# Теория обобщающей способности, критерии выбора моделей и методы отбора признаков

К. В. Воронцов
vokov@forecsys.ru

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

сентябрь 2011

# Содержание

## Задачи выбора метода обучения

**Дано:** $X$ — пространство объектов; $Y$ — множество ответов;
$X^\ell = (x_i, y_i)_{i=1}^\ell$ — обучающая выборка, $y_i = y^*(x_i)$;
$A_t = \{a\colon X \to Y\}$ — модели алгоритмов, $t \in T$;
$\mu_t\colon (X \times Y)^\ell \to A_t$ — 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 (features selection): $\mathscr{F} = \bigl\{f_j\colon X \to D_j\colon j = 1, \dots, n\bigr\}$ — множество признаков; метод обучения $\mu_{\mathscr{G}}$ использует только признаки $\mathscr{G} \subseteq \mathscr{F}$.

## Как оценить качество обучения по прецедентам?

$\mathscr{L}(a,x)$ — функция потерь алгоритма $a$ на объекте $x$;

$Q(a,X^\ell) = \frac{1}{\ell}\sum\limits_{i=1}^{\ell}\mathscr{L}(a,x_i)$ — *функционал качества* $a$ на $X^\ell$.

*Внутренний критерий* оценивает качество на обучении $X^\ell$:

$$Q_\mu(X^\ell) = Q\bigl(\mu(X^\ell), X^\ell\bigr).$$

**Недостаток:** эта оценка смещена, т.к. $\mu$ минимизирует её же.

*Внешний критерий* оценивает качество «вне обучения», например, по отложенной (hold-out) контрольной выборке $X^k$:

$$Q_\mu(X^\ell, X^k) = Q\bigl(\mu(X^\ell), X^k\bigr).$$

**Недостаток:** эта оценка зависит от разбиения $X^L = X^\ell \sqcup X^k$.

## Скользящий контроль (cross-validation, CV)

Оценка *полного скользящего контроля* (complete CV) — усреднение hold-out по всем $C_L^\ell$ разбиениям $X^L = X^\ell \sqcup X^k$:

$$\mathrm{CCV}(\mu, X^L) = \frac{1}{C_L^\ell} \sum_{X^\ell \subset X^L} Q_\mu(X^\ell, X^k) = \mathsf{E} Q_\mu(X^\ell, X^k),$$

если полагать, что все разбиения равновероятны.

**Недостаток 1:** эта оценка вычислительно слишком сложна. **Устраняется** либо *аналитическим оцениванием*, либо *эмпирическим оцениванием* по подмножеству разбиений.

**Недостаток 2:** эта оценка не учитывает дисперсию hold-out. **Устраняется** *оцениванием распределения* значений hold-out:

$$R_\varepsilon(\mu, X^L) = \mathsf{P}\bigl[Q_\mu(X^\ell, X^k) \geqslant \varepsilon\bigr] \quad \text{или}$$

$$Q_\varepsilon(\mu, X^L) = \mathsf{P}\bigl[Q_\mu(X^\ell, X^k) - Q_\mu(X^\ell) \geqslant \varepsilon\bigr].$$

## Эмпирические оценки скользящего контроля

**Контроль по отдельным объектам** (leave one out CV): $k = 1$,

$$\mathrm{LOO}(\mu, X^L) = \frac{1}{L} \sum_{i=1}^{L} Q_\mu(X^L \setminus \{x_i\}, \{x_i\}).$$

**Недостатки** LOO: ресурсоёмкость, высокая дисперсия.

**Контроль по $q$ блокам** ($q$-fold CV): случайное разбиение $X^L = X_1^{\ell_1} \sqcup \dots \sqcup X_q^{\ell_q}$ на $q$ блоков (почти) равной длины,

$$\widehat{\mathrm{CV}}_q(\mu, X^L) = \frac{1}{q} \sum_{n=1}^{q} Q_\mu\bigl(X^L \setminus X_n^{\ell_n}, X_n^{\ell_n}\bigr).$$

**Недостатки** $q$-fold CV:
- оценка существенно зависит от разбиения на блоки;
- каждый объект лишь один раз участвует в контроле.

