Курс «Введение в машинное обучение»

Эволюционные методы машинного обучения

Воронцов Константин Вячеславович
k.v.vorontsov@phystech.edu
http://www.MachineLearning.ru/wiki?title=User:Vokov

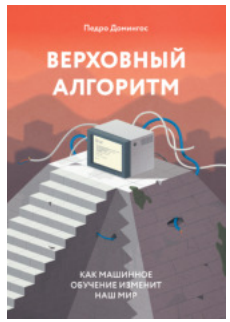
Этот курс доступен на странице вики-ресурса
http://www.MachineLearning.ru/wiki
«Введение в машинное обучение (курс лекций, К.В.Воронцов)»

МФТИ.ФПМИ.ИС.ИАД • 17 апреля 2025

Основные научные школы машинного обучения

1. *символизм* – поиск логических закономерностей
   - Decision Tree, Rule Induction
2. *коннекционизм* – обучаемые нейронные сети
   - BackPropagation, Deep Belief Nets, Deep Learning
     CNN, ResNet, LSTM, GRU, Attention, Transformer
3. *эволюционизм* – саморазвитие сложных моделей
   - Genetic Algorithms, Genetic Programming, Symbolic Regression
4. *байесионизм* и вероятностно-статистические методы
   - MLE, EM, GLM, LR, OBC, Naive Bayes, QD, LDF
     Bayesian Networks, Bayesian Learning, Graphical Models
5. *аналогизм* – «близким объектам близкие ответы»
   - kNN, RBF, SVM, KDE, Kernel Smoothing

⊕ *композиционизм* – кооперация моделей
   - Weighted Voting, Boosting, Bagging, Stacking,
     Random Forest, Яндекс.CatBoost



Педро Домингос. Верховный алгоритм. 2016. 336 с.

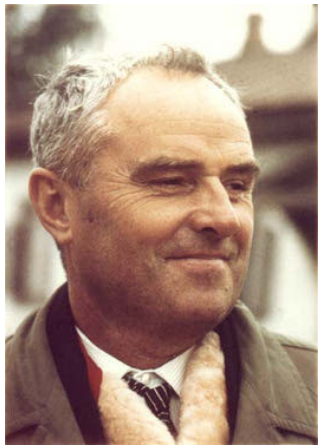
Содержание

Научная школа А. Г. Ива́хненко

Метод группового учёта аргументов, МГУА (Group Method of Data Handling, GMDH)

Принцип *самоорганизации моделей* — перебор структур и оптимизация параметров модели

- Качество моделей оценивается в процессе перебора по многим *внешним критериям:*
  — скользящий контроль
  — помехоустойчивость моделирования
  — баланс / согласованность прогнозов и др.
- Первая 8-слойная глубокая нейросеть (1965)
- Сотни применений, около 300 диссертаций



Алексей Григорьевич Ива́хненко (1913–2007)

Ивахненко А. Г., Лапа В. Г. Кибернетические предсказывающие устройства. 1965.
Ivakhnenko A. G. Heuristic self-organization in problems of engineering cybernetics. 1970.
Ивахненко А. Г. Индуктивный метод самоорганизации моделей сложных систем. 1982.

Задачи выбора модели и метода обучения

Дано: X — пространство объектов; Y — множество ответов;
$X^\ell = (x_i, y_i)_{i=1}^\ell$ — обучающая выборка, $y_i = y(x_i)$;
$A_t = \{a \colon X \times W_t \to Y\}$ — параметрические модели, $t \in T$
W_t — пространство параметров модели A_t
$\mu_t \colon (X \times Y)^\ell \to W_t$ — методы обучения, $t \in T$

Найти: метод μ_t с наилучшей *обобщающей способностью*.

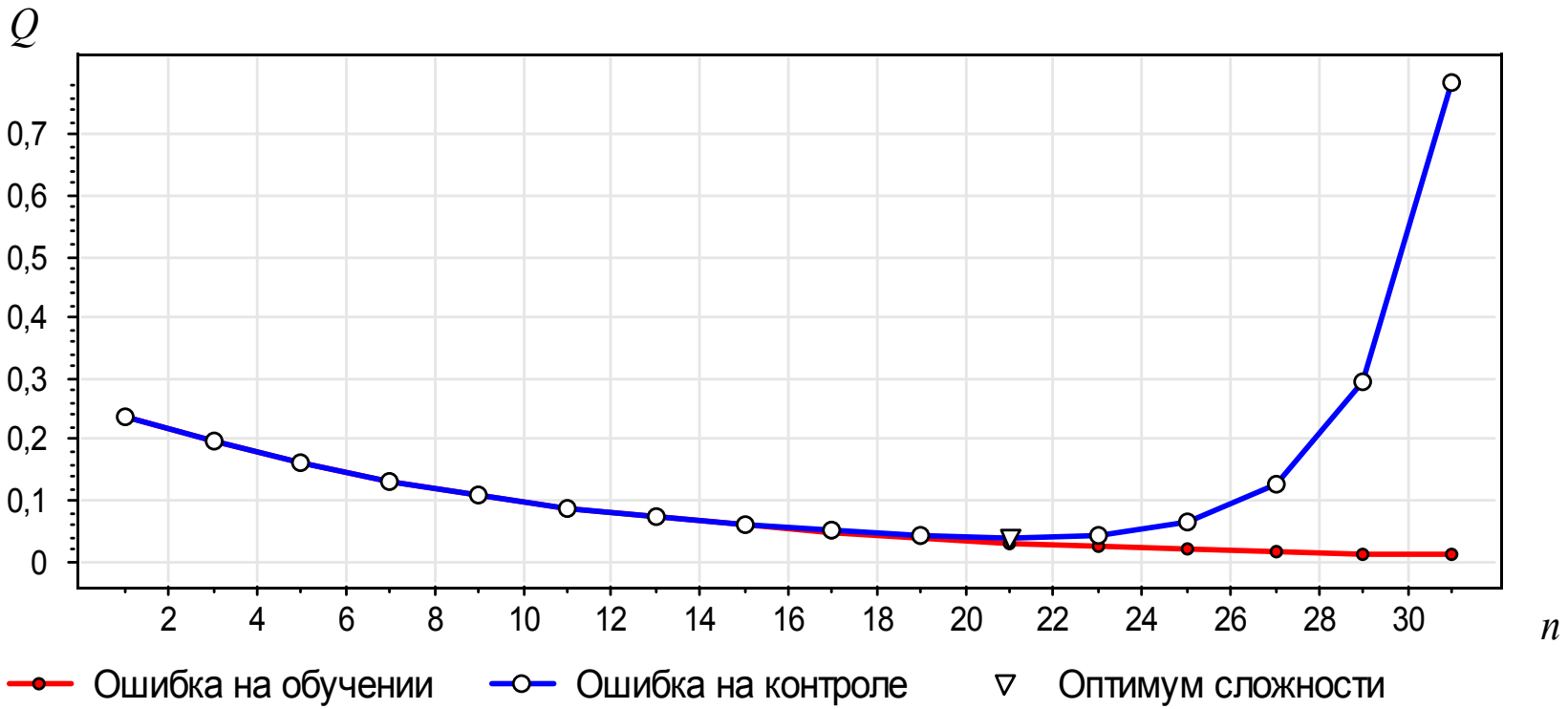
Частные случаи:

- выбор лучшей модели A_t (model selection);
- выбор метода обучения μ_t для заданной модели A (в частности, оптимизация *гиперпараметров*);
- отбор признаков (feature selection)
- перебор формул как суперпозиций элементарных функций (symbolic regression, genetic programming)

Напоминание. Внешние и внутренние критерии

Внутренний критерий монотонно убывает с ростом сложности модели (например, числа признаков).

Внешний критерий имеет характерный минимум, соответствующий оптимальной сложности модели.



Напоминание. Кросс-проверка (cross-validation, CV)

Усреднение по множеству разбиений $X^L = X_n^\ell \sqcup X_n^k$, $n \in N$:

$$\mathrm{CV}(\mu, X^L) = \frac{1}{|N|} \sum_{n \in N} Q_\mu(X_n^\ell, X_n^k)$$

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

Критерии непротиворечивости моделей

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

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

$$D_1(\mu, X^L) = \frac{1}{L} \sum_{i=1}^{L} \left|\mu(X^\ell)(x_i) - \mu(X^k)(x_i)\right|.$$

2. Аналог $\mathrm{CV}_{t\times 2}$: по t разбиениям $X^L = X_s^\ell \sqcup X_s^k$, $s = 1, \ldots, t$:

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

Преимущество: походит также для обучения без учителя.

Недостатки:

— объём обучения сокращается в 2 раза;

— трудоёмкость возрастает в $2t$ раз.

Критерии регуляризации

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):

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

Разновидности L_0-регуляризации

Информационный критерий Акаике (Akaike Information Criterion):

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

где $\hat{\sigma}^2$ — оценка дисперсии ошибки $\mathrm{D}\big(y_i - a(x_i)\big)$,
J — подмножество используемых признаков.

Байесовский информационный критерий (Bayes Inform. Criterion):

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

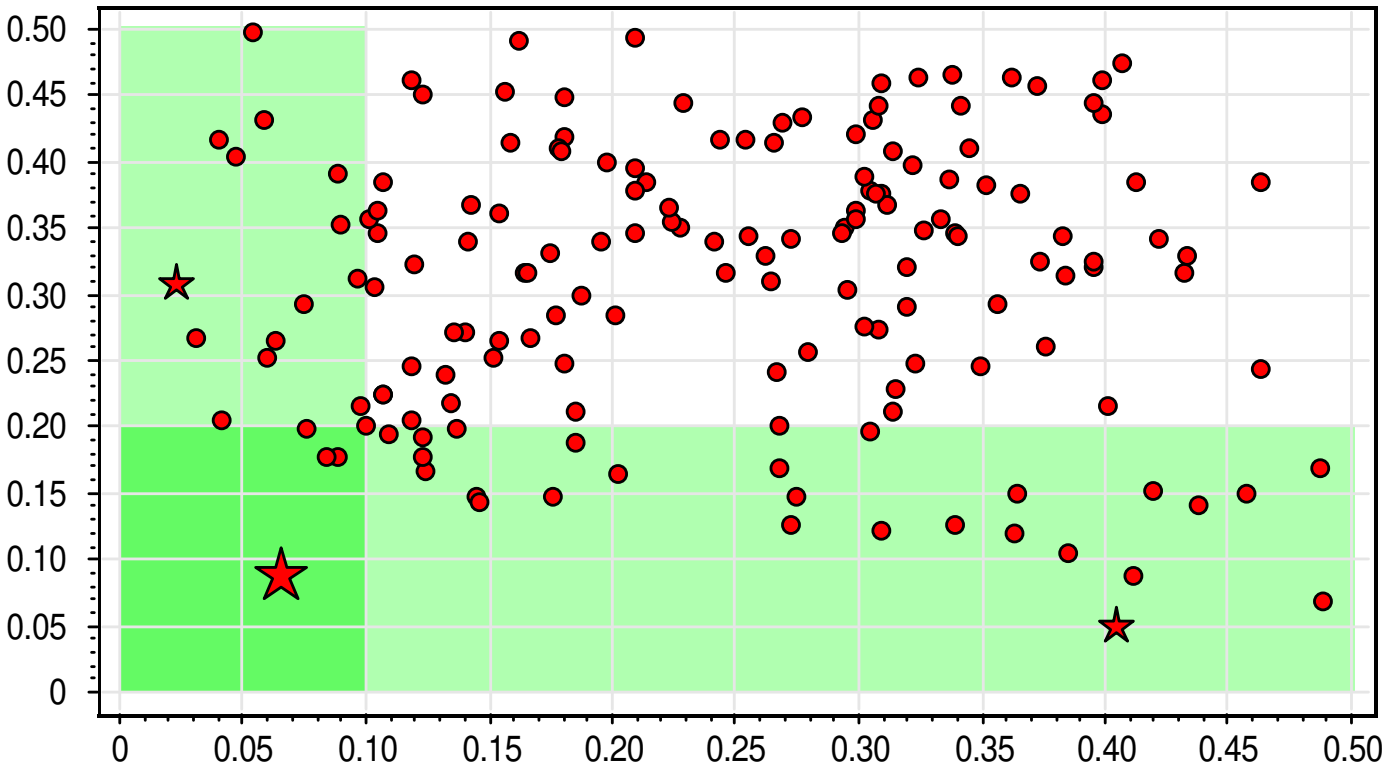
Оценка Вапника-Червоненкиса (VC-bound):

