

# Мат.модели машинного обучения: методы отбора признаков, оценивания и выбора моделей

Воронцов Константин Вячеславович

`k.v.vorontsov@phystech.edu`

`http://www.MachineLearning.ru/wiki?title=User:Vokov`

Этот курс доступен на странице вики-ресурса

`http://www.MachineLearning.ru/wiki`

«Машинное обучение (курс лекций, К.В.Воронцов)»

## 1 Задача отбора признаков

- Задачи и методы отбора признаков
- Фильтры и встроенные методы
- Методы-обёртки и внешние критерии

## 2 Комбинаторный отбор признаков

- Полный перебор
- Жадные и полужадные алгоритмы
- Поиск в глубину и в ширину

## 3 Эволюционные алгоритмы

- Эволюционные алгоритмы отбора признаков
- Случайный поиск с адаптацией
- Символьная регрессия

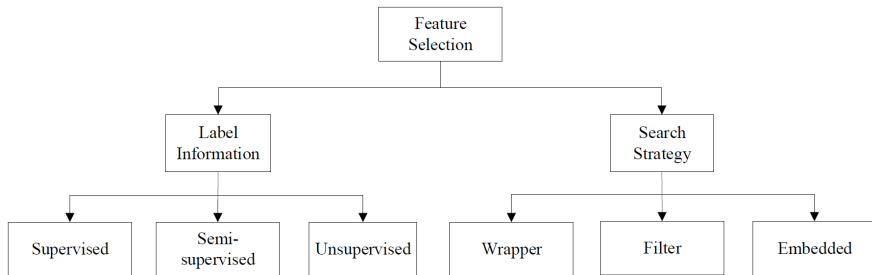
## Почему полезен отбор признаков (feature selection)

$X^\ell = (x_i, y_i)_{i=1}^\ell$  — обучающая выборка,  $y_i = y(x_i)$ ;  
 $f_j(x)$ ,  $j = 1, \dots, n$  — признаки объекта  $x$

### Почему полезно отбирать признаки:

- признак может быть неинформативным:  
 $f_j(x)$  не содержит информации об  $y(x)$
- признак может быть зависимым:  
информация о  $f_j(x)$  содержится в других признаках
- признак может быть настолько зашумлённым, что его использование повышает риск ошибок и переобучения
- минимизация числа используемых признаков может
  - понижать размерность пространства параметров модели,
  - уменьшать риск переобучения,
  - экономить ресурсы времени и памяти

## Обзор подходов к отбору признаков



- с учителем — с использованием ответов  $\{y_i\}$
- без учителя — например, по корреляциям признаков
- обёртки — с использованием готовых методов обучения
- фильтры — оценивание признаков по отдельности
- встроенные методы, в линейных моделях LASSO, ElasticNet

## Фильтрация признаков с учителем и без учителя

### Отбор признаков с учителем:

- вычисление полезности признаков: MI или корреляции  $(y, f_j)$
- ранжирование признаков по их полезности
- отбор top- $k$  полезных признаков по выбранному критерию
- **плюс:** очень быстро
- **минус:** не учитываются зависимости между признаками

### Отбор признаков без учителя:

- вычисление парных корреляций признаков  $(f_j, f_s)$
- выделение кластеров или связанных подграфов
- выбор представителей от каждого кластера  
(метод корреляционных пляд П.В.Терентьева, 1928, 1959)
- **минус:** учитываются только парные зависимости

## Сколько информации об $y$ содержится в $f_j$

$f_j: X \rightarrow V_j$  — дискретный признак,  $V_j$  — конечное множество  
 $p(v)$  — выборочная частотная оценка вероятности  $P(f_j(x) = v)$

Энтропия, мера неопределённости значений  $y$ :

$$H(y) = - \sum_{y \in Y} p(y) \log p(y)$$

Условная энтропия, неопределённость  $y$  при известном  $f_j$ :

$$H(y|f_j) = - \sum_{v \in V_j} p(v) \sum_{y \in Y} p(y|v) \log p(y|v)$$

Взаимная информация (mutual information, information gain):

$$I(y, f_j) = H(y) - H(y|f_j) = \sum_{v \in V_j} \sum_{y \in Y} p(y, v) \log \frac{p(y, v)}{p(y)p(v)}$$

$I(y, f_j) = 0 \Leftrightarrow$  переменные  $y$  и  $f_j$  независимы

## Встроенный отбор признаков в линейных моделях

Негладкие регуляризаторы с параметром селективности  $\mu$ :

$$\sum_{i=1}^{\ell} \mathcal{L}(\langle x_i, w \rangle, y_i) + \tau \sum_{j=1}^n R_{\mu}(w_j) \rightarrow \min_w.$$

**LASSO ( $L_1$ ):**  $R_{\mu}(w) = \mu|w|$

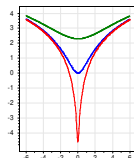
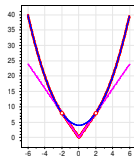
**Elastic Net:**  $R_{\mu}(w) = \mu|w| + w^2$

**Support Features Machine (SFM):**

$$R_{\mu}(w) = \begin{cases} 2\mu|w|, & |w| \leq \mu; \\ \mu^2 + w^2, & |w| \geq \mu; \end{cases}$$

**Relevance Features Machine (RFM):**

$$R_{\mu}(w) = \ln(\mu w^2 + 1)$$



## Задачи оценивания и выбора моделей

### Дано:

$X^\ell = (x_1, \dots, x_\ell)$  — обучающая выборка

$A_t = \{a: X \times W_t \rightarrow Y\}$  — параметрические модели,  $t \in T$

$W_t$  — пространство параметров модели  $A_t$

$\mu_t: (X \times Y)^\ell \rightarrow W_t$  — методы обучения,  $t \in T$

**Найти:** метод  $\mu_t$  с наилучшей *обобщающей способностью*.

### Частные случаи:

- выбор лучшей модели  $A_t$  (model selection);
- выбор метода обучения  $\mu_t$  для заданной модели  $A$  (в частности, оптимизация *гиперпараметров*);
- отбор признаков (feature selection): метод обучения  $\mu_J$  использует только признаки  $J \subseteq F = \{f_1, \dots, f_n\}$
- перебор формул как суперпозиций элементарных функций (symbolic regression, genetic programming)

## Внутренние и внешние критерии качества обучения

$\mathcal{L}(w, x)$  — функция потерь модели  $a(w, x)$  на объекте  $x$

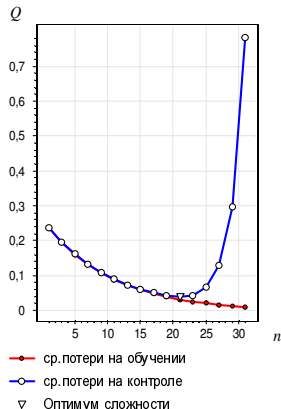
$Q(w, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}(w, x_i)$  — функционал качества  $a(w, x)$  на  $X^\ell$