## Эмпирические оценки скользящего контроля

**Контроль $t$ раз по $q$ блокам** ($t{\times}q$-fold CV) — стандарт «де факто» для тестирования методов обучения.

Выборка $X^L$ разбивается $t$ раз случайным образом на $q$ блоков
$X^L = X_{s1}^{\ell_1} \sqcup \dots \sqcup X_{sq}^{\ell_q}, \quad s = 1, \dots, t, \quad \ell_1 + \dots + \ell_q = L;$

$$\widehat{\mathrm{CV}}_{t\times q}(\mu, X^L) = \frac{1}{t} \sum_{s=1}^{t} \frac{1}{q} \sum_{n=1}^{q} Q_\mu\bigl(X^L \backslash X_{sn}^{\ell_n}, X_{sn}^{\ell_n}\bigr).$$

**Преимущества** $t{\times}q$-fold CV:
- увеличением $t$ можно улучшать точность оценки (компромисс между точностью и временем вычислений);
- каждый объект участвует в контроле ровно $t$ раз;
- можно вычислять доверительные интервалы (при $t \gtrsim 20$).

## Критерии непротиворечивости моделей

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

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

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

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

$$D_t(\mu, X^L) = \frac{1}{t} \sum_{s=1}^{t} \frac{1}{L} \sum_{i=1}^{L} \bigl|\mu(X^\ell_s)(x_i) - \mu(X^k_s)(x_i)\bigr|.$$

**Недостатки:**

- длина обучения сокращается в 2 раза;
- трудоёмкость возрастает в 2 раза.

# Аналитические оценки и их обращение

**Основная идея аналитического подхода:**

1. Получить верхнюю оценку $Q_\varepsilon$, справедливую для любой выборки $X^L$ и широкого класса методов обучения $\mu \in M$:

$$Q_\varepsilon(\mu, X^L) = \mathsf{P}\Bigl[Q_\mu(X^\ell, X^k) - Q_\mu(X^\ell) \geqslant \varepsilon\Bigr] \leqslant \eta(\varepsilon).$$

2. Тогда для любой $X^L$, любого $\mu \in M$ и любого $\eta \in (0,1)$ с вероятностью не менее $(1-\eta)$ справедлива оценка

$$Q_\mu(X^\ell, X^k) \leqslant Q_\mu(X^\ell) + \varepsilon(\eta),$$

где $\varepsilon(\eta)$ — функция, обратная к $\eta(\varepsilon)$, зависящая от $A$ и $\mu$, но не зависящая от скрытой контрольной выборки $X^k$.

3. Оптимизировать метод обучения: $Q_\mu(X^\ell) + \varepsilon(\eta) \to \min\limits_{\mu \in M}$.

## 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:** $A = \bigl\{a(x) = \operatorname{sign}\langle w, x\rangle\bigr\}$ — классификация,

$$A = \bigl\{a(x) = \langle w, x\rangle\bigr\} \text{ — регрессия.}$$

$L_2$-*регуляризация* (ридж-регрессия, weight decay):

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

$L_1$-*регуляризация* (LASSO):

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

$L_0$-*регуляризация* (AIC, BIC):

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

# Разновидности $L_0$-регуляризации

Информационный критерий Акаике (Akaike Information Criterion):

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

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

Байесовский информационный критерий (Bayes Inform. Criterion):

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

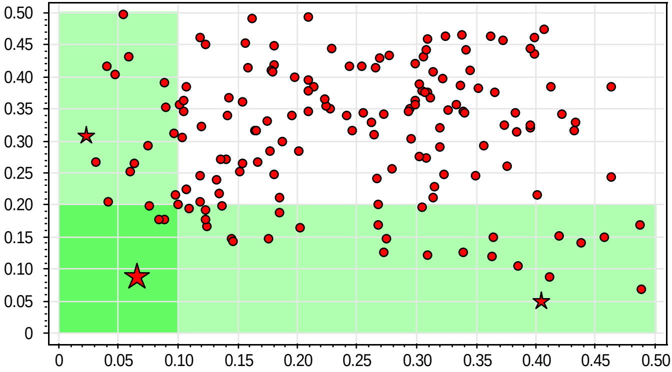
Оценка Вапника-Червоненкиса (VC-bound):

$$\mathrm{VC}(\mu, X^\ell) = Q_\mu(X^\ell) + \sqrt{\frac{h}{\ell}\left(\ln\frac{2\ell}{h} + 1\right) - \frac{\ln\eta}{\ell}},$$

$h$ — VC-размерность; для линейных, опять-таки, $h = |\mathscr{G}|$;
$\eta$ — уровень значимости; обычно $\eta = 0.05$.