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

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

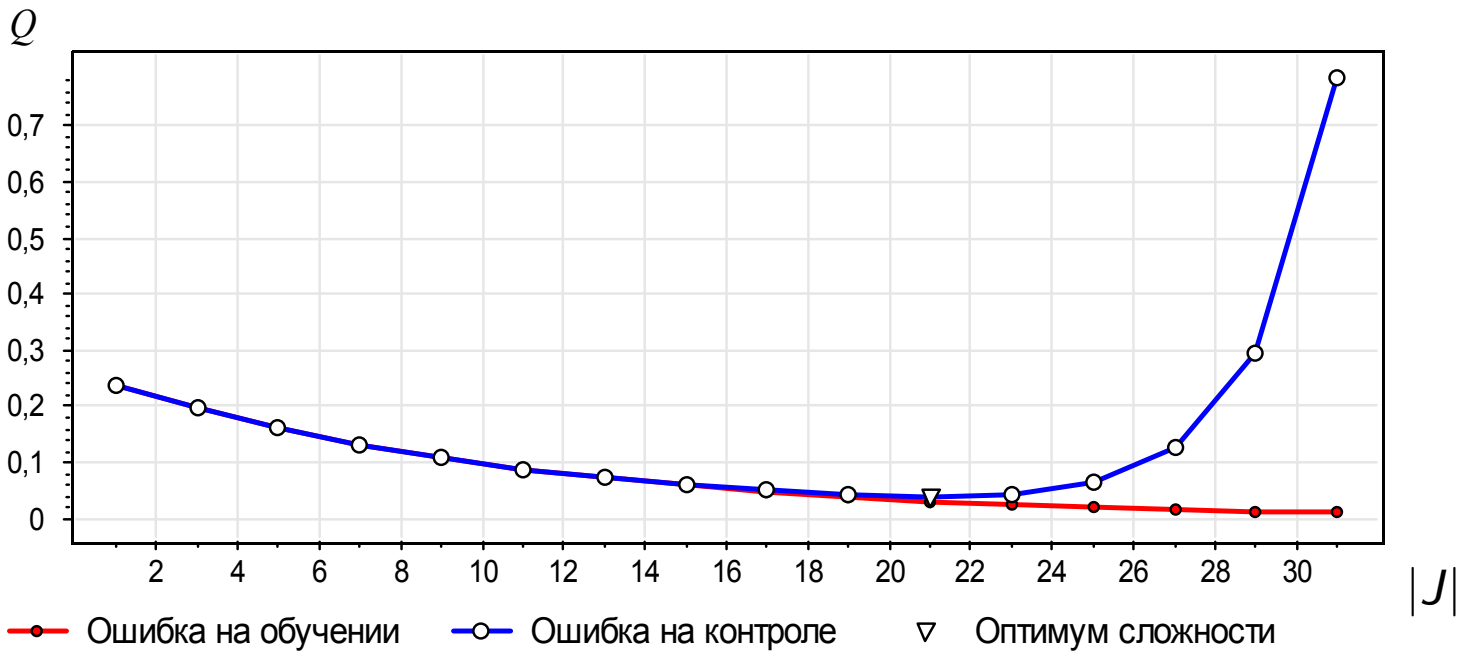
$F = \{f_j : X \to D_j : j = 1, \ldots, n\}$ — множество признаков;

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

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

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

Внутренний критерий и внешний критерий:



Задача отбора признаков в логических закономерностях

Закономерность R — *конъюнкция* пороговых условий:

$$R(x) = \bigwedge_{j \in J} \big[f_j(x) \geqslant a_j \big].$$

Критерий информативности относительно класса $c \in Y$:

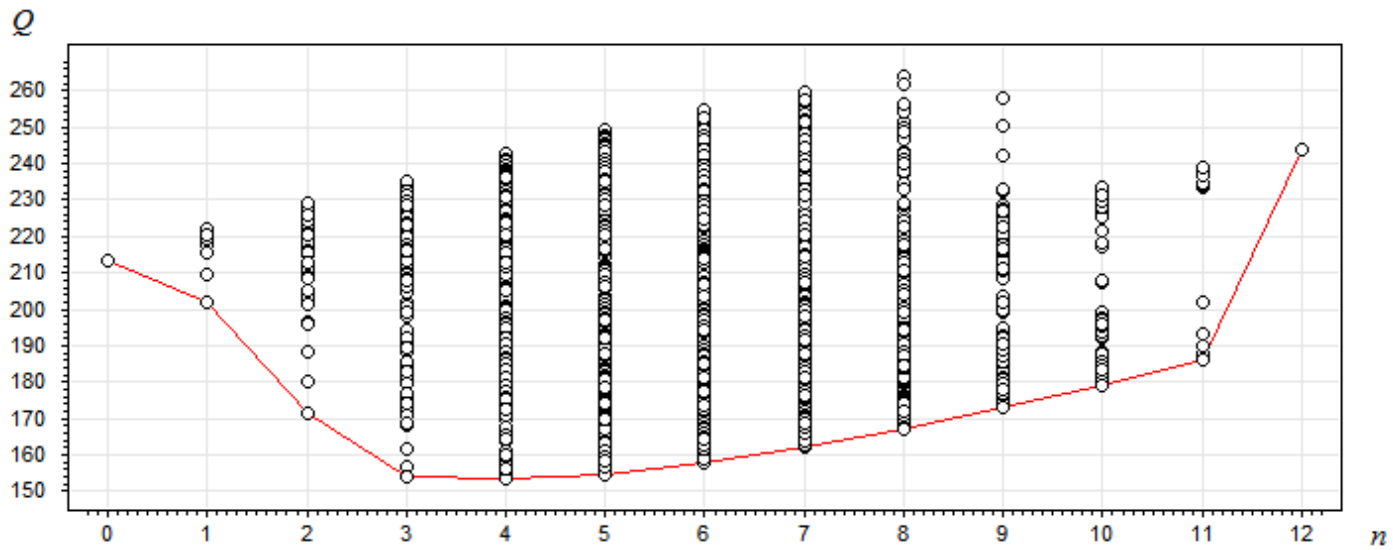
$$I(p,n) \to \max_{J, \{a_j\}}; \quad \begin{cases} p(R) = \#\big\{x_i \colon R(x_i)=1 \text{ и } y_i=c\big\} \to \max \\ n(R) = \#\big\{x_i \colon R(x_i)=1 \text{ и } y_i \neq c\big\} \to \min \end{cases}$$

Аналогично внешним критериям,
информативность $I(p,n)$ имеет оптимум по сложности $|J|$:

- слишком мало признаков $\Rightarrow$ большие n, низкая $I(p,n)$
- оптимально признаков $\Rightarrow$ малые n, большие p, высокая $I(p,n)$
- слишком много признаков $\Rightarrow$ малые $p+n$, низкая $I(p,n)$

Отличие: закономерностей нужно много, это новые признаки

Алгоритм полного перебора (Full Search)



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

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

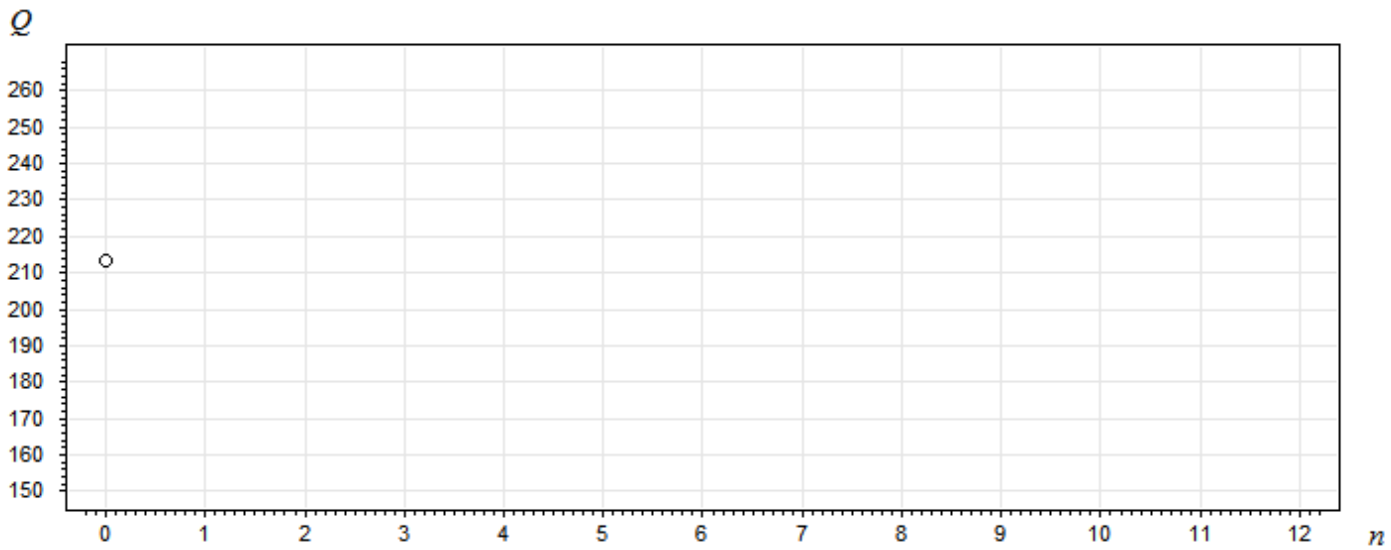
для $j = 1, \ldots, n$, где j — сложность наборов

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

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

если $j - j^* \geqslant d$ **то выход** J_{j^*};

Алгоритм полного перебора (Full Search)



$d = 3$
$j = 0$

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

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

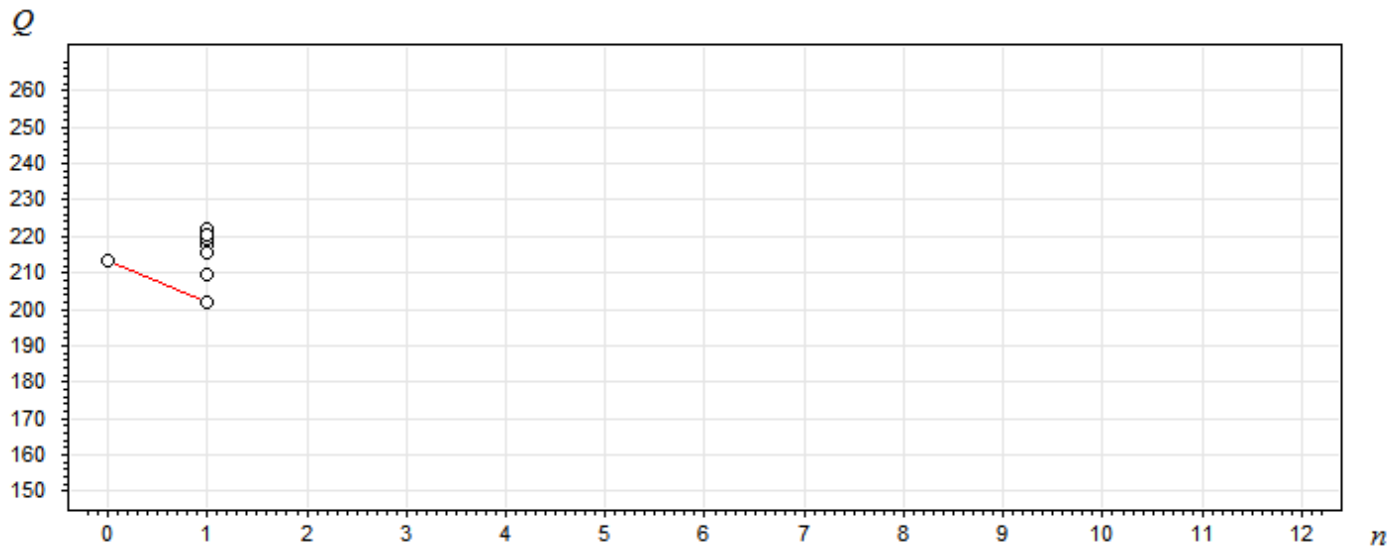
для $j = 1, \ldots, n$, где j — сложность наборов