$Q(\mu(X^\ell), X^\ell)$  — *внутренний критерий*,  
 убывает с ростом сложности модели  
 (числа обучаемых параметров),  
 смещенная оценка  $E_{X^\ell, x} \mathcal{L}(\mu(X^\ell), x)$

$Q(\mu(X^\ell), X^k)$  — *внешний критерий*,  
 имеет минимум по сложности модели,  
 несмещённая оценка  $E_{X^\ell, x} \mathcal{L}(\mu(X^\ell), x)$

**Недостаток** оценки hold-out:

она зависит от разбиения  $X^\ell \sqcup X^k$



## Внешние критерии. Кросс-проверка (cross-validation, CV)

Усреднение по множеству разбиений  $X^L = X_s^\ell \sqcup X_s^k$ ,  $s \in S$ :

$$CV(\mu, X^L) = \frac{1}{|S|} \sum_{s \in S} Q(\mu(X_s^\ell), X_s^k)$$

- $|S| = 1$  — единственное (случайное) разбиение: *hold-out*
- $S$  — множество случайных разбиений: *метод Монте-Карло*
- $S = \{(X^L \setminus \{x_i\}) \sqcup \{x_i\}\}_{i=1..L}$ , каждый объект становится контролем один раз, *скользящий контроль (leave one out)*
- $S = \{(X^L \setminus B_s) \sqcup B_s\}_{s=1..q}$ , где  $B_1 \sqcup \dots \sqcup B_q = X^L$   
 — разбиение *на  $q$  блоков* равной  $\pm 1$  длины ( *$q$ -fold CV*),  
 каждый объект участвует в контроле один раз
- $S = \{(X^L \setminus B_s^r) \sqcup B_s^r\}_{s=1..q, r=1..t}$ , где  $B_1^r \sqcup \dots \sqcup B_q^r = X^L$   
 —  $t$  разбиений *на  $q$  блоков* равной  $\pm 1$  длины ( *$t \times q$ -fold CV*),  
 каждый объект участвует в контроле ровно  $t$  раз
- $S$  — все  $C_{\ell+k}^k$  разбиений: *complete cross-validation, CCV*

## Внешние критерии. Непротиворечивость моделей

**Идея:** Если модель верна, то алгоритмы, настроенные по разным частям данных, не должны противоречить друг другу.

1. По одному случайному разбиению  $X^L \sqcup X^k = X^L$ ,  $\ell = k$ :

$$Q_1(\mu, X^L) = \frac{1}{L} \sum_{i=1}^L |\mu(X^\ell)(x_i) - \mu(X^k)(x_i)|.$$

2. Аналог  $CV_{t \times 2}$ : по  $t$  разбиениям  $X^L = X_s^\ell \sqcup X_s^k$ ,  $s = 1, \dots, t$ :

$$Q_t(\mu, X^L) = \frac{1}{t} \sum_{s=1}^t \frac{1}{L} \sum_{i=1}^L |\mu(X_s^\ell)(x_i) - \mu(X_s^k)(x_i)|.$$

**Преимущество:** подходит также для обучения без учителя.

**Недостатки:**

- объём обучения сокращается в 2 раза;
- трудоёмкость возрастает в  $2t$  раз.

## Внешние критерии. Регуляризация

*Регуляризатор* — аддитивная добавка к внутреннему критерию, обычно штраф за сложность (complexity penalty) модели  $A$ :

$$Q_{\text{рег}}(\mu, X^\ell) = Q_\mu(X^\ell) + \text{штраф}(A),$$

**Линейные модели:**  $A = \{a(x) = \text{sign}\langle w, x \rangle\}$  — классификация,  
 $A = \{a(x) = \langle w, x \rangle\}$  — регрессия.

$L_2$ -регуляризация (ридж-регрессия):

$$\text{штраф}(w) = \tau \|w\|_2^2 = \tau \sum_{j=1}^n w_j^2.$$

$L_1$ -регуляризация (LASSO):

$$\text{штраф}(w) = \tau \|w\|_1 = \tau \sum_{j=1}^n |w_j|.$$

$L_0$ -регуляризация (AIC, BIC):

$$\text{штраф}(w) = \tau \|w\|_0 = \tau \sum_{j=1}^n [w_j \neq 0].$$

## Разновидности $L_0$ -регуляризации

Информационный критерий Акаике (Akaike Information Criterion):

$$\text{AIC}(\mu, x) = Q_\mu(X^\ell) + \frac{2\hat{\sigma}^2}{\ell}|J|,$$

где  $\hat{\sigma}^2$  — оценка дисперсии ошибки  $D(y_i - a(x_i))$ ,

$J$  — подмножество используемых признаков.

Байесовский информационный критерий (Bayes Inform. Criterion):

$$\text{BIC}(\mu, X^\ell) = \frac{\ell}{\hat{\sigma}^2} \left( Q_\mu(X^\ell) + \frac{\hat{\sigma}^2 \ln \ell}{\ell} |J| \right).$$

Оценка Вапника-Червоненкиса (VC-bound):

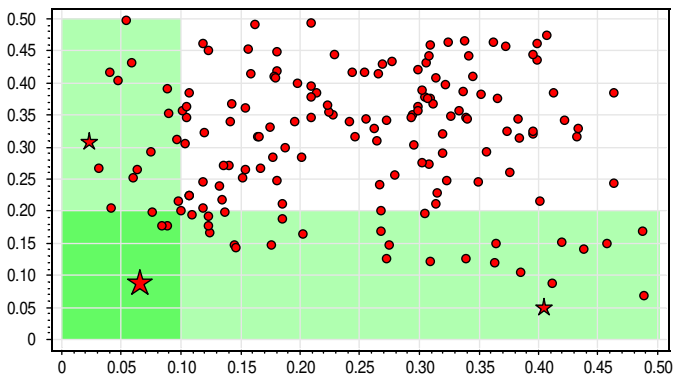
$$\text{VC}(\mu, X^\ell) = Q_\mu(X^\ell) + \sqrt{\frac{h}{\ell} \ln \frac{2e\ell}{h} + \frac{1}{\ell} \ln \frac{9}{4\eta}},$$

$h$  — VC-размерность; для линейных моделей  $h = |J|$ ;

$\eta$  — уровень значимости; обычно  $\eta = 0.05$ .

## Многокритериальный выбор модели

Модель, немного неоптимальная по обоим критериям, может оказаться лучше, чем модель, оптимальная по одному критерию, но сильно не оптимальная по другому.



