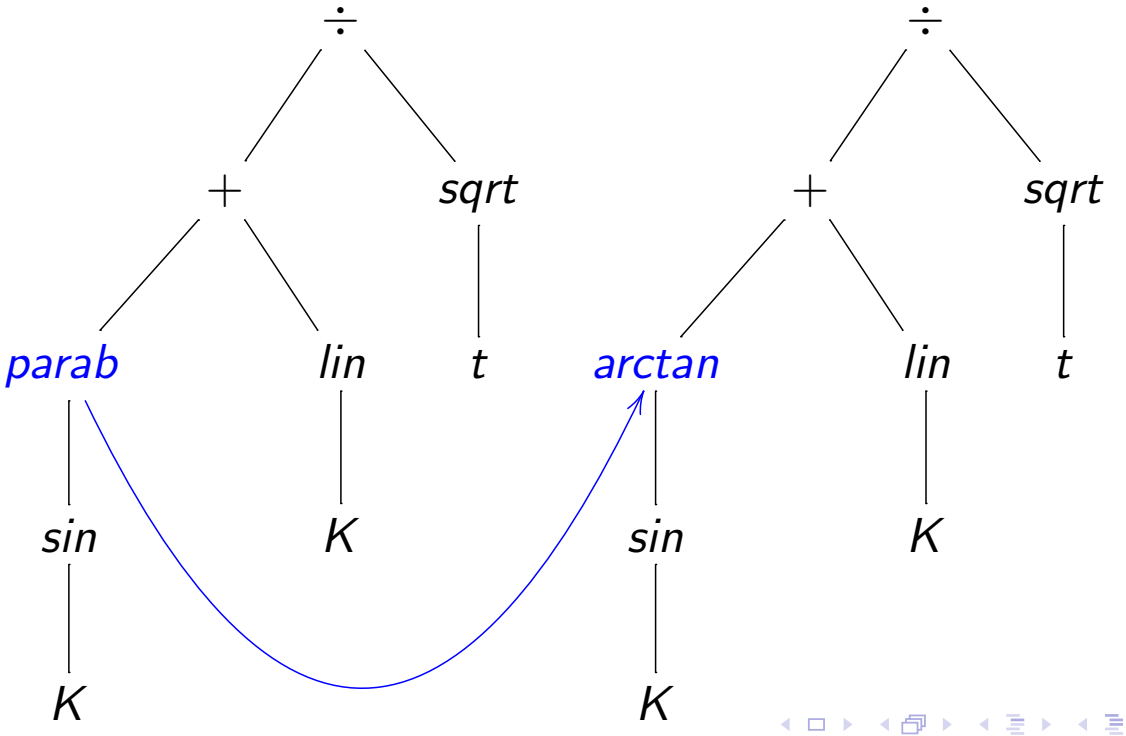


Схема и гиперсхема для графового представления

Определение

Схемой называется дерево, содержащее функции из множества $G \bigcup \{*\}$ *и свободные переменные из множества* $T \bigcup \{*\}$, *где* G *и* T — *множества порождающих функций и свободных переменных соответственно. Порождающая функция* $*$ *означает произвольный символ, который может быть любой функцией или свободной переменной. Порядком схемы* $O(H)$ *называется количество вершин в ней, не являющихся* $*$.

Определение

Гиперсхемой называется дерево, содержащее функции из множества $G \bigcup \{*\}$ *и свободные переменные из множества* $T \bigcup \{*, \#\}$. *Порождающая функция* $*$ *определяется также, как и для схемы, а свободная переменная* $\#$ *означает любое допустимое дерево.*

Теорема схем для графового представления

Теорема, Поли

Для операции замены поддерева с сохранением структуры вероятность сохранения схемы H:

$$\alpha(H,t) = (1 - p_{x0})p(H,t) + p_{x0}\alpha_{x0}(H,t),$$

где

$$\alpha_{x0}(H,t) = \sum_{h_1}\sum_{h_2} \frac{p(h_1,t)p(h_2,t)}{NC(h_1,h_2)} \times$$

$$\times \sum_{i \in C(h_1,h_2)} \delta(h_1 \in U(H,i))\delta(h_2 \in L(H,i)),$$

p_{x0} — вероятность замены,

$p(H,t)$ — вероятность выбора вершины из схемы H,

$NC(h_1,h_2)$ — количество вершин с равной структурой родительских вершин,

$L(H,i)$ — гиперсхема, получаемая из H заменой всех вершин от корня до i на $=$, а всех поддеревьев на $\#$,

$U(H,i)$ — гиперсхема, получаемая из H заменой поддеревьев ниже i на $\#$.