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

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

  если $j - j^* \geqslant d$ **то выход** J_{j^*};

Алгоритм полного перебора (Full Search)



$d = 3$
$j = 1$
$j^* = 1$

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

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

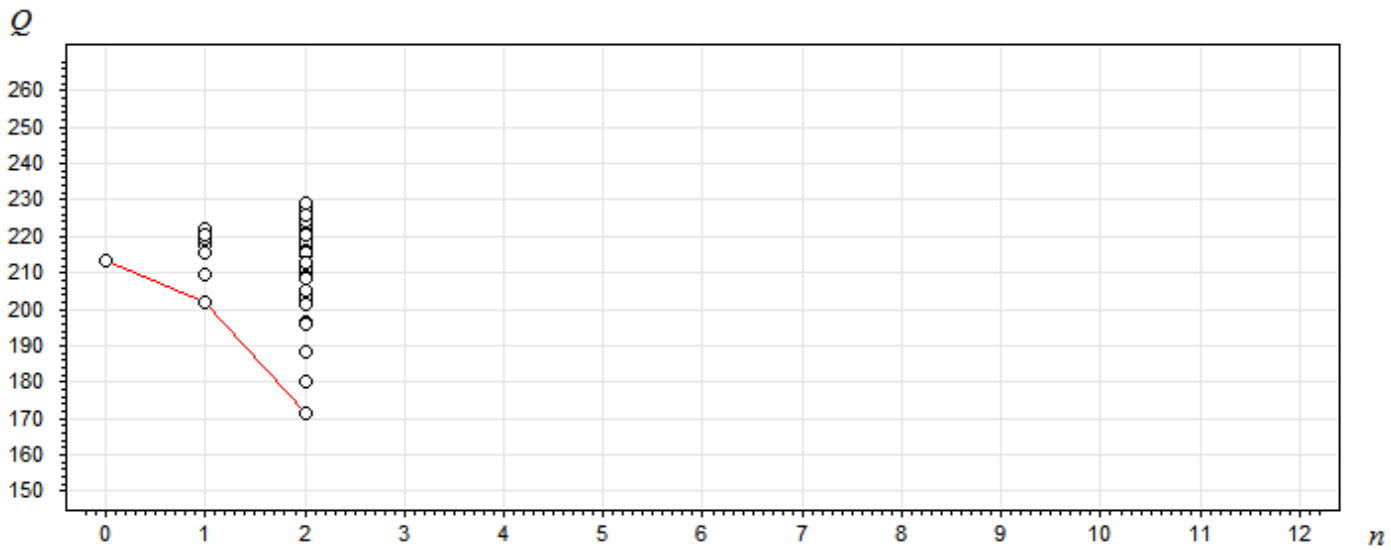
для $j = 1, \ldots, n$, где j — сложность наборов

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

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

  если $j - j^* \geqslant d$ **то выход** J_{j^*};

Алгоритм полного перебора (Full Search)



$d = 3$
$j = 2$
$j^* = 2$

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

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

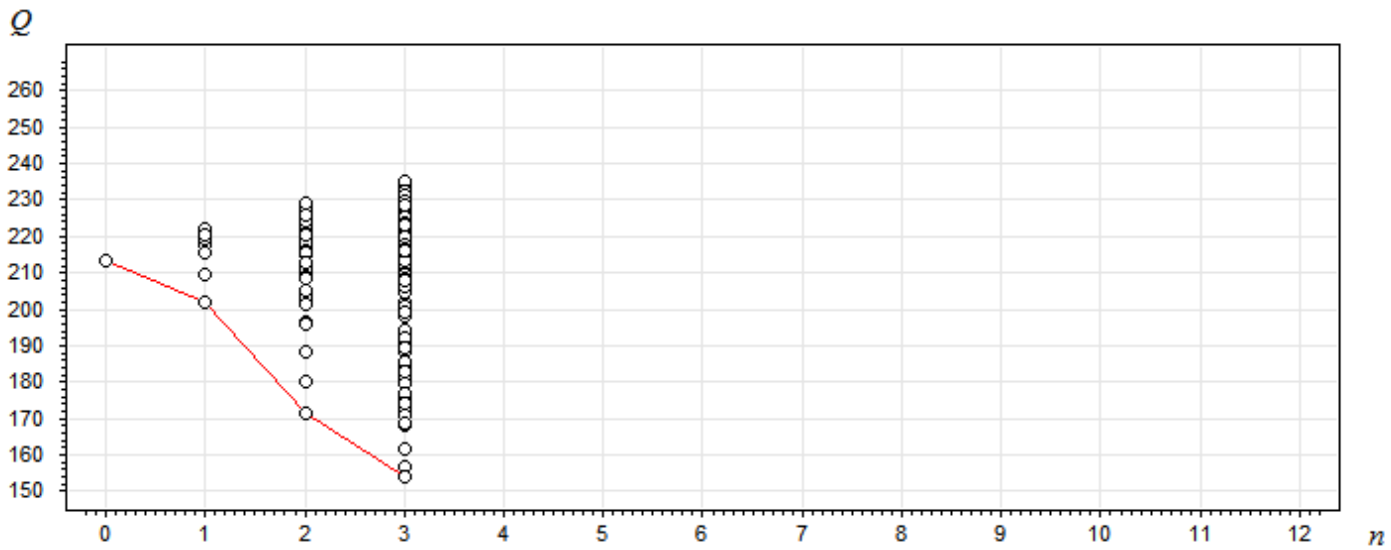
для $j = 1, \ldots, n$, где j — сложность наборов

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

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

  если $j - j^* \geqslant d$ **то выход** J_{j^*};

Алгоритм полного перебора (Full Search)



$d = 3$
$j = 3$
$j^* = 3$

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

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

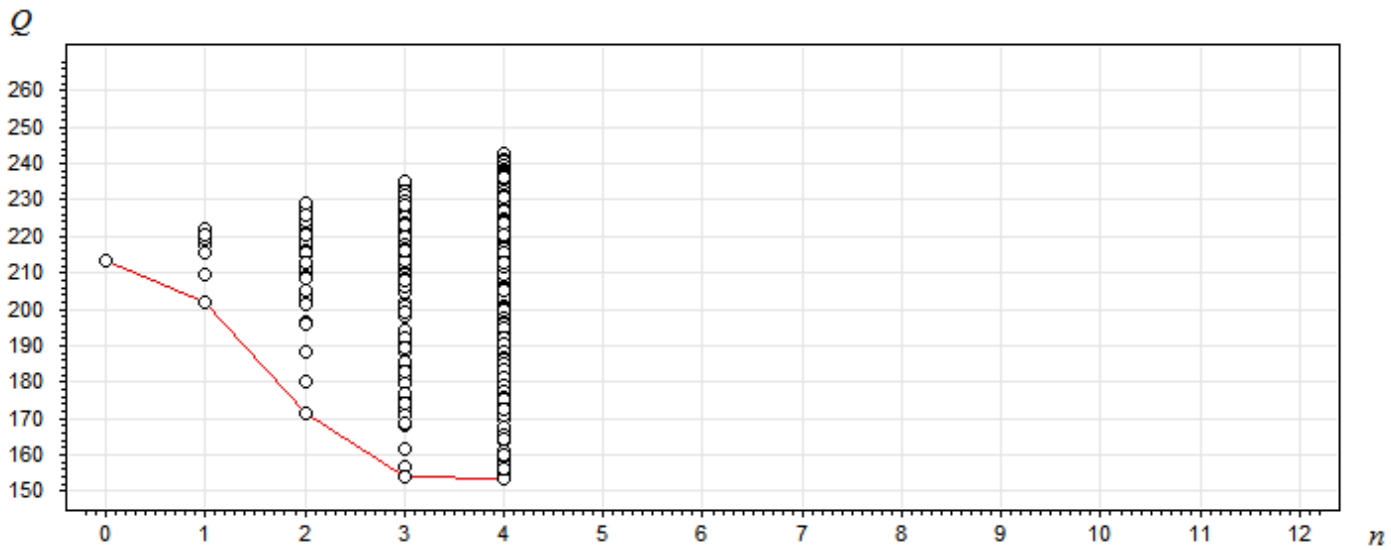
для $j = 1, \ldots, n$, где j — сложность наборов

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

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

  если $j - j^* \geqslant d$ **то выход** J_{j^*};

Алгоритм полного перебора (Full Search)



$d = 3$
$j = 4$
$j^* = 4$

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

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

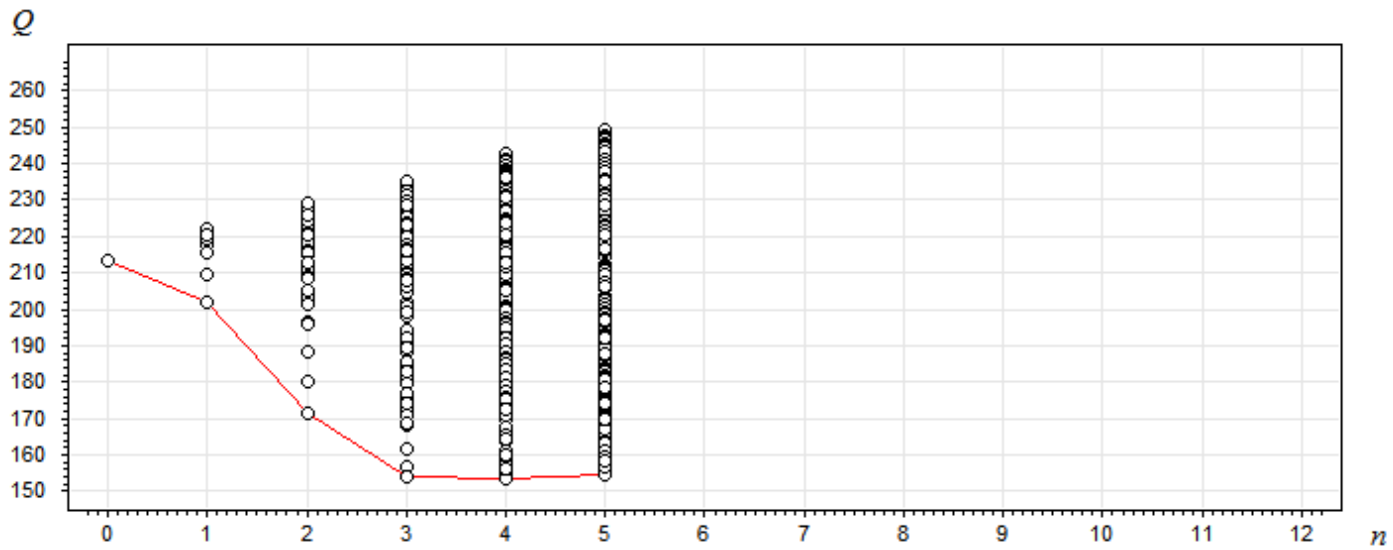
для $j = 1, \ldots, n$, где j — сложность наборов