## Задача отбора признаков по внешнему критерию

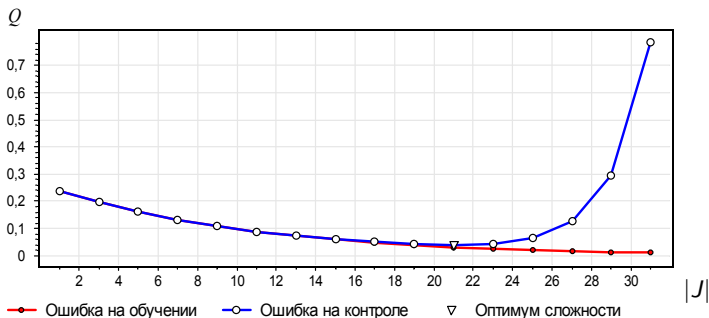
$F = \{f_j: X \rightarrow D_j: j = 1, \dots, n\}$  — множество признаков;

$\mu_J$  — метод обучения, использующий только признаки  $J \subseteq F$ ;

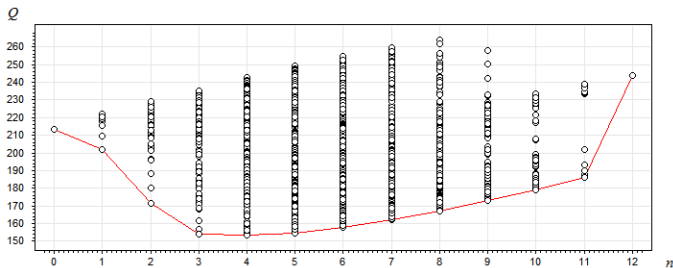
$Q(J) = Q(\mu_J, X^\ell)$  — выбранный внешний критерий.

$Q(J) \rightarrow \min$  — задача дискретной оптимизации.

Внутренний критерий и внешний критерий:



## Алгоритм полного перебора (Full Search)



**Вход:** множество  $F$ , **внешний критерий**  $Q$ , параметр  $d$ ;

инициализация:  $Q^* := Q(\emptyset)$ ;

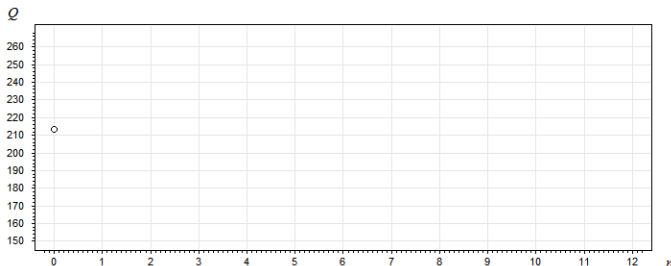
**для**  $j = 1, \dots, n$ , где  $j$  — сложность наборов

$J_j := \arg \min_{J: |J|=j} Q(J)$  — найти лучший набор сложности  $j$ ;

**если**  $Q(J_j) < Q^*$  **то**  $j^* := j$ ;  $Q^* := Q(J_j)$ ;

**если**  $j - j^* \geq d$  **то вернуть**  $J_{j^*}$ ;

## Алгоритм полного перебора (Full Search)



$$d = 3$$

$$j = 0$$

**Вход:** множество  $F$ , **внешний критерий**  $Q$ , параметр  $d$ ;

инициализация:  $Q^* := Q(\emptyset)$ ;

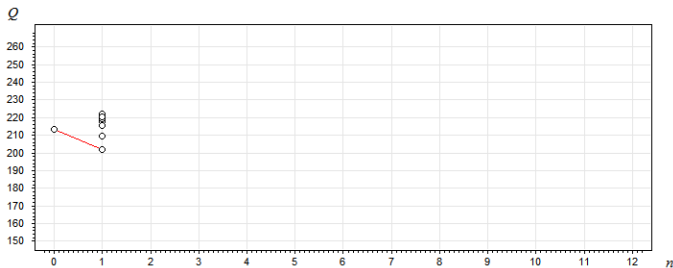
**для**  $j = 1, \dots, n$ , где  $j$  — сложность наборов

$J_j := \arg \min_{J: |J|=j} Q(J)$  — найти лучший набор сложности  $j$ ;

**если**  $Q(J_j) < Q^*$  **то**  $j^* := j$ ;  $Q^* := Q(J_j)$ ;

**если**  $j - j^* \geq d$  **то вернуть**  $J_{j^*}$ ;

## Алгоритм полного перебора (Full Search)



$$d = 3$$

$$j = 1$$

$$j^* = 1$$

**Вход:** множество  $F$ , **внешний критерий**  $Q$ , параметр  $d$ ;

инициализация:  $Q^* := Q(\emptyset)$ ;

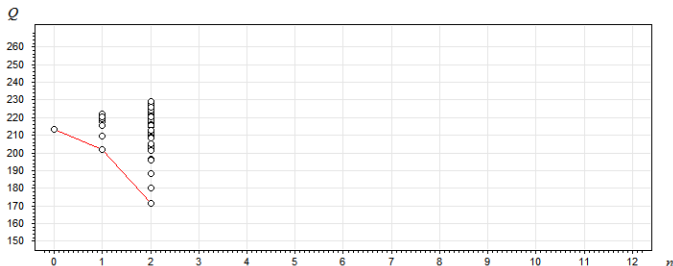
**для**  $j = 1, \dots, n$ , где  $j$  — сложность наборов

$J_j := \arg \min_{J: |J|=j} Q(J)$  — найти лучший набор сложности  $j$ ;

**если**  $Q(J_j) < Q^*$  **то**  $j^* := j$ ;  $Q^* := Q(J_j)$ ;

**если**  $j - j^* \geq d$  **то вернуть**  $J_{j^*}$ ;

## Алгоритм полного перебора (Full Search)



$$d = 3$$

$$j = 2$$

$$j^* = 2$$

**Вход:** множество  $F$ , **внешний критерий**  $Q$ , параметр  $d$ ;

инициализация:  $Q^* := Q(\emptyset)$ ;

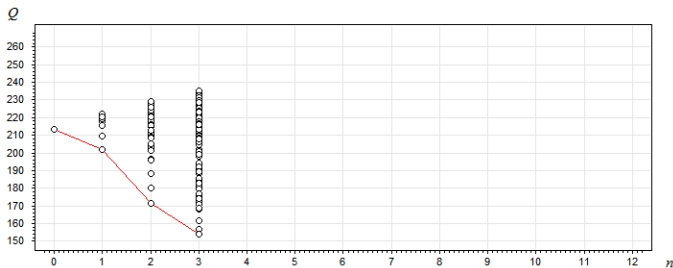
**для**  $j = 1, \dots, n$ , где  $j$  — сложность наборов

$J_j := \arg \min_{J: |J|=j} Q(J)$  — найти лучший набор сложности  $j$ ;

**если**  $Q(J_j) < Q^*$  **то**  $j^* := j$ ;  $Q^* := Q(J_j)$ ;

**если**  $j - j^* \geq d$  **то вернуть**  $J_{j^*}$ ;

## Алгоритм полного перебора (Full Search)



$$d = 3$$

$$j = 3$$

$$j^* = 3$$

**Вход:** множество  $F$ , **внешний критерий**  $Q$ , параметр  $d$ ;

инициализация:  $Q^* := Q(\emptyset)$ ;