Правило замены подграфов

При порождении суперпозиций возникает большое число различных графов, соответствующих одинаковым отображениям. Для упрощения графа вводится понятие правила замены подграфа.

Определение

Подграф L, удаляемый из графа G в алгоритме упрощения, будет называться заменяемым подграфом.

Определение

Создаваемый подграф R, помещаемый в граф G в алгоритме упрощения, будет называться замещающим подграфом.

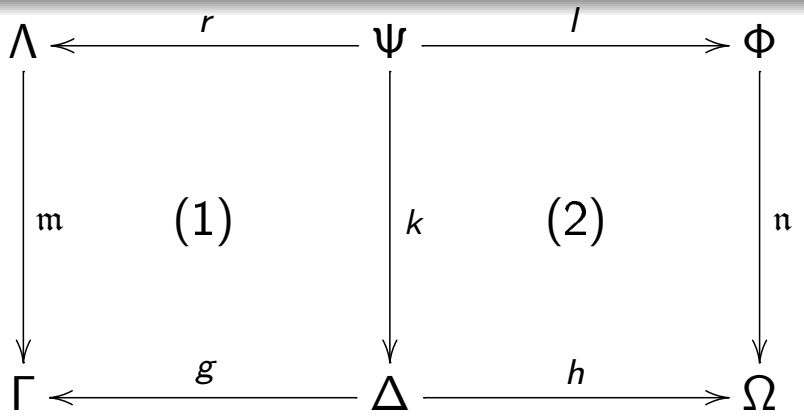
Категория трансформаций графов

Теорема

Трансформации t графов являются категорией на множестве графов Г.

- Классом объектов категории является множество графов Г.
- Стрелками в данной категории являются трансформации t графов.
- Тождественным морфизмом является трансформация графа с пустыми подграфами L и R.
- Ассоциативность операции трансформации графов тривиальна.

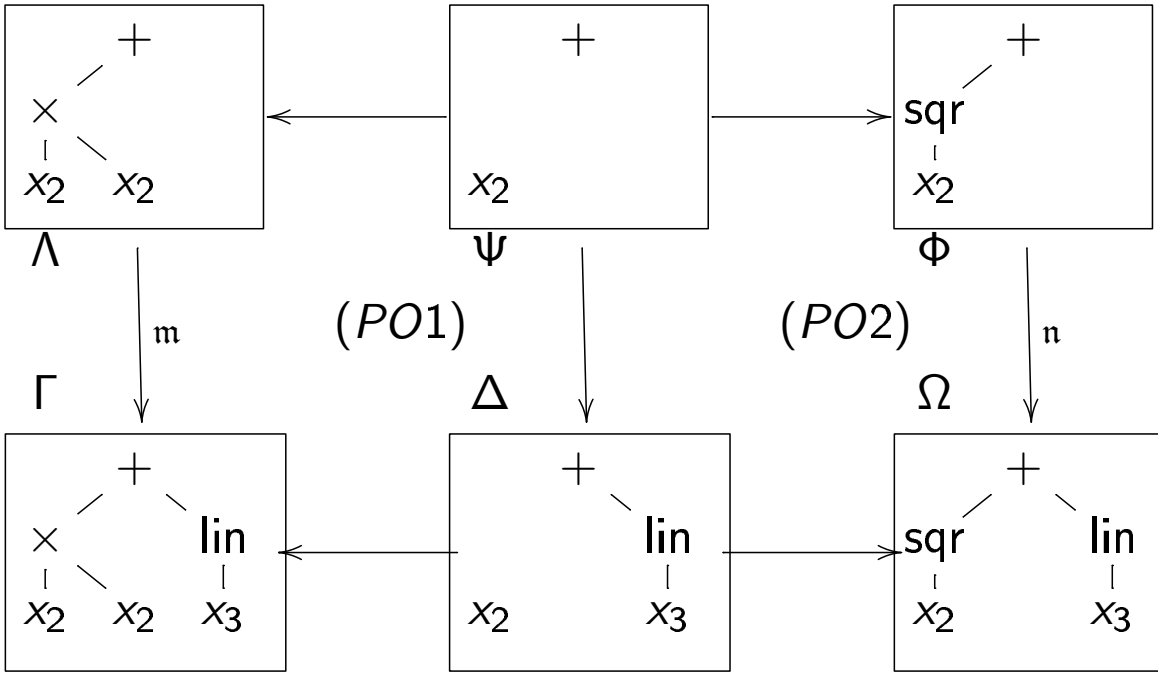
Правило замены подграфов двойным декартовым квадратом