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

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

  если $j - j^* \geqslant d$ **то выход** J_{j^*};

Алгоритм полного перебора (Full Search)



$d = 3$
$j = 5$
$j^* = 4$

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

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

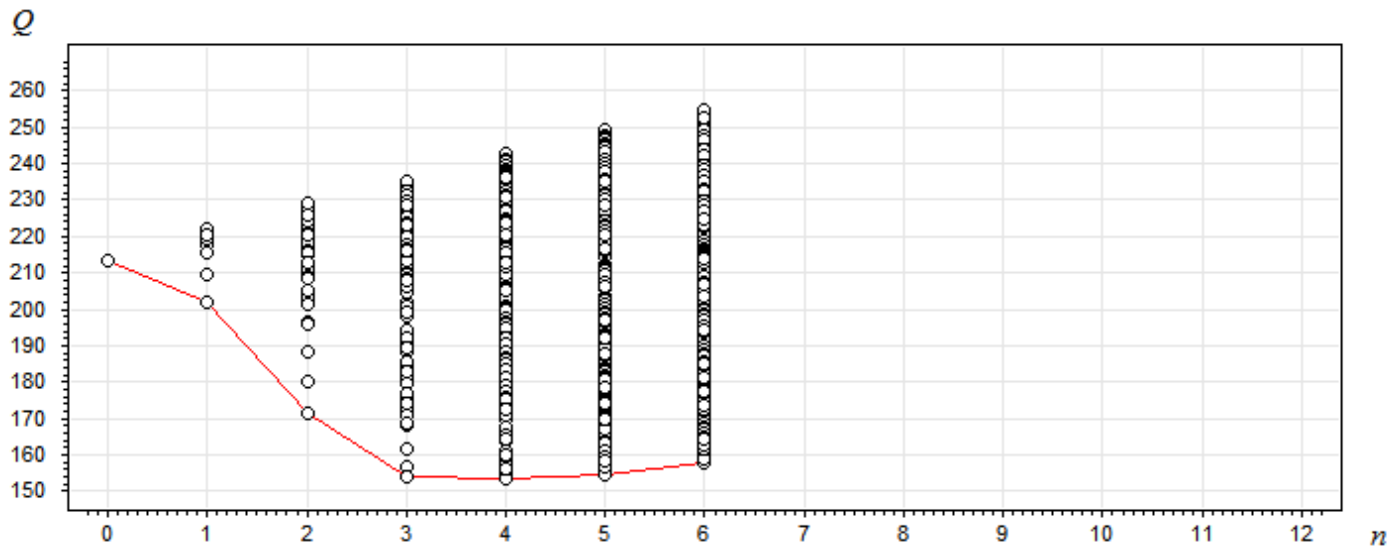
для $j = 1, \ldots, n$, где j — сложность наборов

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

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

  если $j - j^* \geqslant d$ **то выход** J_{j^*};

Алгоритм полного перебора (Full Search)



$d = 3$
$j = 6$
$j^* = 4$

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

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

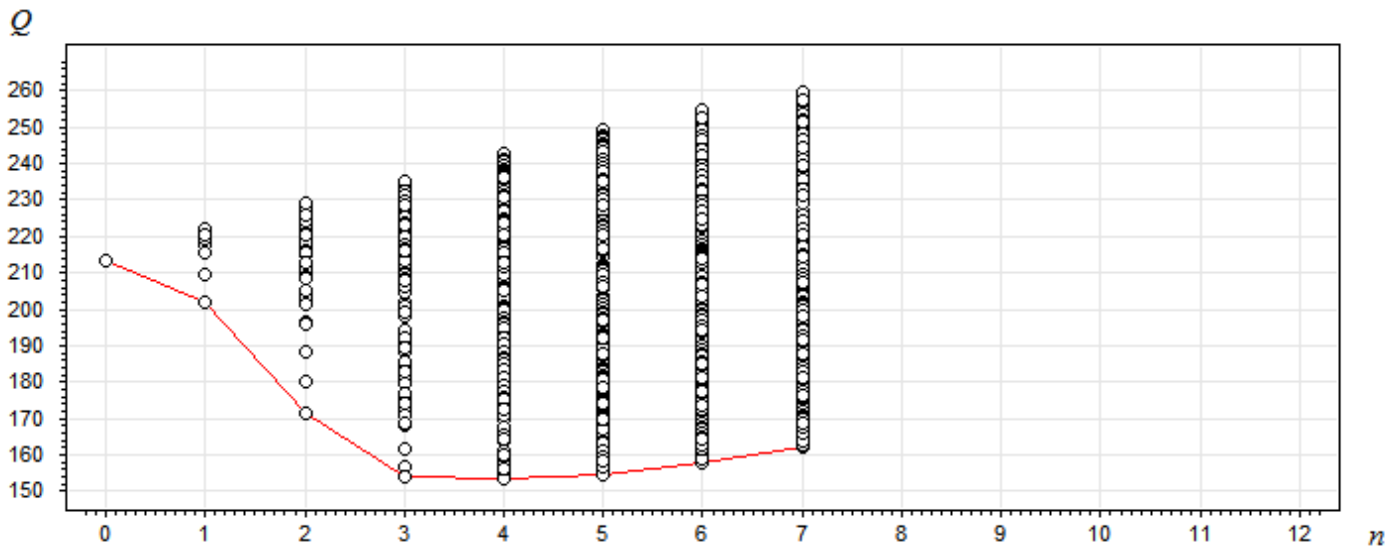
для $j = 1, \ldots, n$, где j — сложность наборов

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

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

  если $j - j^* \geqslant d$ **то выход** J_{j^*};

Алгоритм полного перебора (Full Search)



$d = 3$
$j = 7$
$j^* = 4$

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

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

для $j = 1, \ldots, n$, 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 $j - j^* \geqslant d$ **то выход** 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 ооооооочень 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: $J_0 := \varnothing$; $Q^* := Q(\varnothing)$;

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

найти признак, наиболее выгодный для добавления:

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

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 $j - j^* \geqslant d$ **то выход** 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)

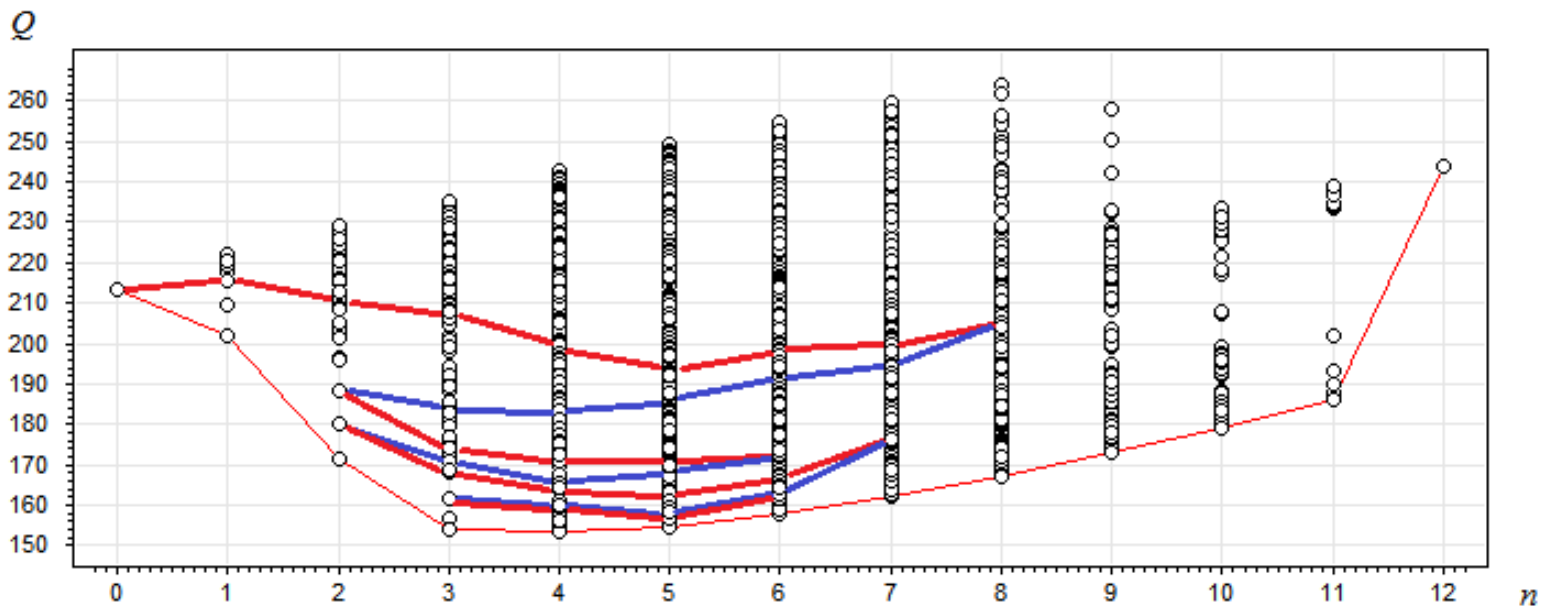
Преимущества:

— 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: $J_0 := \varnothing$; $Q^* := Q(\varnothing)$; $t := 0$;

повторять

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

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

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

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

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

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

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

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

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

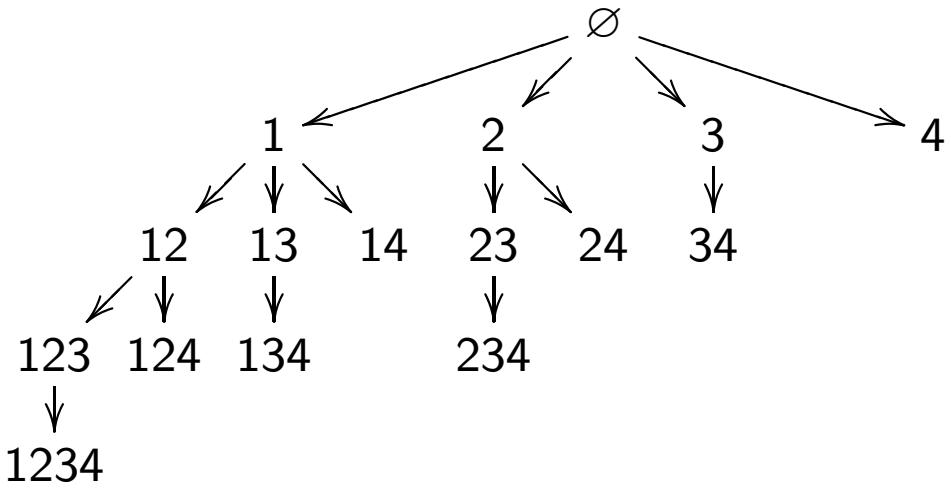
    если $t - t^* \geqslant d$ **то прервать цикл**;

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

выход 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.