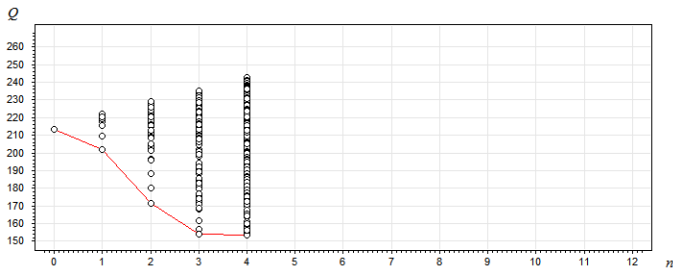
**для**  $j = 1, \dots, n$ , где  $j$  — сложность наборов

$J_j := \arg \min_{J: |J|=j} Q(J)$  — найти лучший набор сложности  $j$ ;

**если**  $Q(J_j) < Q^*$  **то**  $j^* := j$ ;  $Q^* := Q(J_j)$ ;

**если**  $j - j^* \geq d$  **то вернуть**  $J_{j^*}$ ;

## Алгоритм полного перебора (Full Search)



$$d = 3$$

$$j = 4$$

$$j^* = 4$$

**Вход:** множество  $F$ , **внешний критерий**  $Q$ , параметр  $d$ ;

инициализация:  $Q^* := Q(\emptyset)$ ;

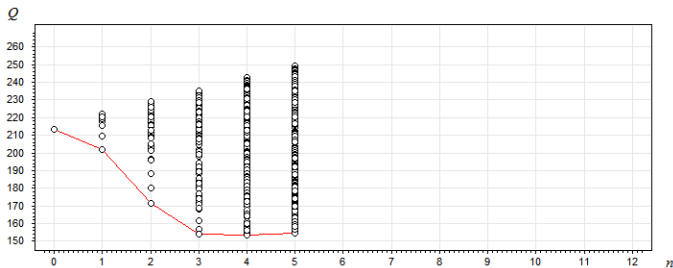
**для**  $j = 1, \dots, n$ , где  $j$  — сложность наборов

$J_j := \arg \min_{J: |J|=j} Q(J)$  — найти лучший набор сложности  $j$ ;

**если**  $Q(J_j) < Q^*$  **то**  $j^* := j$ ;  $Q^* := Q(J_j)$ ;

**если**  $j - j^* \geq d$  **то вернуть**  $J_{j^*}$ ;

## Алгоритм полного перебора (Full Search)



$$d = 3$$

$$j = 5$$

$$j^* = 4$$

**Вход:** множество  $F$ , **внешний критерий**  $Q$ , параметр  $d$ ;

инициализация:  $Q^* := Q(\emptyset)$ ;

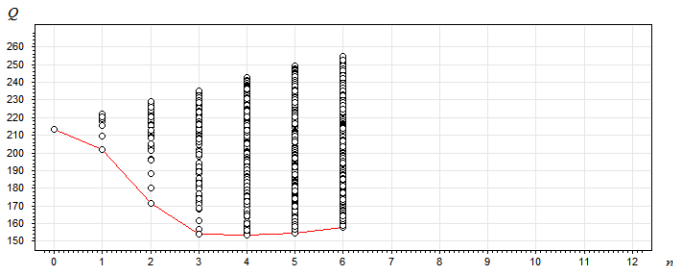
**для**  $j = 1, \dots, n$ , где  $j$  — сложность наборов

$J_j := \arg \min_{J: |J|=j} Q(J)$  — найти лучший набор сложности  $j$ ;

**если**  $Q(J_j) < Q^*$  **то**  $j^* := j$ ;  $Q^* := Q(J_j)$ ;

**если**  $j - j^* \geq d$  **то вернуть**  $J_{j^*}$ ;

## Алгоритм полного перебора (Full Search)



$$d = 3$$

$$j = 6$$

$$j^* = 4$$

**Вход:** множество  $F$ , **внешний критерий**  $Q$ , параметр  $d$ ;

инициализация:  $Q^* := Q(\emptyset)$ ;

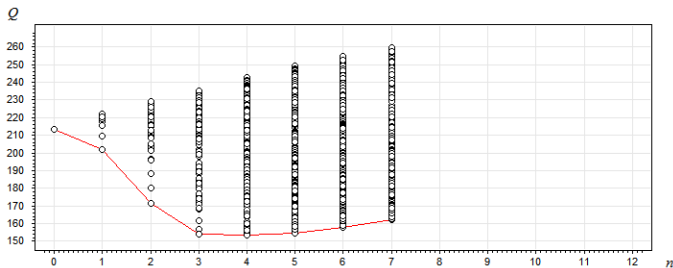
**для**  $j = 1, \dots, n$ , где  $j$  — сложность наборов

$J_j := \arg \min_{J: |J|=j} Q(J)$  — найти лучший набор сложности  $j$ ;

**если**  $Q(J_j) < Q^*$  **то**  $j^* := j$ ;  $Q^* := Q(J_j)$ ;

**если**  $j - j^* \geq d$  **то вернуть**  $J_{j^*}$ ;

## Алгоритм полного перебора (Full Search)



$$d = 3$$

$$j = 7$$

$$j^* = 4$$

**Вход:** множество  $F$ , **внешний критерий**  $Q$ , параметр  $d$ ;

инициализация:  $Q^* := Q(\emptyset)$ ;

**для**  $j = 1, \dots, n$ , где  $j$  — сложность наборов

$J_j := \arg \min_{J: |J|=j} Q(J)$  — найти лучший набор сложности  $j$ ;

**если**  $Q(J_j) < Q^*$  **то**  $j^* := j$ ;  $Q^* := Q(J_j)$ ;

**если**  $j - j^* \geq d$  **то вернуть**  $J_{j^*}$ ;

## Алгоритм полного перебора (Full Search)

### Преимущества:

- простота реализации;
- гарантированный результат;
- полный перебор эффективен, когда
  - информативных признаков не много,  $j^* \lesssim 5$ ;
  - всего признаков не много,  $n \lesssim 20..100$ .

### Недостатки:

- в остальных случаях оооооочень долго —  $O(2^n)$ ;
- чем больше перебирается вариантов, тем больше переобучение (особенно, если лучшие из вариантов существенно различны и одинаково плохи).

### Способы устранения:

- эвристические методы сокращённого перебора.