Определение

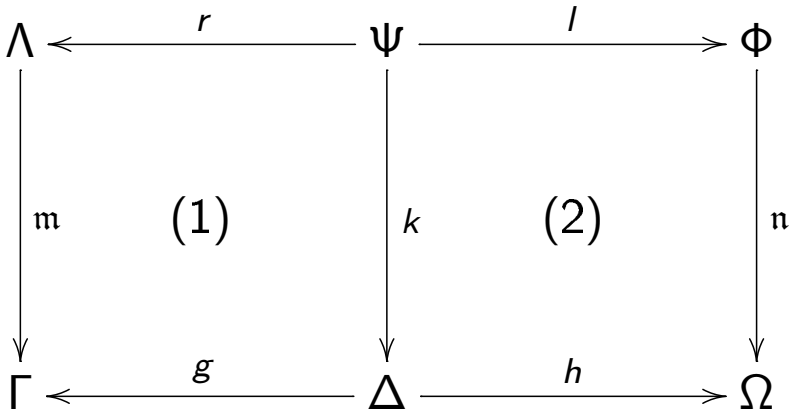
Правило — это тройка $p = (\Lambda, \Psi, \Phi)$, *где* Λ *и* Φ *являются заменяемым и замещающим подграфами. Подграф* Ψ *описывает часть графа, необходимую для применения правила, но неизменную в процессе применения. Множество* $\Lambda \setminus \Psi$ *— удаляемая часть графа, вместо неё создается множество* $\Phi \setminus \Psi$. *Процедура поиска* $\mathfrak{m} : \Lambda \to \Gamma$ *ставит в соответствие заменяемому графу эквивалентный ему подграф.*

Пример трансформации графа



(1)

Условие соединения графов



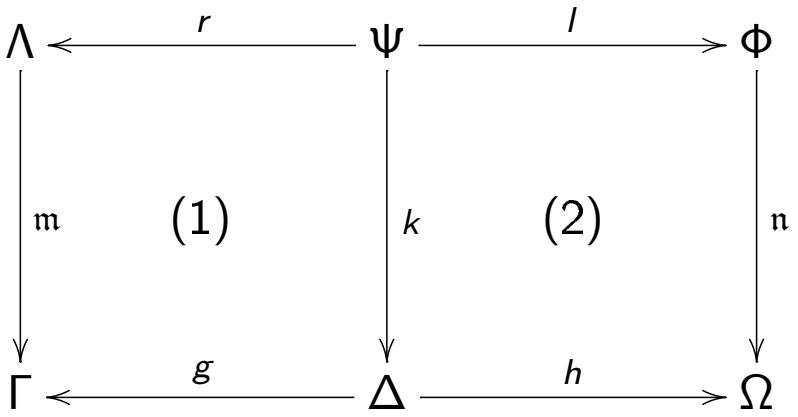
Определение

Точки соединения — вершины и ребра в Λ, которые не удаляются при применении правила p.

Определение

Подвешенные вершины — вершины в Λ, образы которых относительно $\mathfrak{m}$ в Γ имеют входящие или выходящие ребра, не содержащиеся в Λ.

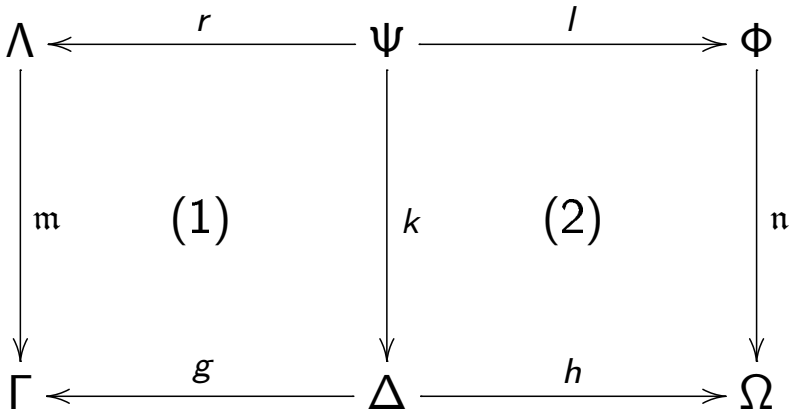
Теорема о применимости правила



Теорема, Эриг

Пусть дано правило $p = (\Lambda \leftarrow \Psi \rightarrow \Phi)$, граф Γ и процедура поиска $\mathfrak{m} : \Lambda \rightarrow \Gamma$. Тогда правило p с процедурой поиска $\mathfrak{m}$ удовлетворяет условию соединения если все подвешенные вершины являются точками соединения.