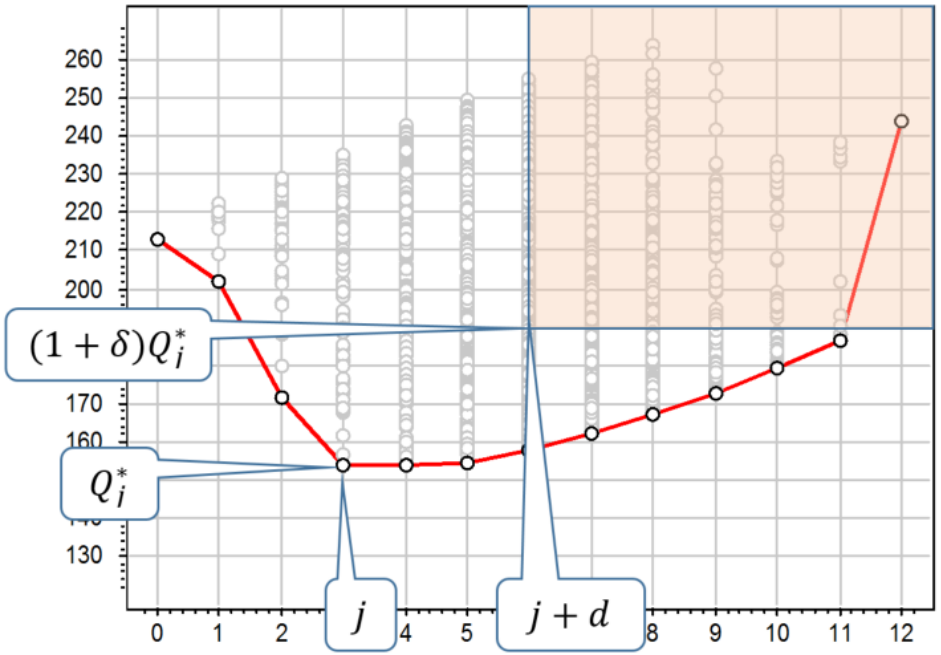
Поиск в глубину (DFS, метод ветвей и границ)

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

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

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



Чем меньше d и δ, тем сильнее сокращается перебор.

Поиск в глубину (DFS, метод ветвей и границ)

Вход: множество F, критерий Q, параметры d и δ;

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

  если найдётся j: $j \leqslant |J| - d$ и $Q(J) \geqslant (1+\delta)Q^*_j$, **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:

$Q^*_j := Q(\varnothing)$ для всех $j = 1, \ldots, n$;

упорядочить признаки по убыванию информативности;

нарастить $(\varnothing)$;

выход J, для которого $Q(J) = \min\limits_{j=1,\ldots,n} Q^*_j$;

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

Поиск в ширину (не пучком, не лучом, не «лучевой поиск»)
Он же *многорядный итерационный алгоритм МГУА*
(Метод Группового Учёта Аргументов А.Г.Ивахненко)

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*:

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

где $B_j \leqslant B$ — параметр *ширины пучка* 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:
$R_1 := \{\{f_1\}, \ldots, \{f_n\}\}$; $Q^* = Q(\varnothing)$;

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

- отсортировать ряд $R_j = \left\{J_j^1, \ldots, J_j^{B_j}\right\}$
  по возрастанию критерия: $Q\left(J_j^1\right) \leqslant \ldots \leqslant Q\left(J_j^{B_j}\right)$;
- **если** $B_j > B$ **то**
  - $R_j := \left\{J_j^1, \ldots, J_j^B\right\}$ — оставить B лучших наборов ряда;
- **если** $Q(J_j^1) < Q^*$ **то** $j^* := j$; $Q^* := Q(J_j^1)$;
- **если** $j - j^* \geqslant d$ **то выход** $J_{j^*}^1$;
- породить следующий ряд:
  $R_{j+1} := \left\{J \cup \{f\} \;\middle|\; J \in R_j,\ f \in F \setminus J\right\}$;

Усечённый поиск в ширину: дополнительные эвристики

- **Трудоёмкость:**
  $O(Bn^2)$, точнее $O(Bn(j^* + d))$.
- **Проблема дубликатов:**
  после сортировки $Q\left(J_j^1\right) \leqslant \ldots \leqslant Q\left(J_j^{B_j}\right)$ 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»);

$R_t := \left\{ J_t^1, \ldots, J_t^{B_t} \right\}$ — *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: $\beta = (\beta'_1, \ldots, \beta'_s, \beta''_{s+1}, \ldots, \beta''_n)$, $s \sim \text{uni}(1, \ldots, n)$,
  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:
$B_1 := B$; $R_1 := \left\{J_1^1, \ldots, J_1^{B_1}\right\}$; $Q^* := Q(\varnothing)$;
для $t = 1, \ldots, T$, где t — номер поколения:
- ранжирование индивидов: $Q(J_t^1) \leqslant \ldots \leqslant Q(J_t^{B_t})$;
- **если** $B_t > B$ **то** селекция: $R_t := \left\{J_t^1, \ldots, J_t^B\right\}$;
- **если** $Q(J_t^1) < Q^*$ **то** $t^* := t$; $Q^* := Q(J_t^1)$;
- **если** $t - t^* \geqslant d$ **то выход** $J_{t^*}^1$;
- породить $t+1$-е поколение путём скрещиваний и мутаций:
$R_{t+1} := \left\{\sim(J' \times J'') \;\middle|\; J', J'' \in R_t\right\} \cup R_t$;

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 $H = (h_1, \ldots, h_n)$, 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:

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

Интерпретация теоремы схемы

Число индивидов схемы H в популяции поколения t:

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

- *Строительный блок* — схема H с низким порядком $o(H)$, короткой длиной $d(H)$, высокой приспособленностью $f(H,t)$
- Если приспособленность строительного блока 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

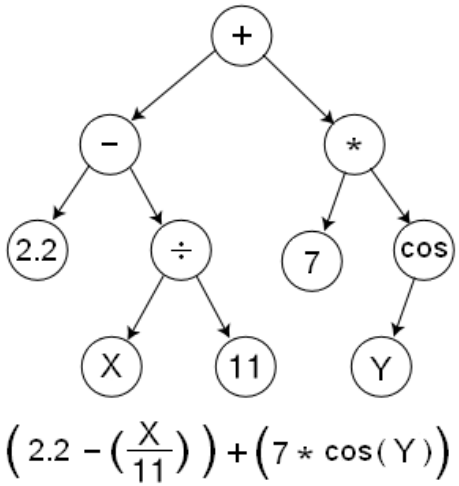
Обобщение: случайный поиск с адаптацией (СПА)

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

равные вероятности признаков: $p_1 = \cdots = p_n := 1/n$;
инициализировать случайную популяцию из B_1 наборов:
$R_1 := \left\{ J_1^1, \ldots, J_1^{B_1} \sim \{p_1, \ldots, p_n\} \right\}$; $Q^* := Q(\varnothing)$;
для $t = 1, \ldots, T$, где t — номер поколения:
- ранжирование индивидов: $Q(J_t^1) \leqslant \ldots \leqslant Q(J_t^{B_t})$;
- **если** $B_t > B$ **то** селекция: $R_t := \left\{ J_t^1, \ldots, J_t^B \right\}$;
- **если** $Q(J_t^1) < Q^*$ **то** $t^* := t$; $Q^* := Q(J_t^1)$;
- **если** $t - t^* \geqslant d$ **то выход** $J_{t^*}^1$;
- 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:
$R_{t+1} := \left\{ J_{t+1}^1, \ldots, J_{t+1}^{B_t} \sim \{p_1, \ldots, p_n\} \right\} \cup R_t$;

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

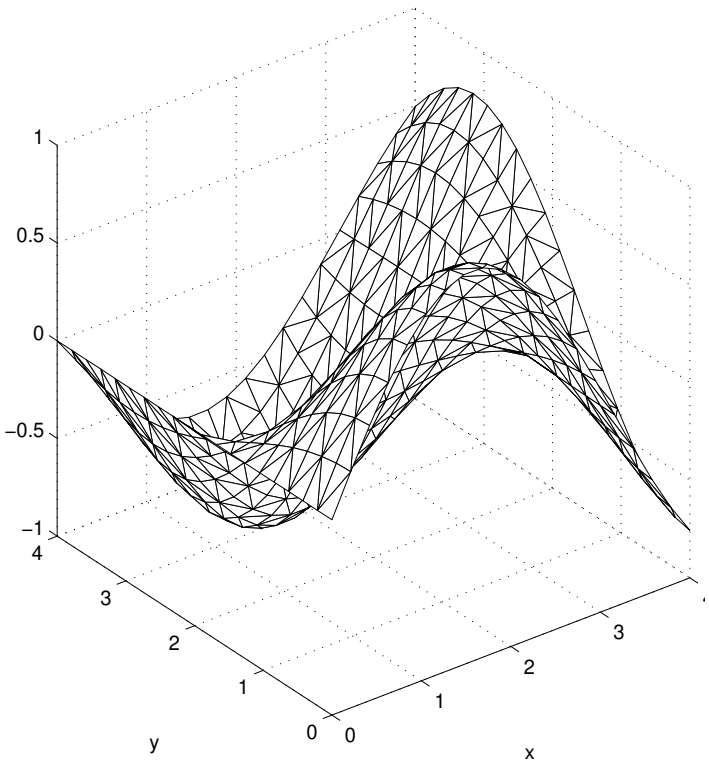


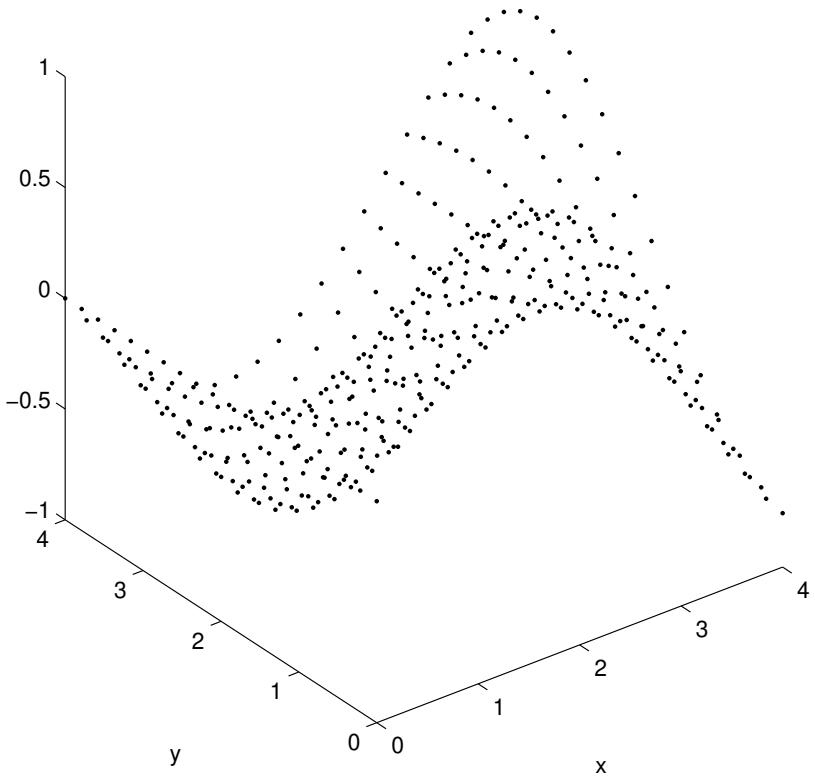
J.Koza. On the programming of computers by means of natural selection. 1992
Ying Jin et al. Bayesian Symbolic Regression. 2019
Р.А.Сологуб. Алгоритмы индуктивного порождения и трансформации моделей в задачах нелинейной регрессии. Диссертация к.ф.-м.н., 2014.