## Алгоритм жадного добавления (Add)

**Вход:** множество  $F$ , критерий  $Q$ , параметр  $d$ ;

инициализация:  $J_0 := \emptyset$ ;  $Q^* := Q(\emptyset)$ ;

**для**  $j = 1, \dots, n$ , где  $j$  — сложность наборов:

найти признак, наиболее выгодный для добавления:

$$f^* := \arg \min_{f \in F \setminus J_{j-1}} Q(J_{j-1} \cup \{f\});$$

добавить этот признак в набор:

$$J_j := J_{j-1} \cup \{f^*\};$$

**если**  $Q(J_j) < Q^*$  **то**  $j^* := j$ ;  $Q^* := Q(J_j)$ ;

**если**  $j - j^* \geq d$  **то вернуть**  $J_{j^*}$ ;

**Преимущество:** скорость  $O(n^2)$ , точнее  $O(nj^*)$ , вместо  $O(2^n)$

**Недостаток:** склонность включать в набор лишние признаки

**Способы устранения:** Del, Add-Del, Beam Search

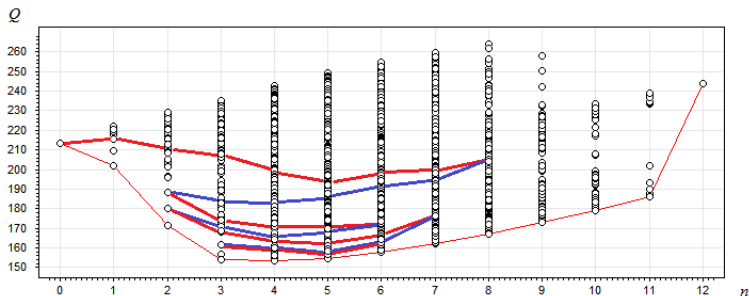
## Алгоритм поочерёдного добавления и удаления (Add-Del)

### Преимущества:

- как правило, лучше, чем Add и Del по отдельности;
- возможны быстрые инкрементные алгоритмы, пример — *шаговая регрессия* (step-wise regression).

### Недостатки:

- работает дольше, оптимальность не гарантирует.



## Алгоритм поочерёдного добавления и удаления (Add-Del)

инициализация:  $J_0 := \emptyset$ ;  $Q^* := Q(\emptyset)$ ;  $t := 0$ ;

**повторять**

**пока**  $|J_t| < n$  добавлять признаки (итерации Add):

$t := t + 1$  — началась следующая итерация;

$f^* := \arg \min_{f \in F \setminus J_{t-1}} Q(J_{t-1} \cup \{f\})$ ;  $J_t := J_{t-1} \cup \{f^*\}$ ;

**если**  $Q(J_t) < Q^*$  **то**  $t^* := t$ ;  $Q^* := Q(J_t)$ ;

**если**  $t - t^* \geq d$  **то прервать цикл**;

**пока**  $|J_t| > 0$  удалять признаки (итерации Del):

$t := t + 1$  — началась следующая итерация;

$f^* := \arg \min_{f \in J_{t-1}} Q(J_{t-1} \setminus \{f\})$ ;  $J_t := J_{t-1} \setminus \{f^*\}$ ;

**если**  $Q(J_t) < Q^*$  **то**  $t^* := t$ ;  $Q^* := Q(J_t)$ ;

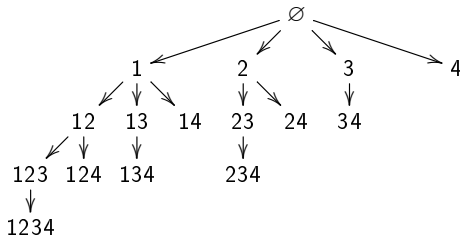
**если**  $t - t^* \geq d$  **то прервать цикл**;

**пока** значения критерия  $Q(J_{t^*})$  уменьшаются;

**вернуть**  $J_{t^*}$ ;

## Поиск в глубину (DFS, метод ветвей и границ)

Пример: дерево наборов признаков,  $n = 4$



Основные идеи:

- нумерация признаков по возрастанию номеров — чтобы избежать повторов при переборе подмножеств;
- если набор  $J$  бесперспективен, то больше не пытаться его наращивать.

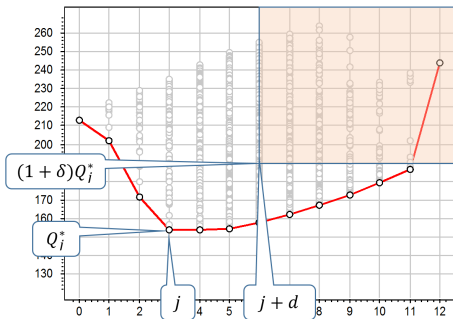
## Поиск в глубину (DFS, метод ветвей и границ)

Обозначим  $Q_j^*$  — значение критерия на самом лучшем наборе мощности  $j$  из всех до сих пор просмотренных.

**Оценка перспективности:**  
 набор  $J$  не наращивается,  
 если найдётся  $j$  такой, что

$$\begin{cases} Q(J) \geq (1 + \delta)Q_j^*; \\ |J| \geq j + d; \end{cases}$$

$d \geq 0$  и  $\delta \geq 0$  — параметры.



Чем меньше  $d$  и  $\delta$ , тем сильнее сокращается перебор.

## Поиск в глубину (DFS, метод ветвей и границ)

**Вход:** множество  $F$ , критерий  $Q$ , параметры  $d$  и  $\delta$ ;

**процедура** *нарастить* ( $J \subseteq F$ )

если найдётся  $j: j \leq |J| - d$  и  $Q(J) \geq (1 + \delta)Q_j^*$ , то

- └ набор  $J$  бесперспективный; **выход**;
- $Q_{|J|}^* := \min \{ Q_{|J|}^*, Q(J) \}$ ;
- для всех**  $f_s \in F$  таких, что  $s > \max \{ t \mid f_t \in J \}$ :
- └ *нарастить* ( $J \cup \{f_s\}$ );

инициализировать массив лучших значений критерия:

$Q_j^* := Q(\emptyset)$  для всех  $j = 1, \dots, n$ ;

**упорядочить** признаки по убыванию информативности;

*нарастить* ( $\emptyset$ );

**вернуть**  $J$ , для которого  $Q(J) = \min_{j=1, \dots, n} Q_j^*$ ;

## Поиск в ширину (beam search, breadth-first search, BFS)

(не пучка, не пучком, не лучом, и не «лучевой поиск»)

Он же *многорядный итерационный алгоритм МГУА*

(Метод Группового Учёта Аргументов А.Г.Ива́хненко)

*Принцип неокончательных решений* Габора: оставлять больше свободы выбора для принятия последующих решений

Усовершенствуем алгоритм Add:

на каждой  $j$ -й итерации будем строить не один набор,  
а множество из  $B_j$  наборов, называемое  $j$ -м рядом:

$$R_j = \{J_j^1, \dots, J_j^{B_j}\}, \quad J_j^b \subseteq F, \quad |J_j^b| = j, \quad b = 1, \dots, B_j.$$

где  $B_j \leq B$  — параметр *ширины* (пучка, ряда) *поиска*.

---

*Ивахненко А. Г., Зайченко Ю. П., Димитров В. Д.* Принятие решений на основе самоорганизации. 1976.

*Ивахненко А. Г., Юрачковский Ю. П.* Моделирование сложных систем по экспериментальным данным, 1987.