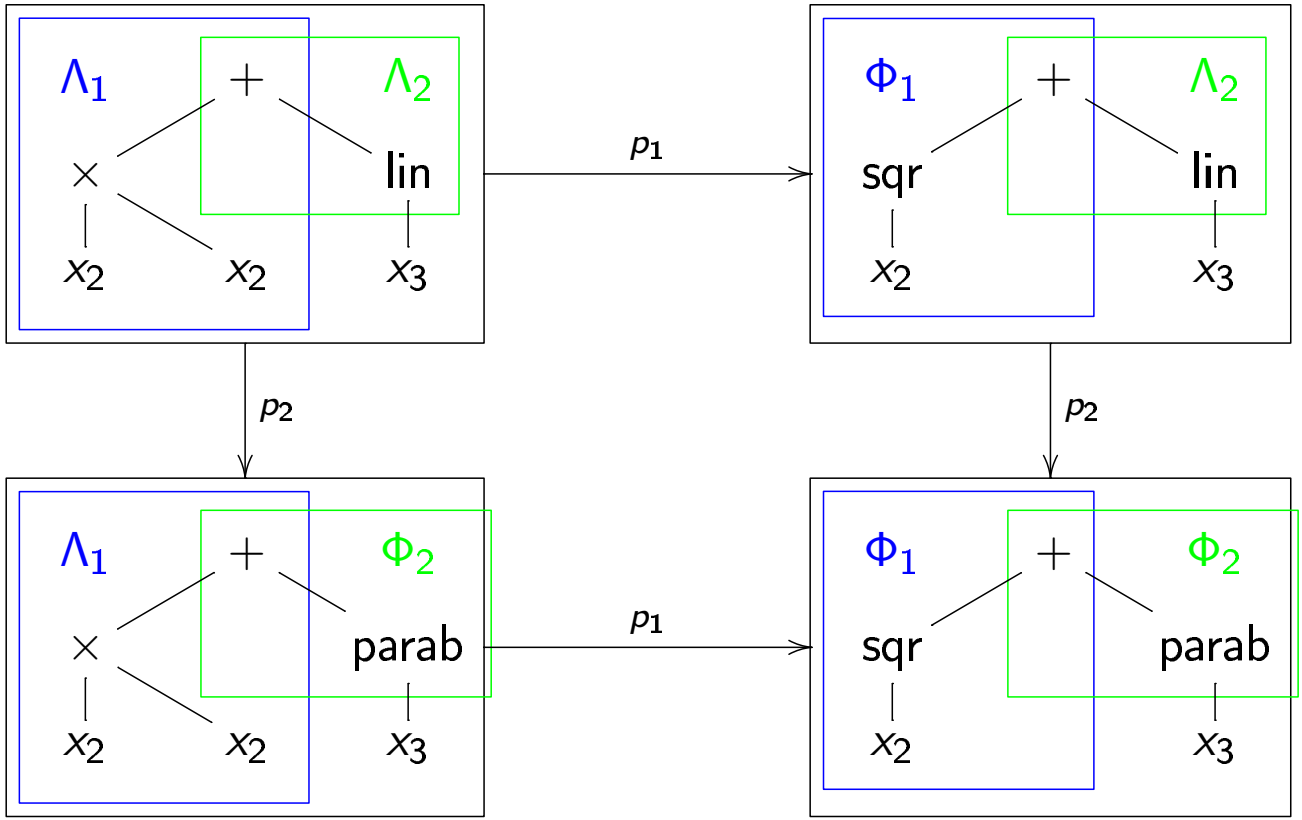
Теорема о построении трансформации



Теорема, Сологуб

Любой трансформации $t = (\Lambda_t, \Psi_t, \Phi_t)$ графа соответствует набор правил $p_t = (\Lambda_{p_t}, \Psi_{p_t}, \Phi_{p_t})$, удовлетворяющих условию соединения, такой что любое применение трансформации t аналогично применению одного из правил p_t.

Параллельная и последовательная независимость трансформаций



Параллельная и последовательная независимость трансформаций

Определение

Две трансформации графов $\Gamma \overset{p_1,m_1}{\to} \Omega_1$ *и* $\Gamma \overset{p_2,m_2}{\to} \Omega_2$ *называются параллельно независимыми, если все вершины и ребра, попадающие в образ обоих морфизмов поиска, являются соединительными:*