Пример. Эксперимент на синтетических данных

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

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



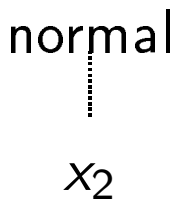


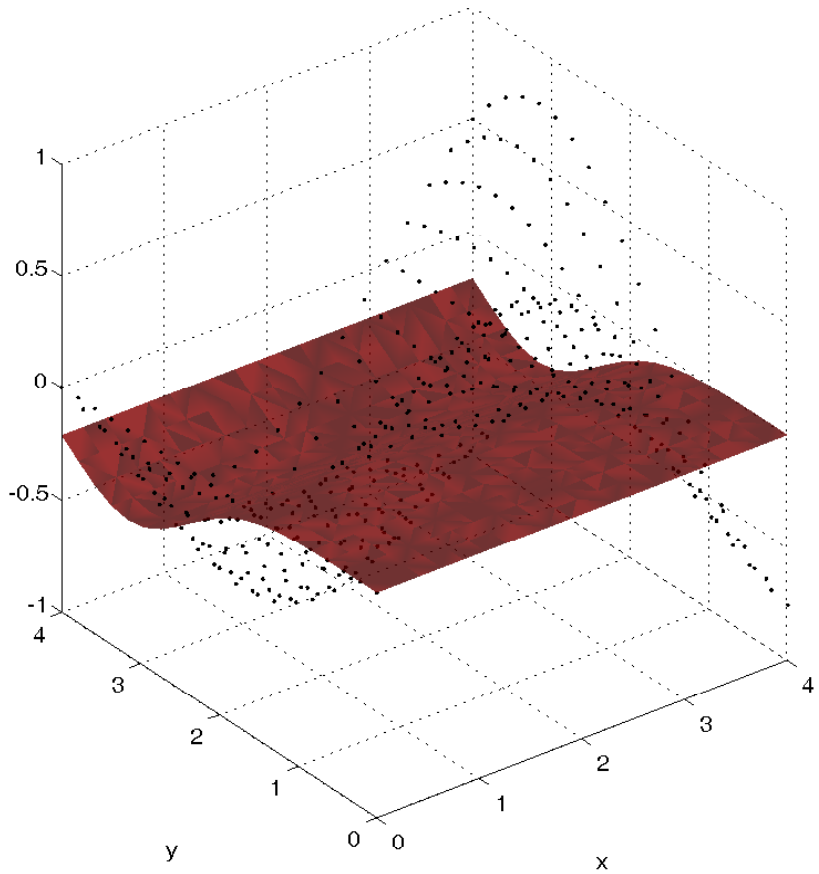
Пример. Система порождающих функций

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	$y = \frac{\lambda}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\xi)^2}{2\sigma^2}\right) + a$	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

Пример. Перебор моделей-претендентов

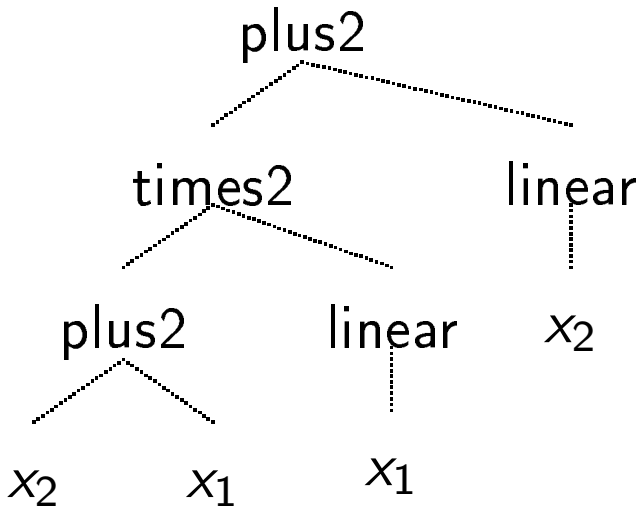
$\text{normal}(w_{1:3}, x_2)$

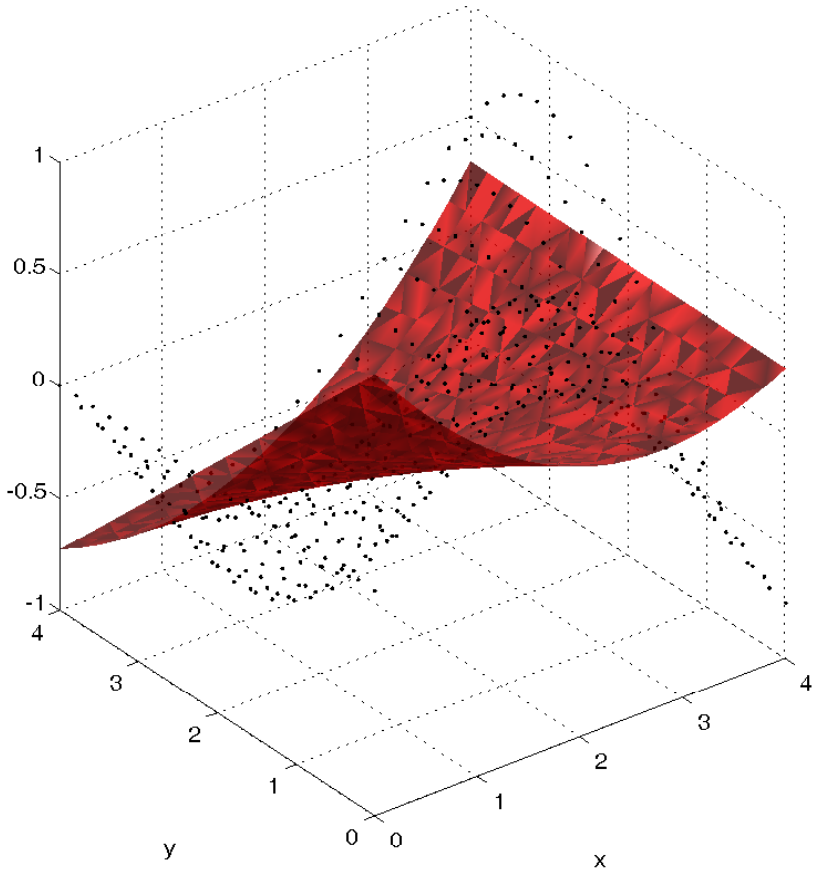




Пример. Перебор моделей-претендентов

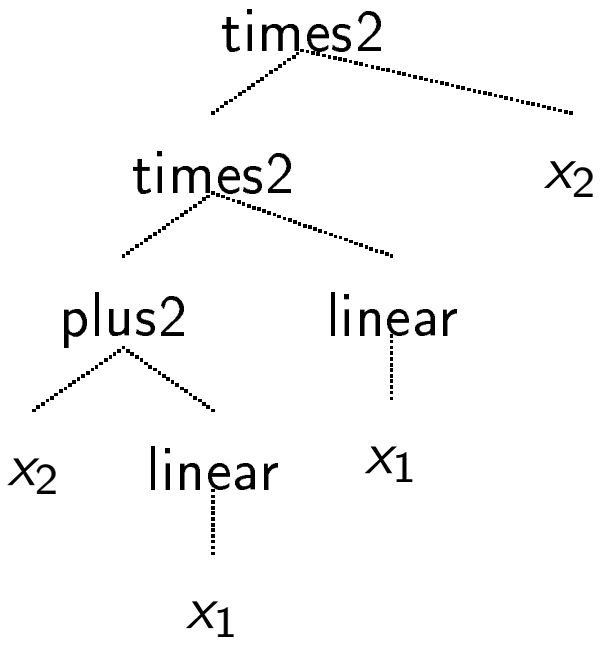
$$\text{plus2}(\varnothing,\text{times2}(\varnothing,\text{plus2}(\varnothing,x_2,x_1),\text{linear}(w_{1:2},x_1)),\text{linear}(w_{3:4},x_2))$$

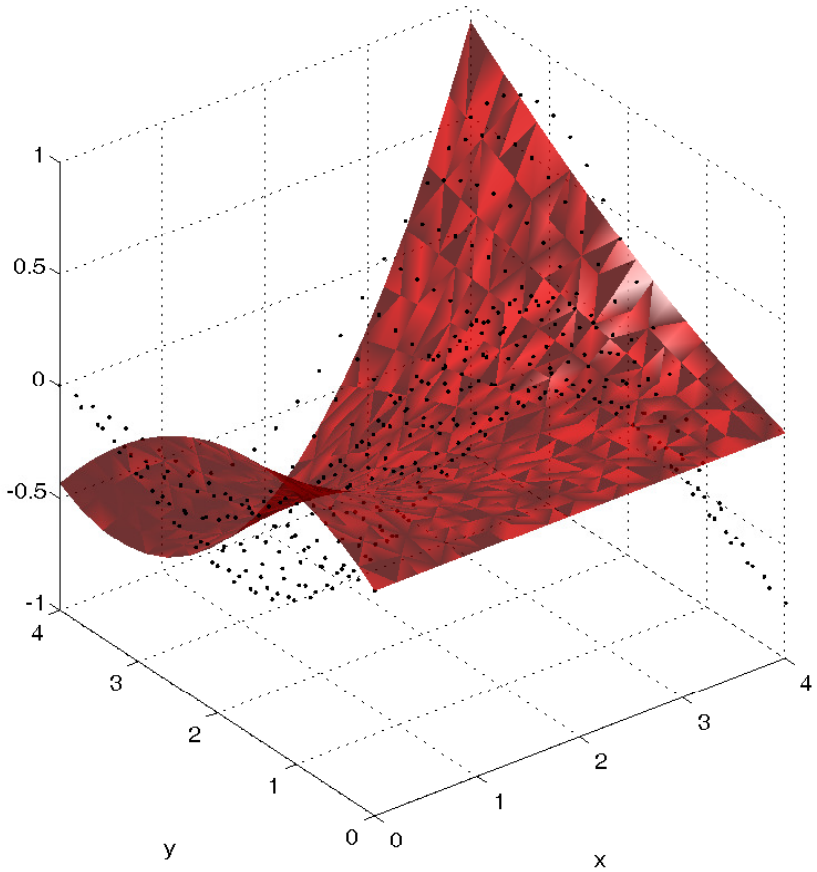




Пример. Перебор моделей-претендентов

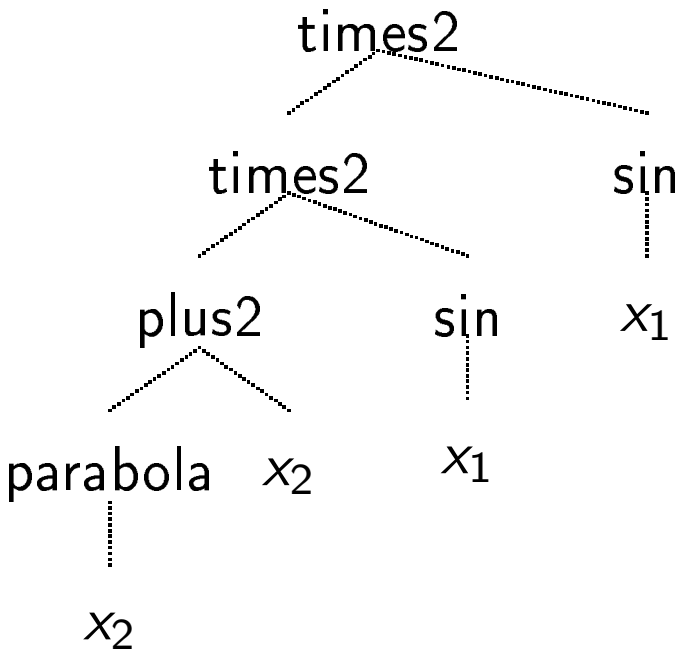
$$\text{times2}(\varnothing,\text{times2}(\varnothing,\text{plus2}(\varnothing,x_2,\text{linear}(w_{1:2},x_1)),\text{linear}(w_{3:4},x_1)),x_2)$$