## Выбор модели по совокупности внешних критериев

Модель, немного неоптимальная по обоим критериям, скорее всего, лучше, чем модель, оптимальная по одному критерию, но не оптимальная по другому.



## Резюме

- Обобщающая способность может быть формализована с помощью функционалов, построенных по принципу полного скользящего контроля (CCV).
- Эмпирические оценки CCV просто реализуются, но требуют многократного применения $\mu$, следовательно, длительных вычислений.
- Аналитические оценки CCV не требуют многократного применения $\mu$, но могут быть сильно завышенными или иметь узкие границы применимости.

# Бинарная функция потерь. Матрица ошибок

$X^L = \{x_1, \ldots, x_L\}$ — конечное *генеральное множество* объектов;
$A = \{a_1, \ldots, a_D\}$ — конечное семейство *алгоритмов*;
$\mathscr{L}(a, x) \equiv I(a, x) = [$алгоритм $a$ ошибается на объекте $x]$;

$L{\times}D$-*матрица ошибок* с попарно различными столбцами:

| | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ | $\cdots$ | $a_D$ | |
|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | 1 | 1 | 0 | 0 | 0 | 1 | $\cdots$ | 1 | $X^\ell$ — наблюдаемая |
| $\ldots$ | 0 | 0 | 0 | 0 | 1 | 1 | $\cdots$ | 1 | (обучающая) выборка |
| $x_\ell$ | 0 | 0 | 1 | 0 | 0 | 0 | $\cdots$ | 0 | длины $\ell$ |
| $x_{\ell+1}$ | 0 | 0 | 0 | 1 | 1 | 1 | $\cdots$ | 0 | $X^k$ — скрытая |
| $\ldots$ | 0 | 0 | 0 | 1 | 0 | 0 | $\cdots$ | 1 | (контрольная) выборка |
| $x_L$ | 0 | 1 | 1 | 1 | 1 | 1 | $\cdots$ | 0 | длины $k = L - \ell$ |

$n(a, X) = \sum\limits_{x \in X} I(a, x)$ — *число ошибок* $a \in A$ на выборке $X \subset X^L$;

$\nu(a, X) = n(a, X)/|X|$ — *частота ошибок* $a$ на выборке $X$;

## Задача оценивания вероятности переобучения

**Основное вероятностное предположение:**
все разбиения $X^\ell \sqcup X^k = X^L$ равновероятны
(слабый вариант гипотезы независимости выборки $X^L$).

*Переобученность* — разность частот ошибок на $X^k$ и на $X^\ell$:
$$\delta(\mu, X^\ell) = \nu\bigl(\mu(X^\ell), X^k\bigr) - \nu\bigl(\mu(X^\ell), X^\ell\bigr).$$
*Переобучение* — это событие $\delta(\mu, X^\ell) \geqslant \varepsilon$.

**Основная задача** — оценить *вероятность переобучения*:
$$Q_\varepsilon(\mu, X^L) = \mathsf{P}\bigl[\delta(\mu, X^\ell) \geqslant \varepsilon\bigr].$$

## Простейший, но важный частный случай

Пусть $A = \{a\}$ — одноэлементное множество, $m = n(a, X^L)$.

Тогда вероятность переобучения есть вероятность большого отклонения частот ошибок в двух подвыборках:

$$Q_\varepsilon(a, X^L) = \mathsf{P}\bigl[\nu(a, X^k) - \nu(a, X^\ell) \geqslant \varepsilon\bigr].$$

### Теорема

Для любого $X^L$, любого $\varepsilon \in [0, 1]$

$$Q_\varepsilon(a, X^L) = \mathcal{H}_L^{\ell, m}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right),$$

где $\mathcal{H}_L^{\ell, m}(z) = \sum\limits_{s=0}^{\lfloor z \rfloor} \dfrac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell}$ — функция гипергеометрического распределения.