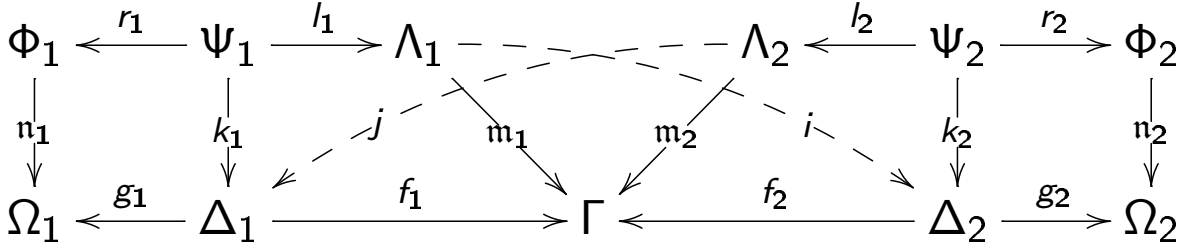
$$\mathfrak{m}_1(\Lambda_1) \bigcap \mathfrak{m}_2(\Lambda_2) \subseteq \mathfrak{m}_1(l_1(\Psi_1)) \bigcap \mathfrak{m}_2(l_1(\Psi_2)).$$

Определение

Две трансформации графов $\Gamma \overset{p_1,m_1}{\to} \Omega_1$ *и* $\Omega_1 \overset{p_2,m_2}{\to} \Omega_2$ *называются последовательно независимыми, если все вершины и ребра, попадающие в пересечение морфизмов* $\mathfrak{n}_1$ *и* m_2*, являются соединительными:*

$$\mathfrak{n}_1(\Phi_1) \bigcap \mathfrak{m}_2(\Lambda_2) \subseteq \mathfrak{n}_1(r_1(\Psi_1)) \bigcap \mathfrak{m}_2(l_2(\Psi_2)).$$

Критерий параллельной независимости



Теорема, Эриг

Две трансформации графов $\Gamma \overset{p_1,m_1}{\to} \Omega_1$ и $\Gamma \overset{p_2,m_2}{\to} \Omega_2$ являются параллельно независимыми, если существуют морфизмы $i : \Lambda_1 \to \Delta_2$ и $j : \Lambda_2 \to \Delta_1$, такие что $f_2 \circ i = \mathfrak{m}_1$ и $f_1 \circ j = \mathfrak{m}_2$.

Теорема, Сологуб

Для графов-деревьев достаточным условием независимости трансформаций $\Gamma \overset{p_1,m_1}{\to} \Omega_1$ и $\Gamma \overset{p_2,m_2}{\to} \Omega_2$ является непопадание корней $v_1 \in \mathfrak{m}_1(\Lambda_1)$ и $v_2 \in \mathfrak{m}_2(\Lambda_2)$ заменяемых поддеревьев в образы морфизмов поиска

$$v_1 \notin \mathfrak{m}_2(\Lambda_2), \qquad v_2 \notin \mathfrak{m}_1(\Lambda_1).$$

Алгоритм упрощения моделей

Определение

Шаблон θ — гиперсхема с наименьшей сложностью среди всех гиперсхем, при взаимном замещении которых получаемые модели эквивалентны. Сложность гиперсхемы определяется как сложность суперпозиции при замещении всех символов $\{*\}$ и $\{\#\}$ на константы.

Экспертно выбирается некоторый набор шаблонов Θ.
Процедура упрощения состоит из двух шагов:

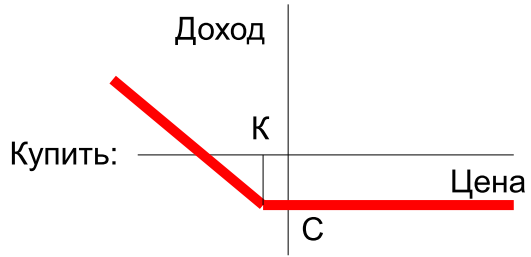
1. Все поддеревья Γ_j в выбранном дереве Γ проверяются на эквивалентность шаблонам из Θ согласно заданным правилам.
2. Если какое-либо поддерево Γ_j в дереве эквивалентно дереву из Θ, данное поддерево заменяется соответствующим элементом из Θ.

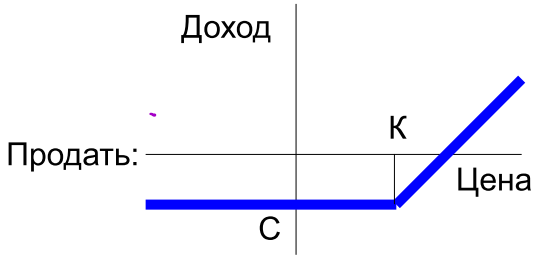
Построение модели волатильности опционов

Опцион — контракт, дающий покупателю право покупки или продажи актива по заданной цене в заданный момент времени.

$$C_t = F(\sigma, P, B, K, t)$$

C_t — цена опциона,
σ — волатильность,
P — цена базового актива,
B — безрисковая ставка,
K — цена исполнения опциона,
t — время до исполнения.