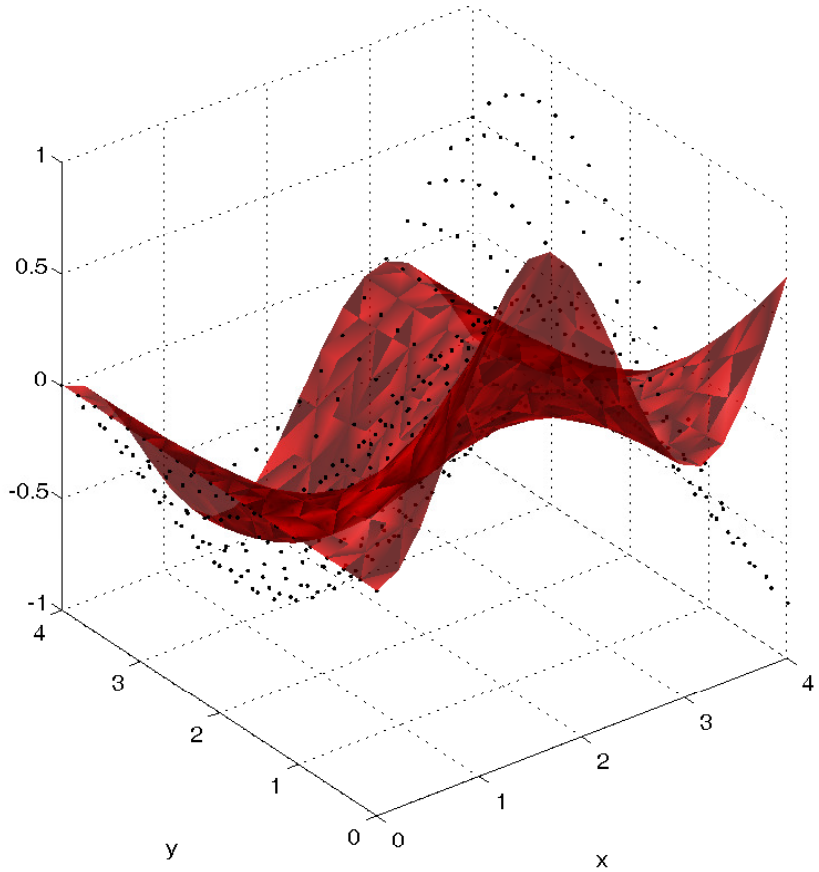




Пример. Перебор моделей-претендентов

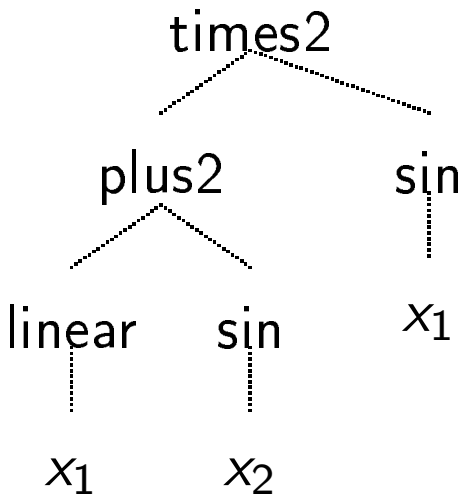
$$\text{times2}(\varnothing,\text{times2}(\varnothing,\text{plus2}(\varnothing,\text{parabola}(w_{1:3},x_2),x_2),\sin(\varnothing,x_1)),\sin(\varnothing,x_1))$$

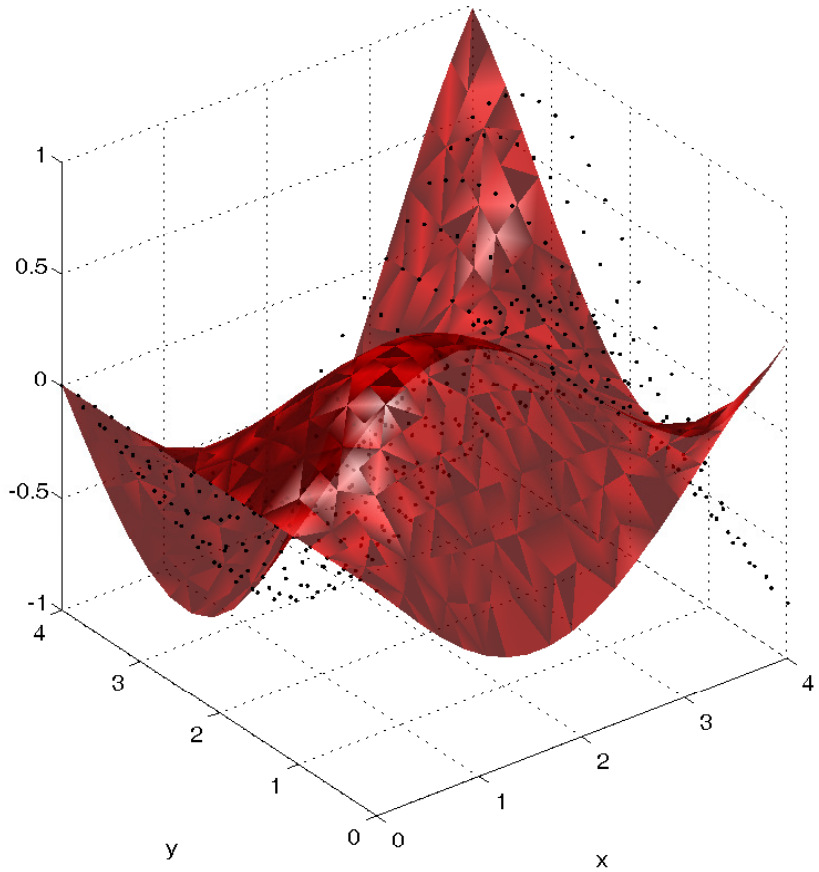




Пример. Перебор моделей-претендентов

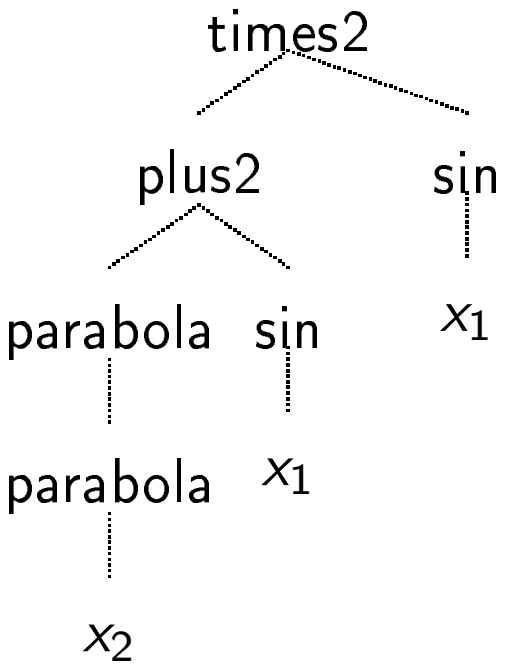
$$\text{times2}(\varnothing,\text{plus2}(\varnothing,\text{linear}(w_{1:2},x_1),\sin(\varnothing,x_2)),\sin(\varnothing,x_1))$$

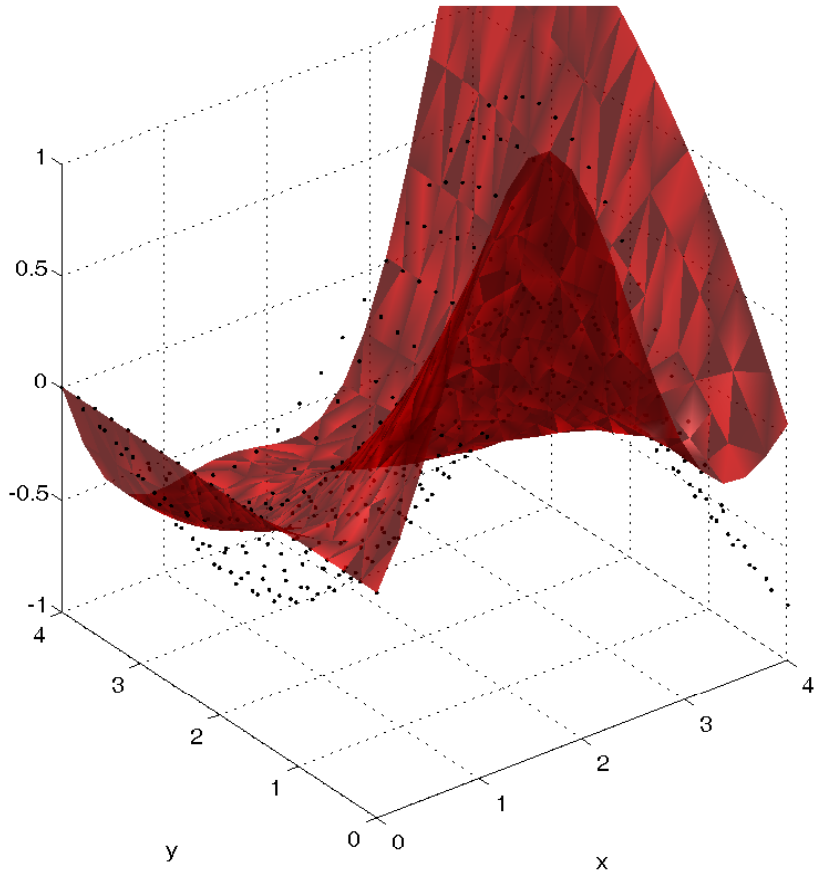




Пример. Перебор моделей-претендентов

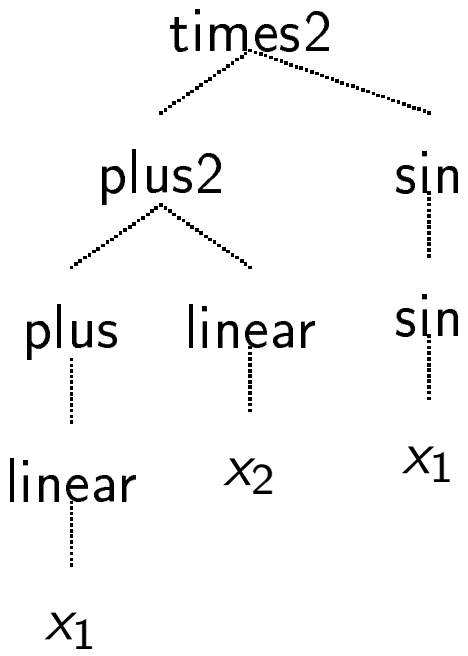
$$\text{times2}(\varnothing,\text{plus2}(\varnothing,\text{parabola}(w_{1:3},\text{parabola}(w_{4:6},x_2)),\sin(\varnothing,x_1)),\sin(\varnothing,x_1))$$