## Доказательство

1. Обозначим $s = n(a, X^\ell)$.

2. «Школьная» задача по теории вероятностей:
в урне $L$ шаров, $m$ из них чёрные; извлекаем $\ell$ шаров наугад.
Какова вероятность того, что $s$ из них чёрные?

$$\mathsf{P}\bigl[n(a, X^\ell) = s\bigr] = C_m^s C_{L-m}^{\ell-s} / C_L^\ell.$$

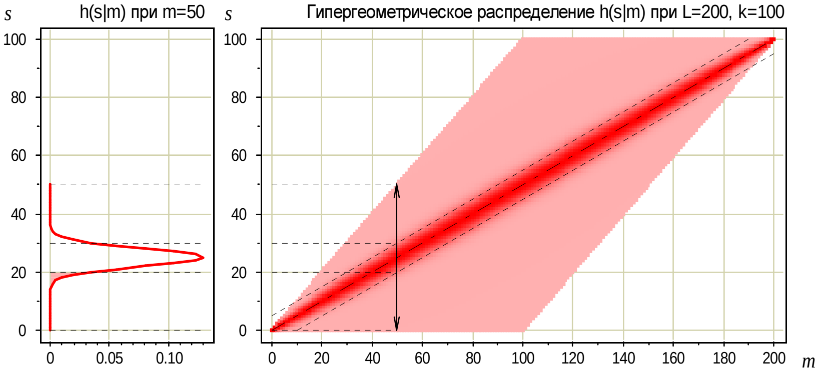
3. Распишем $Q_\varepsilon$, подставив $\nu(a, X^k) = \frac{m-s}{k}$, $\nu(a, X^\ell) = \frac{s}{\ell}$:

$$Q_\varepsilon(a, X^L) = \mathsf{P}\bigl[\nu(a, X^k) - \nu(a, X^\ell) \geqslant \varepsilon\bigr] =$$

$$= \sum_{s=0}^{\ell} \underbrace{\Bigl[\, \frac{m-s}{k} - \frac{s}{\ell} \geqslant \varepsilon \,\Bigr]}_{s \leqslant \frac{\ell}{L}(m-\varepsilon k)} \underbrace{\mathsf{P}\bigl[n(a, X^\ell) = s\bigr]}_{C_m^s C_{L-m}^{\ell-s} / C_L^\ell} =$$

$$= \mathcal{H}_L^{\ell,\,m}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right). \qquad \blacksquare$$

# Гипергеометрическое распределение $h(s|m) = C_m^s C_{L-m}^{\ell-s} / C_L^\ell$




Предсказание числа $m = n(a, X^L)$ по числу $s = n(a, X^\ell)$ возможно благодаря узости гипергеометрического пика, причём при $\ell, k \to \infty$ он сужается, и $\nu(a, X^\ell) \to \nu(a, X^k)$ (явление *концентрации вероятности*, закон больших чисел).

## Теория Вапника–Червоненкиса

Рассмотрим общий случай — $A$ произвольное, конечное.

1. Вероятность переобучения оценим сверху вероятностью большого *равномерного отклонения* частот: для любых $X^L$, $\mu$

$$Q_\varepsilon(\mu, X^L) = \mathsf{P}\bigl[\delta(\mu, X^\ell) \geqslant \varepsilon\bigr] \leqslant$$
$$\leqslant \mathsf{P}\Bigl[\max_{a\in A} \delta(a, X^\ell) \geqslant \varepsilon\Bigr] = \widetilde{Q}_\varepsilon(A, X^L).$$

2. Оценим вероятность объединения событий суммой их вероятностей (неравенство Буля, union bound):

$$\widetilde{Q}_\varepsilon(A, X^L) = \mathsf{P}\max_{a\in A}\bigl[\delta(a, X^\ell) \geqslant \varepsilon\bigr] \leqslant$$
$$\leqslant \mathsf{P}\sum_{a\in A}\bigl[\delta(a, X^\ell) \geqslant \varepsilon\bigr] = \sum_{a\in A} \underbrace{\mathsf{P}\bigl[\delta(a, X^\ell) \geqslant \varepsilon\bigr]}_{Q_\varepsilon(a, X^L)}.$$