Формула Блэка-Шоулза справедливой цены опциона:

$$C_T = \Phi\left(\frac{ln(\frac{P}{K}) + T(B + \frac{\sigma^2}{2})}{\sigma\sqrt{T}}\right) - Ke^{-BT}\Phi\left(\frac{ln(\frac{P}{K}) + T(B - \frac{\sigma^2}{2})}{\sigma\sqrt{T}}\right).$$

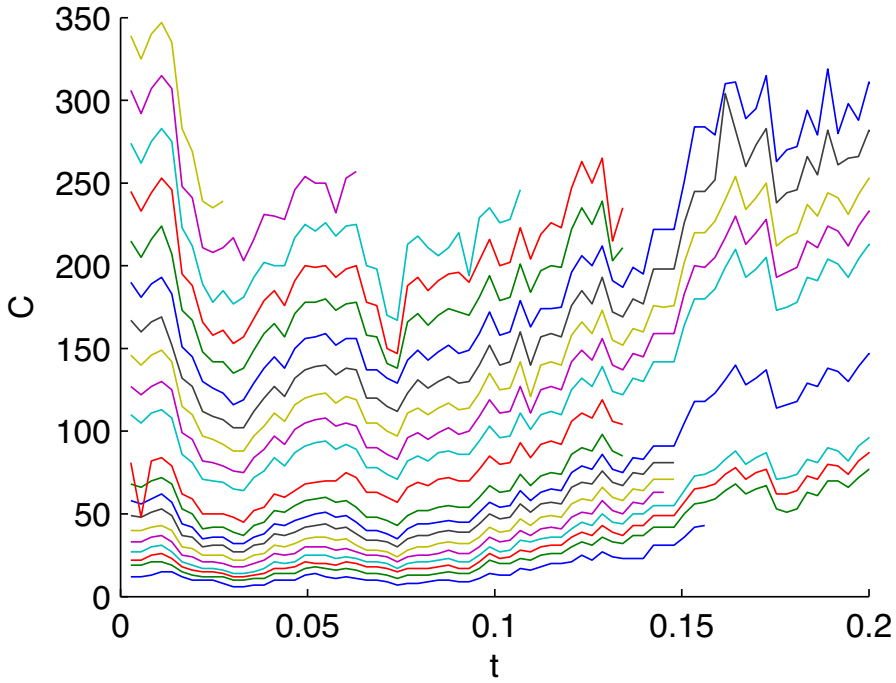
Историческая цена актива

t – время до исполнения опциона в годах,

P — цена актива в указанный период.

Горизонтальные линии соответствуют различным ценам исполнения K.

Исторические цены опционов для всех K



t — время до исполнения опциона в годах,

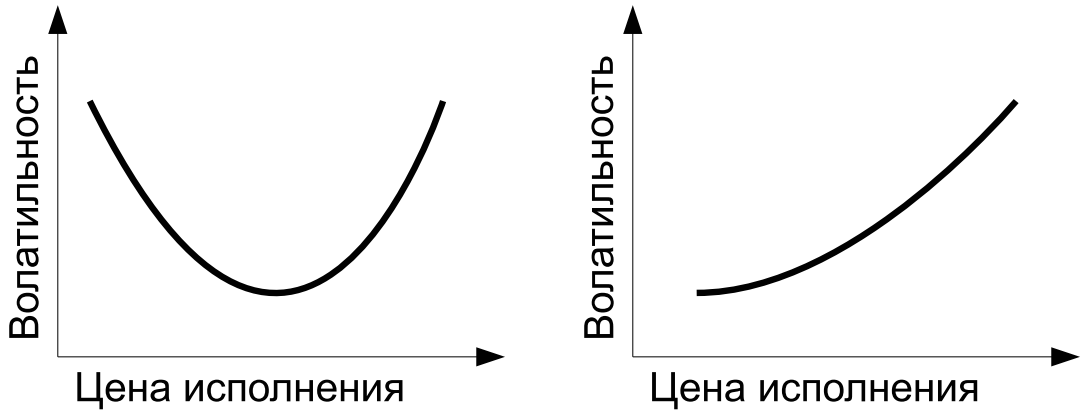
C — цена опциона в указанный период.

Вычисление волатильности опциона

Волатильность — характеристика изменчивости цены базового актива,

$$\sigma^{\text{imp}} = \arg\min_{\sigma}(C_{\text{hist}} - C(\sigma, P, B, K, t)).$$

Предполагаемая волатильность — зависимая переменная искомой регрессионной модели, зависящая от времени t и цены исполнения K.