## Усечённый поиск в ширину (Beam Search)

**Вход:** множество  $F$ , критерий  $Q$ , параметры  $d, B$ ;

первый ряд состоит из всех наборов длины 1:

$$R_1 := \{\{f_1\}, \dots, \{f_n\}\}; \quad Q^* = Q(\emptyset);$$

**для**  $j = 1, \dots, n$ , где  $j$  — сложность наборов:

отсортировать ряд  $R_j = \{J_j^1, \dots, J_j^{B_j}\}$

по возрастанию критерия:  $Q(J_j^1) \leq \dots \leq Q(J_j^{B_j})$ ;

**если**  $B_j > B$  **то**

└  $R_j := \{J_j^1, \dots, J_j^B\}$  — оставить  $B$  лучших наборов ряда;

**если**  $Q(J_j^1) < Q^*$  **то**  $j^* := j$ ;  $Q^* := Q(J_j^1)$ ;

**если**  $j - j^* \geq d$  **то вернуть**  $J_{j^*}^1$ ;

породить следующий ряд:

$$R_{j+1} := \{J \cup \{f\} \mid J \in R_j, f \in F \setminus J\};$$

## Усечённый поиск в ширину: дополнительные эвристики

- **Трудоёмкость:**  
 $O(Bn^2)$ , точнее  $O(Bn(j^* + d))$ .
- **Проблема дубликатов:**  
после сортировки  $Q(J_j^1) \leq \dots \leq Q(J_j^{B_j})$  проверить на совпадение только соседние наборы с равными значениями внутреннего и внешнего критерия.
- **Адаптивный отбор признаков:**  
на последнем шаге добавлять к  $j$ -му ряду только признаки  $f$  с наибольшей информативностью  $I_j(f)$ :

$$I_j(f) = \sum_{b=1}^{B_j} [f \in J_j^b].$$

## Эволюционный алгоритм поиска (идея и терминология)

$J \subseteq F$  — индивид (в МГУА «модель»);

$R_t := \{J_t^1, \dots, J_t^{B_t}\}$  — поколение (в МГУА — «ряд»);

$\beta = (\beta_j)_{j=1}^n$ ,  $\beta_j = [f_j \in J]$  — хромосома, кодирующая  $J$ ;

Бинарная операция скрещивания (crossover)  $\beta = \beta' \times \beta''$ :

- вариант 1:  $\beta_j = \rho_j \beta'_j + (1 - \rho_j) \beta''_j$ ,  $\rho_j \sim \text{uni}(0, 1)$
- вариант 2:  $\beta = (\beta'_1, \dots, \beta'_s, \beta''_{s+1}, \dots, \beta''_n)$ ,  $s \sim \text{uni}(1, \dots, n)$ ,  
 надо задавать «естественное» ранжирование признаков

Унарная операция мутации  $\beta = \sim \beta'$ :

- $\beta_j = \rho_j (1 - \beta'_j) + (1 - \rho_j) \beta'_j$ ,  $\rho_j \sim \text{bin}(p_m)$ ,  
 где  $p_m$  — параметр вероятности мутации.

## Эволюционный (генетический) алгоритм

**Вход:** множество  $F$ , критерий  $Q$ , параметры:  $d, p_m$ ,  
 $B$  — размер популяции,  $T$  — число поколений;

инициализировать случайную популяцию из  $B$  наборов:

$B_1 := B$ ;  $R_1 := \{J_1^1, \dots, J_1^{B_1}\}$ ;  $Q^* := Q(\emptyset)$ ;

**для**  $t = 1, \dots, T$ , где  $t$  — номер поколения:

ранжирование индивидов:  $Q(J_t^1) \leq \dots \leq Q(J_t^{B_t})$ ;

**если**  $B_t > B$  **то** селекция:  $R_t := \{J_t^1, \dots, J_t^B\}$ ;

**если**  $Q(J_t^1) < Q^*$  **то**  $t^* := t$ ;  $Q^* := Q(J_t^1)$ ;

**если**  $t - t^* \geq d$  **то вернуть**  $J_{t^*}^1$ ;

породить  $t+1$ -е поколение путём скрещиваний и мутаций:

$R_{t+1} := \{\sim(J' \times J'') \mid J', J'' \in R_t\} \cup R_t$ ;

## Эвристики для управления процессом эволюции

- Увеличивать вероятности перехода признаков от более успешного родителя к потомку.
- Накапливать оценки информативности признаков. Чем более информативен признак, тем выше вероятность его включения в набор во время мутации.
- Применение совокупности критериев качества.
- Скрещивать только лучшие индивиды (элитаризм).
- Переносить лучшие индивиды в следующее поколение.
- В случае стагнации увеличивать вероятность мутаций.
- Параллельно выращивается несколько изолированных популяций (островная модель эволюции).

## Попытка обоснования. Теорема схемы

Схема — вектор  $H = (h_1, \dots, h_n)$ , где  $h_j \in \{0, 1, *\}$

$o(H)$  — порядок схемы, число не\* в схеме

$d(H)$  — длина схемы, расстояние между первым и последним не\*, число мест, в которых кроссовер может нарушить схему

$f(H, t)$  — степень приспособленности схемы, среднее  $Q$  по всем векторам, подходящим под схему в поколении  $t$

$\bar{f}(t) = f(*^n, t)$  — средняя приспособленность популяции

$p_c$  — вероятность кроссовера (только второй вариант)

$p_m$  — вероятность мутации

### Теорема схемы [Холланд, 1975]

Число индивидов схемы  $H$  в популяции поколения  $t$ :

$$Em(H, t + 1) \geq m(H, t) \frac{f(H, t)}{\bar{f}(t)} \left( 1 - p_c \frac{d(H)}{n - 1} - p_m o(H) \right)$$

## Интерпретация теоремы схемы

Число индивидов схемы  $H$  в популяции поколения  $t$ :

$$Em(H, t + 1) \geq m(H, t) \frac{f(H, t)}{\bar{f}(t)} \left( 1 - p_c \frac{d(H)}{n-1} - p_m o(H) \right)$$

- **Строительный блок** — схема  $H$  с низким порядком  $o(H)$ , короткой длиной  $d(H)$ , высокой приспособленностью  $f(H, t)$
- Если приспособленность строительного блока оказалась выше средней в популяции, то число его индивидов будет расти экспоненциально в последующих популяциях
- **Гипотеза** (building block hypothesis): «строительные блоки объединяются, чтобы сформировать ещё лучшие блоки»

---

*John Henry Holland.* Adaptation in natural and artificial systems. 1992

*David White.* An overview of schema theory. 2014

## Обобщение: случайный поиск с адаптацией (СПА)

**Вход:** множество  $F$ , критерий  $Q$ , параметры:  $d$ ,

$B$  — размер популяции,  $T$  — число поколений;

инициализация  $p_1 = \dots = p_n := 1/n$ ;

популяция  $R_1 := \{J_1^1, \dots, J_1^{B_1} \sim \{p_1, \dots, p_n\}\}$ ;  $Q^* := Q(\emptyset)$ ;