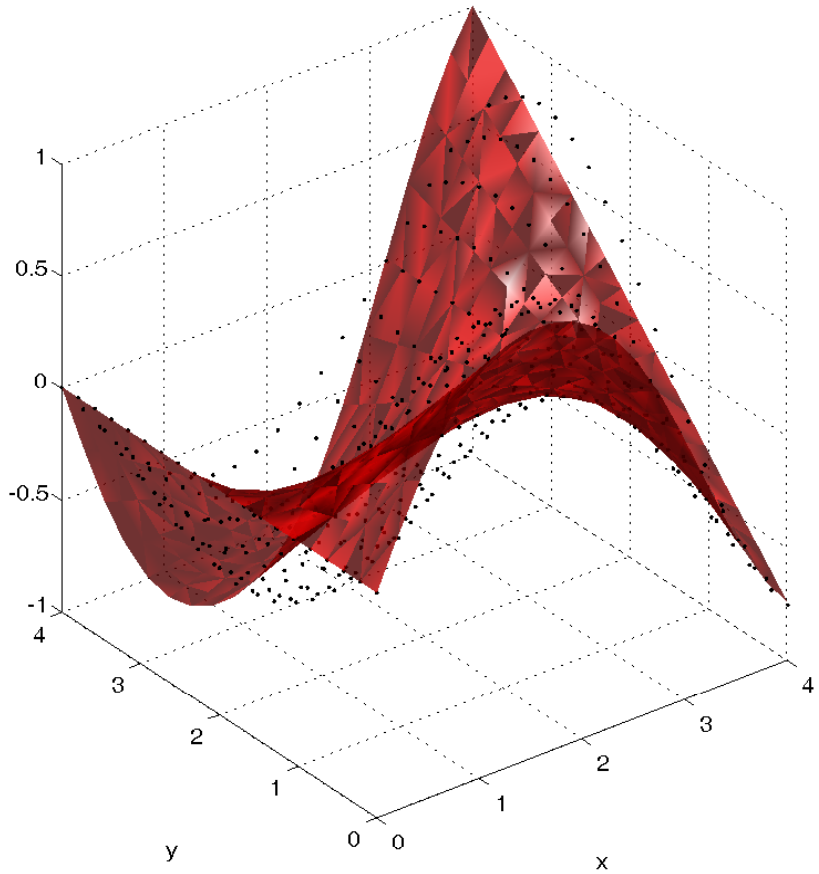




Пример. Перебор моделей-претендентов

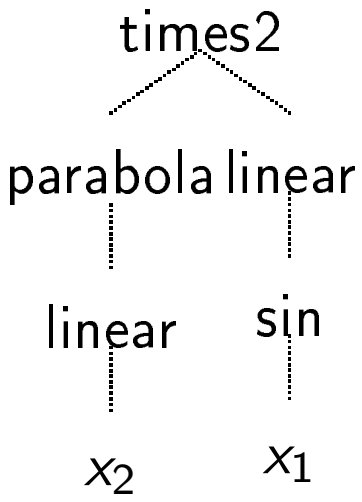
$$\text{times2}(\varnothing,\text{plus2}(\varnothing,\text{plus}(w_1,\text{linear}(w_{2:3},x_1)),\text{linear}(w_{4:5},x_2)),\sin(\varnothing,\sin(\varnothing,x_1))$$

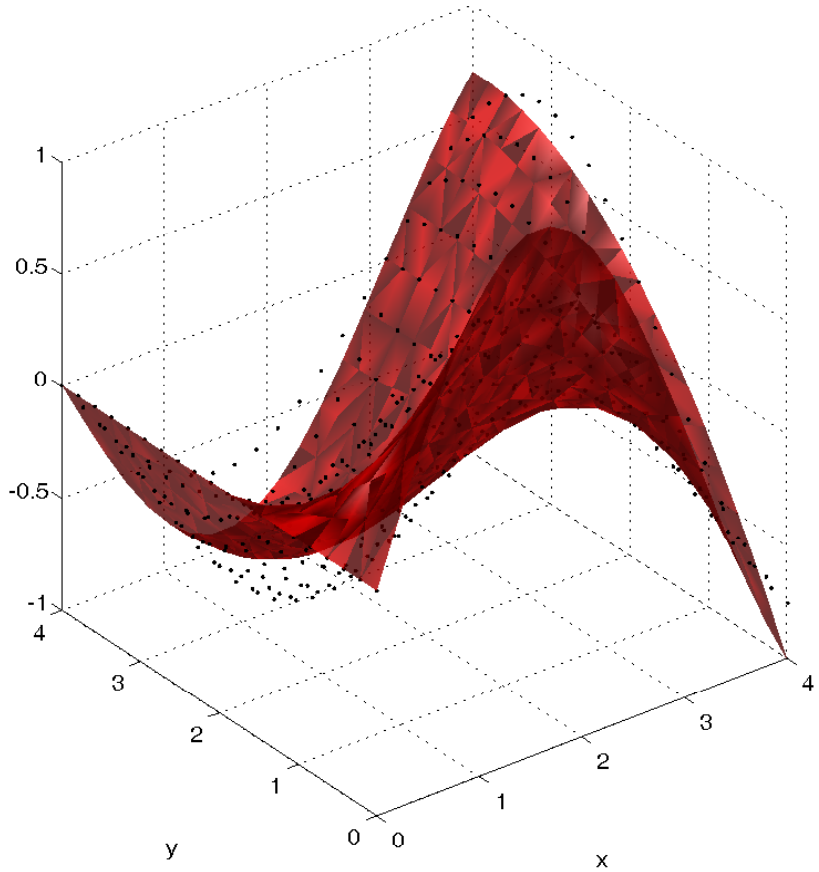




Пример. Перебор моделей-претендентов

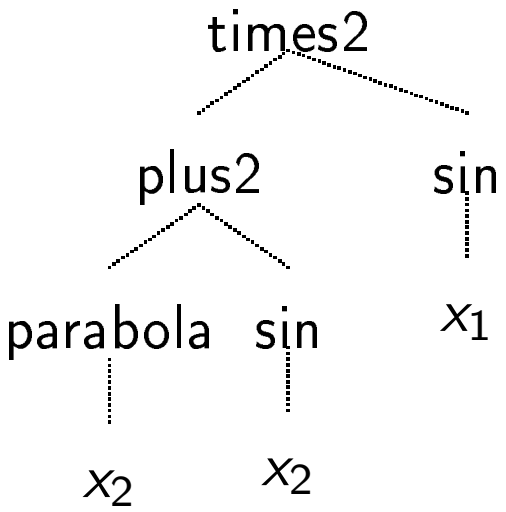
$$\text{times2}(\varnothing, \text{parabola}(w_{1:3}, \text{linear}(w_{4:5}, x_2)), \text{linear}(w_{6:7}, \sin(\varnothing, x_1)))$$

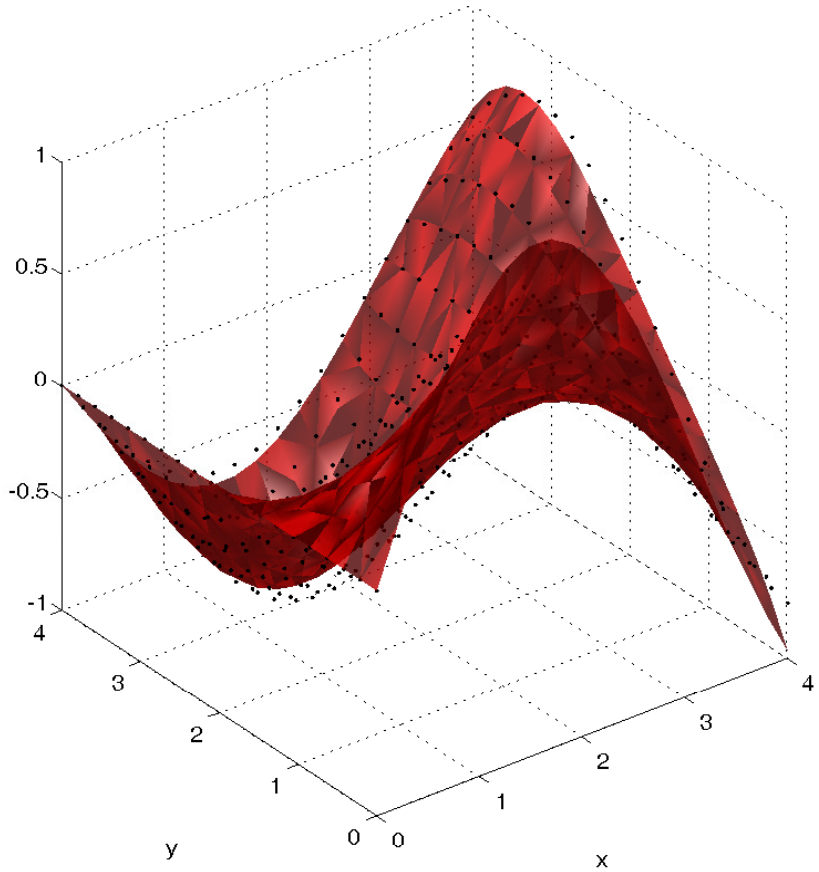




Пример. Перебор моделей-претендентов

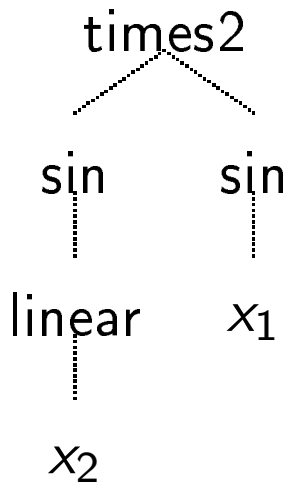
$$\text{times2}(\varnothing,\text{plus2}(\varnothing,\text{parabola}(w_{1:3},x_2),\sin(\varnothing,x_2)),\sin(\varnothing,x_1))$$

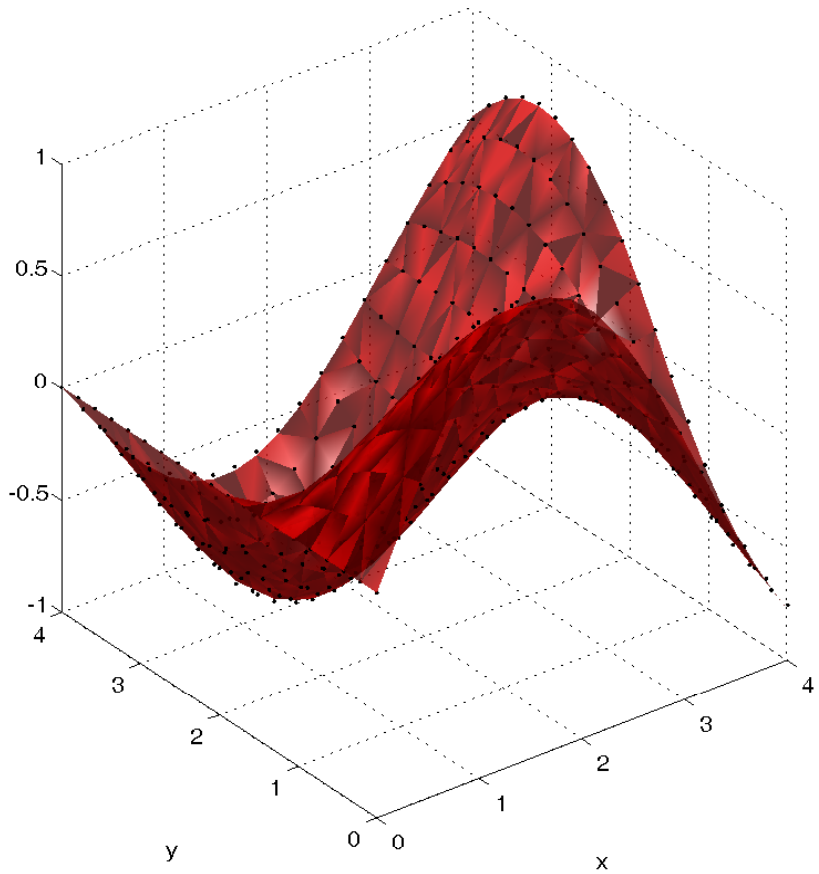




Пример. Перебор моделей-претендентов

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





Резюме. Методы отбора признаков и моделей

- Для отбора признаков могут использоваться любые эвристические методы дискретной оптимизации

$$Q(J) \to \min_{J \subseteq F}.$$

- $Q(J)$ 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