Модель волатильности, предложенная экспертами

Модель для игроков рынка ММВБ-РТС

$$\sigma = \sigma(\mathbf{w}) = w_1 + w_2(1 - \exp(-w_3 x^2)) + \frac{w_4 \arctan(w_5 x)}{w_5},$$

$$\text{где} \quad x = \frac{\log(K) - \log(C(t))}{\sqrt{t}}.$$

Модельные предположения [Дэглиш, 2006]:

1. волатильность зависит только от цены опциона:

$$\frac{d\sigma}{dP} = \frac{\partial\sigma}{\partial C(P)} \frac{dC(P)}{dP},$$

2. волатильность зависит от времени пропорционально обратному корню

$$\sigma = \Phi\left(\frac{ln(K/F)}{\sqrt{t}}\right).$$

Исторические данные

- В работе использовались данные по ценам на полугодовой опцион на нефть марки Brent Crude Oil, торгующийся на бирже Chicago Board Exchange, с 02.01.2001 по 26.06.2001. Тип опциона — put, символ CLZ01.
- Модель начального приближения предложена экспертами ММВБ-РТС

$$\sigma = \sigma(\mathbf{w}) = w_1 + w_2(1 - \exp(-w_3 x^2)) + \frac{w_4 \arctan(w_5 x)}{w_5},$$

где $x = \frac{\ln K - \ln C(t)}{\sqrt{t}}$.

Моделируемая выборка

Время	K1	K2	K3	K4	K5	K6	**Цена**
-91	0.105	0.16	0.24	0.36	0.56	0.725	**11.27**
-90	0.105	0.16	0.24	0.35	0.56	0.725	**11.29**
-87	0.105	0.16	0.21	0.36	0.56	0.725	**11.34**
-86	0.105	0.16	0.21	0.32	0.56	0.725	**11.2**
-85	0.105	0.16	0.21	0.32	0.48	0.725	**11.18**
-84	0.105	0.16	0.215	0.33	0.625	0.725	**11.5**
...	...	...	...	...	...	...	...

Данные для задачи регрессии формируются следующим образом:

$$\sigma_{t,K} \to \sigma_i, i = \tau + k(|\mathbf{T}| - 1),$$

$$(t_i, K_i) \in \mathbf{T} \times \mathbf{K}.$$

Регрессионная модель:

$$\sigma_i = f(t_i, K_i).$$

Порождающие функции

Функция	Описание	Параметры
	$g(\mathbf{w}, x_1, x_2)$	
plus2	$y = x_1 + x_2$	–
times2	$y = x_1 x_2$	–
frac2	$y = x_1 / x_2$	–
	$g(\mathbf{w}, x)$	
inv	$y = 1/x$	–
add	$y = x + a$	a
normalpdf	$y = \frac{\lambda}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\xi)^2}{2\sigma^2}\right) + a$	λ, σ, ξ, a
linear	$y = ax + b$	a, b
parabolic	$y = ax^2 + bx + c$	a, b, c
sqrt	$y = \sqrt{x}$	–
arctan	$y = \arctan(ax)$	a