для  $t = 1, \dots, T$ , где  $t$  — номер поколения:

ранжирование индивидов:  $Q(J_t^1) \leq \dots \leq Q(J_t^{B_t})$ ;

если  $B_t > B$  то селекция:  $R_t := \{J_t^1, \dots, J_t^B\}$ ;

если  $Q(J_t^1) < Q^*$  то  $t^* := t$ ;  $Q^* := Q(J_t^1)$ ;

если  $t - t^* \geq d$  то вернуть  $J_{t^*}^1$ ;

увеличить  $p_j$  для признаков из лучших наборов;

уменьшить  $p_j$  для признаков из худших наборов;

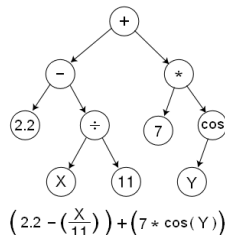
породить  $t+1$ -е поколение из  $B_t$  наборов:

$R_{t+1} := \{J_{t+1}^1, \dots, J_{t+1}^{B_{t+1}} \sim \{p_1, \dots, p_n\}\} \cup R_t$ ;

Н.Г.Загоруйко, В.Н.Ёлкина, Г.С.Лбов. Алгоритмы обнаружения эмпирических закономерностей. 1985

## Символьная регрессия

- 1 система порождающих функций  $g_k(x; w)$  с параметрами  $w$
- 2 механизм перебора суперпозиций (генетическое программирование)
  - обмен поддеревьями
  - случайные мутации
- 3 обучение параметров суперпозиции: SG
- 4 отбор — по внешним критериям
- 5 ограничение сложности суперпозиций
  - регуляризация, байесовское обучение



*J.Koza. On the programming of computers by means of natural selection. 1992*

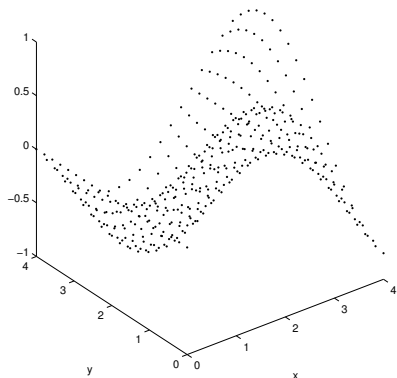
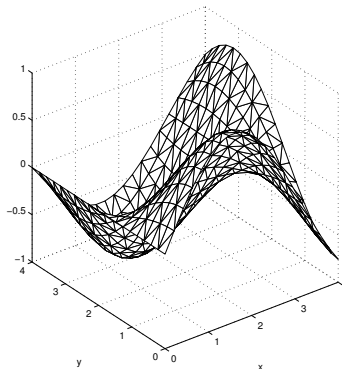
*Ying Jin et al. Bayesian Symbolic Regression. 2019*

*Р.А.Сологуб. Алгоритмы индуктивного порождения и трансформации моделей в задачах нелинейной регрессии. Диссертация к.ф.-м.н., 2014.*

## Пример. Эксперимент на синтетических данных

Модель:  $y = f(w, x) = \sin(x_1) * \sin(w_1 x_2 + w_2)$

Выборка из 380 точек,  
 $x_i \in \mathbb{R}^2$ ,  $y_i \in \mathbb{R}$



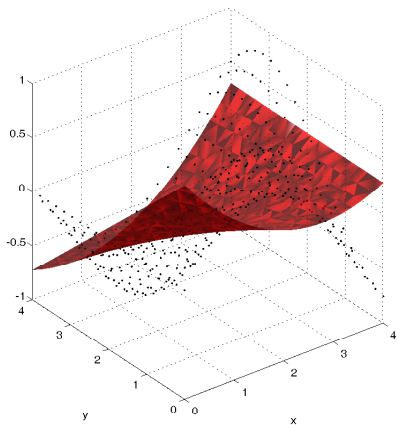
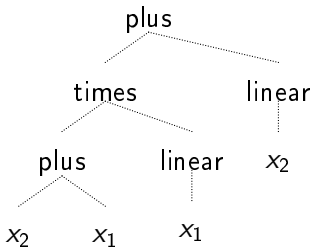
## Пример. Система порождающих функций

Функция	Описание	Параметры
$g(x_1, x_2; w)$		
plus	$y = x_1 + x_2$	–
times	$y = x_1 x_2$	–
$g(x_1; w)$		
divide	$y = 1/x$	–
multiply	$y = ax$	$a$
add	$y = x + a$	$a$
normal	$y = \frac{\lambda}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x-\xi)^2}{2\sigma^2}\right) + a$	$\lambda, \sigma, \xi, a$
linear	$y = ax + b$	$a, b$
parab	$y = ax^2 + bx + c$	$a, b, c$
sin	$y = \sin(x)$	–
logsig	$y = \frac{\lambda}{1 + \exp(-\sigma(x-\xi))} + a$	$\lambda, \sigma, \xi, a$



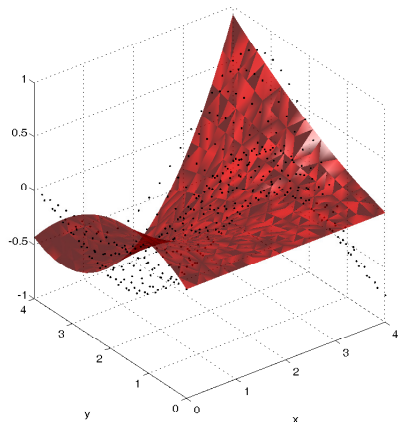
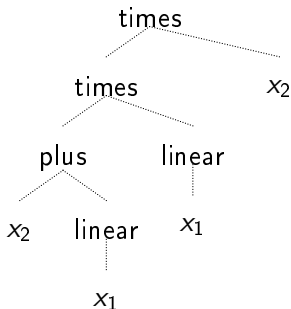
## Пример. Перебор моделей-претендентов

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



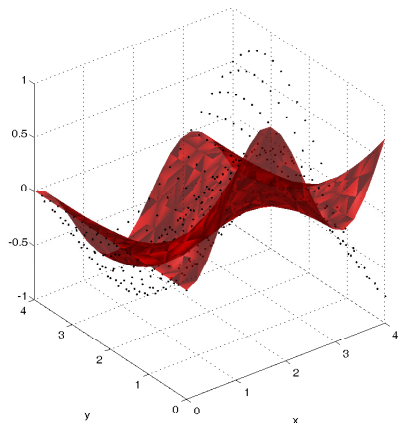
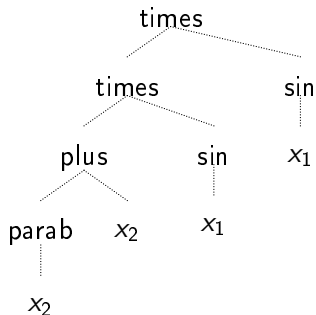
## Пример. Перебор моделей-претендентов