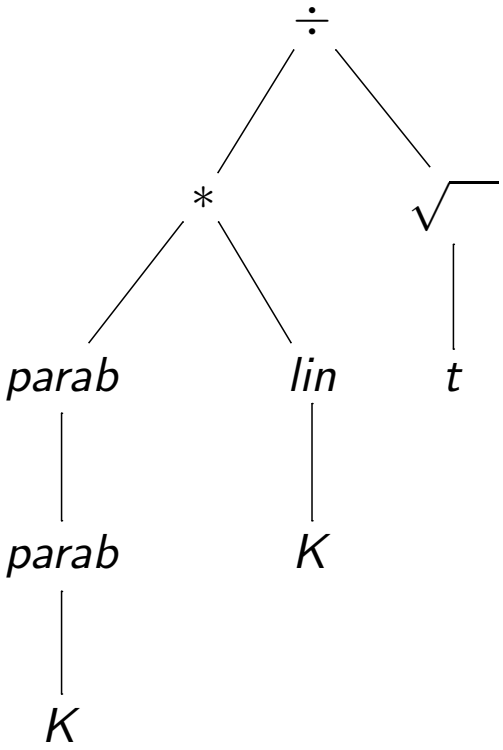
Экспертная интерпретация результатов

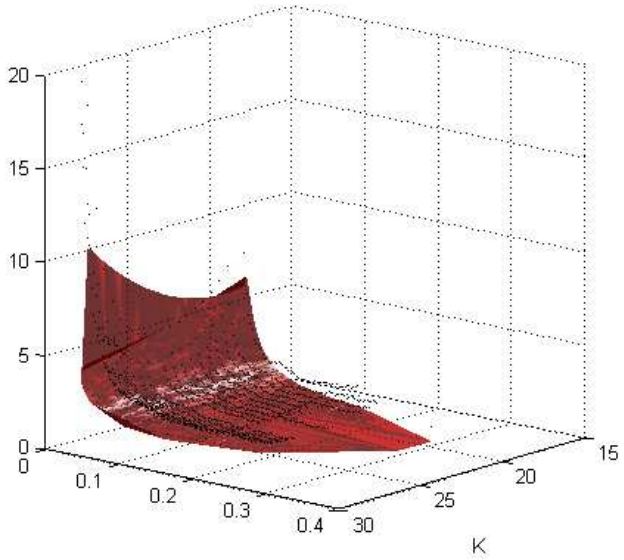
Порождено и трансформировано более 25000 моделей. Для лучших моделей:

- Зависимость волатильности от времени обратно-корневая.
- Зависимость волатильности от цены исполнения полиномиальная.
- Цена опциона как переменная практически не встречается в лучших моделях, в случае её задания как независимой переменной.
- Получена применимая модель волатильности, интерпретируемая экспертами.

Модели волатильности для опциона CLZ

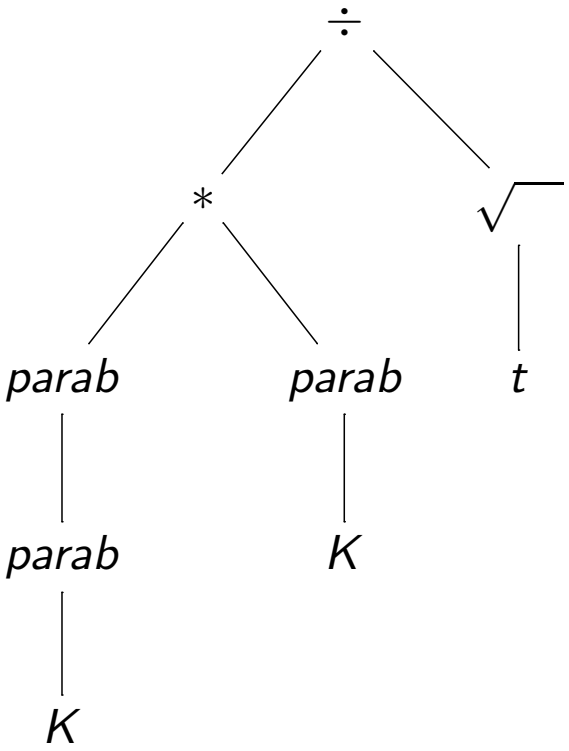
$$\sigma = \frac{(w_1 K + w_2)(w_1 K^2 + w_2 K + w_3)}{\sqrt{t}}$$

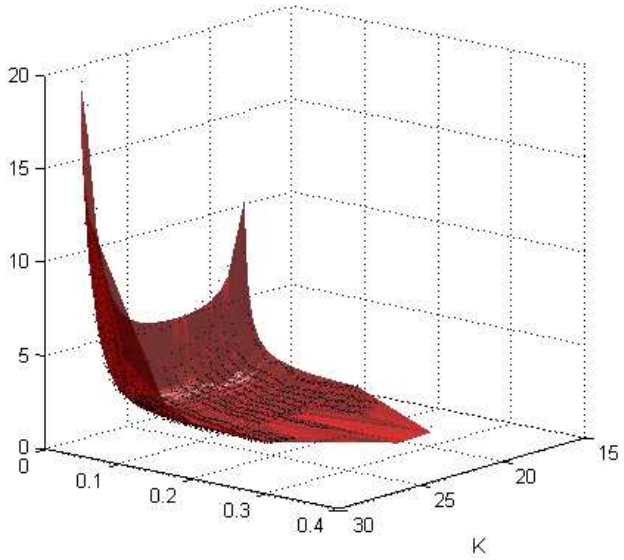




Модели волатильности для опциона CLZ

$$\sigma = \frac{(w_1 K + w_2)(w_1 K^2 + w_2 K + w_3)^2}{\sqrt{t}}$$





Результаты, выносимые на защиту

1. Разработан алгоритм направленного порождения моделей. Разработаны новые алгоритмы вычисления структурной сложности порождаемых суперпозиций и алгоритмы вычисления расстояния между порождаемыми суперпозициями.
2. Разработан метод последовательного направленного порождения суперпозиций, исследованы свойства порождаемых суперпозиций.
3. Введено понятие изоморфных суперпозиций, разработан метод их обнаружения. Разработан алгоритм поиска изоморфных подграфов, соответствующих порожденным суперпозициям.
4. Разработан новый метод порождения экспертно-интерпретируемых моделей. Создана базовая библиотека правил порождения экспертно-интерпретируемых моделей.