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



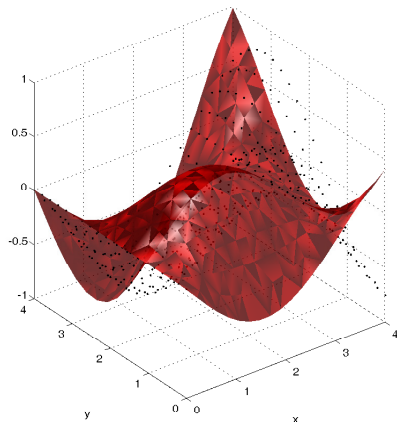
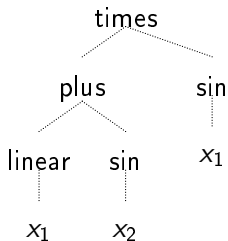
## Пример. Перебор моделей-претендентов

$\text{times}(\emptyset, \text{times}(\emptyset, \text{plus}(\emptyset, \text{parab}(w_{1:3}, x_2), x_2), \sin(\emptyset, x_1)), \sin(\emptyset, x_1))$



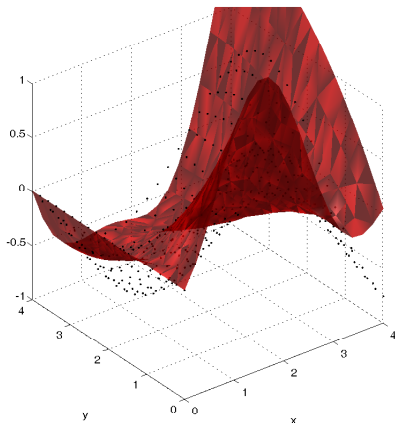
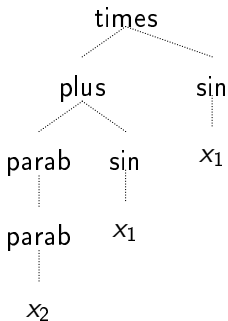
## Пример. Перебор моделей-претендентов

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



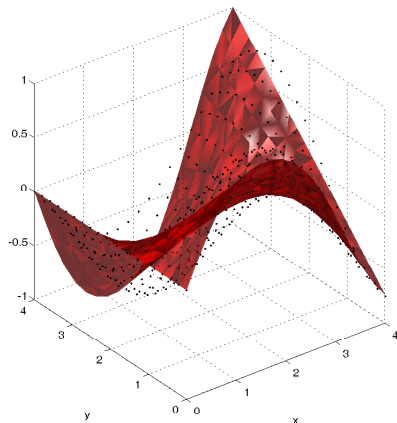
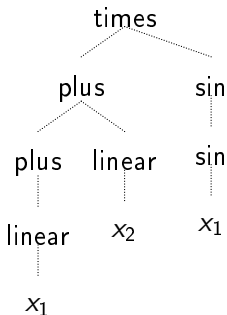
## Пример. Перебор моделей-претендентов

$\text{times}(\emptyset, \text{plus}(\emptyset, \text{parab}(w_{1:3}, \text{parab}(w_{4:6}, x_2)), \text{sin}(\emptyset, x_1)), \text{sin}(\emptyset, x_1))$



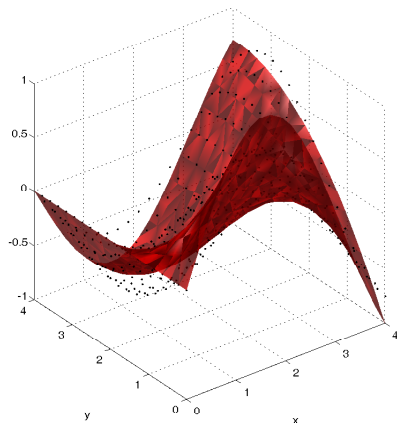
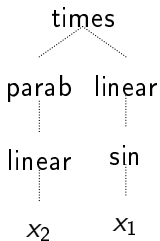
## Пример. Перебор моделей-претендентов

$\text{times}(\emptyset, \text{plus}(\emptyset, \text{plus}(w_1, \text{linear}(w_{2:3}, x_1))), \text{linear}(w_{4:5}, x_2)), \text{sin}(\emptyset, \text{sin}(\emptyset, x_1)))$



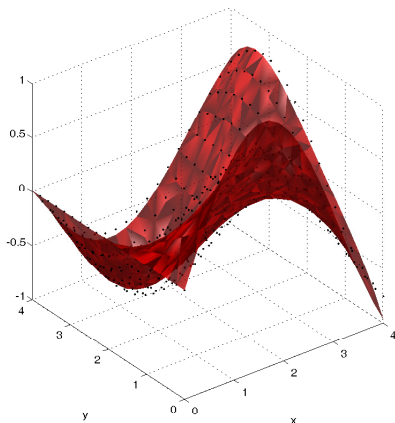
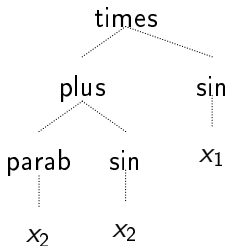
## Пример. Перебор моделей-претендентов

$\text{times}(\emptyset, \text{parab}(w_{1:3}, \text{linear}(w_{4:5}, x_2)), \text{linear}(w_{6:7}, \sin(\emptyset, x_1)))$



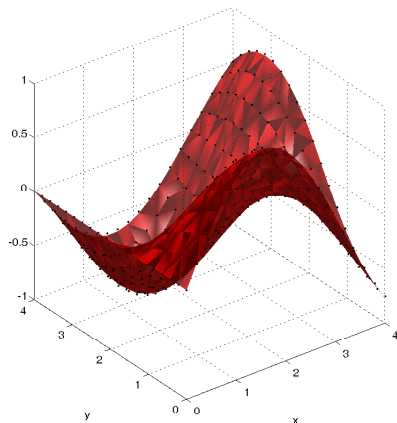
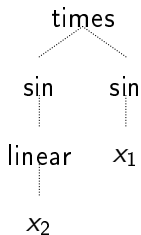
## Пример. Перебор моделей-претендентов

$\text{times}(\emptyset, \text{plus}(\emptyset, \text{parab}(w_{1:3}, x_2), \text{sin}(\emptyset, x_2)), \text{sin}(\emptyset, x_1))$



## Пример. Перебор моделей-претендентов

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



- Комбинаторный отбор признаков может использовать любые эвристические методы дискретной оптимизации

$$Q(J) \rightarrow \min_{J \subseteq F}$$

- $Q(J)$  должен быть внешним критерием, с характерным минимумом по сложности модели
- Большинство эвристик эксплуатируют две основные идеи:
  - признаки ранжируются по их полезности
  - $Q(J)$  изменяется не сильно при малом изменении  $J$
- МГУА, ЭА и СПА очень похожи — на их основе можно изобретать новые «симбиотические» мета-эвристики
- Символьная регрессия — перебор суперпозиций формул
- Генетическое программирование — перебор программ