## Теория Вапника–Червоненкиса

Таким образом, доказали важную теорему:

### Теорема

Для любых $X^L$, $\mu$, конечного $A$ и $\varepsilon \in [0,1]$

$$\widetilde{Q}_\varepsilon(A, X^L) \leqslant \sum_{a \in A} \mathcal{H}_L^{\ell,\, m}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right),$$

где $m = n(a, X^L)$.

### Следствие (оценка Вапника–Червоненкиса, 1974)

Для любых $X^L$, $\mu$, конечного $A$ и $\varepsilon \in [0,1]$

$$\widetilde{Q}_\varepsilon(A, X^L) \leqslant |A| \cdot \max_m \mathcal{H}_L^{\ell,\, m}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right) \leqslant$$
$$\leqslant |A| \cdot \tfrac{3}{2} \exp\bigl(-\varepsilon^2 \ell\bigr), \quad \text{при } \ell = k.$$

## Обобщение на случай бесконечных семейств $A$

*Функция роста* $\Delta^A(L)$ семейства $A$ — это максимальное по $X^L$ число различных векторов ошибок $\vec{a} = \bigl(I(a,x_1),\dots,I(a,x_L)\bigr)$.
В оценке надо заменить $|A|$ на функцию роста $\Delta^A(L)$.

*Ёмкость* (размерность Вапника-Червоненкиса) семейства $A$ — это максимальная длина выборки $h$, для которой $\Delta^A(h) = 2^h$.

**Теорема**

Если такое $h$ существует, то $\Delta^A(L) \leqslant C_L^0 + \dots + C_L^h \leqslant \frac{3}{2}\frac{L^h}{h!}$.

**Теорема**

Ёмкость семейства линейных классификаторов на два класса

$$a(x) = \operatorname{sign}(w_1 x^1 + \dots + w_n x^n), \quad x = (x^1,\dots,x^n) \in X.$$

равна размерности пространства параметров, $\operatorname{VCdim}(A) = n$.

## Обращение оценки Вапника-Червоненкиса

1. Оценка: $\mathsf{P}\Bigl[\max\limits_{a\in A}\bigl(\nu(a,X^k)-\nu(a,X^\ell)\bigr)\geqslant\varepsilon\Bigr]\leqslant\Delta\,\frac{3}{2}\exp\bigl(-\ell\varepsilon^2\bigr)$.

Тогда для любого $a\in A$ с вероятностью не менее $(1-\eta)$

$$\nu(a,X^k)\leqslant\underbrace{\nu(a,X^\ell)}_{\text{эмпирический риск}}+\underbrace{\sqrt{\frac{1}{\ell}\ln\Delta+\frac{1}{\ell}\ln\frac{2}{3\eta}}}_{\text{штраф за сложность}}.$$

2. Оценка: $\mathsf{P}\Bigl[\max\limits_{a\in A}\bigl(\nu(a,X^k)-\nu(a,X^\ell)\bigr)\geqslant\varepsilon\Bigr]\leqslant\frac{3}{2}\frac{L^h}{h!}\cdot\frac{3}{2}\exp\bigl(-\ell\varepsilon^2\bigr)$.

Тогда для любого $a\in A$ с вероятностью не менее $(1-\eta)$

$$\nu(a,X^k)\leqslant\underbrace{\nu(a,X^\ell)}_{\text{эмпирический риск}}+\underbrace{\sqrt{\frac{h}{\ell}\ln\left(\frac{2e\ell}{h}\right)+\frac{1}{\ell}\ln\frac{4}{9\eta}}}_{\text{штраф за сложность}}.$$

## Метод структурной минимизации риска (СМР)

**Дано:** система вложенных подсемейств возрастающей ёмкости

$$A_0 \subset A_1 \subset \cdots \subset A_h \subset \cdots$$

**Найти:** оптимальную ёмкость $h^*$, такую, что

$$\nu(a, X^k) \leqslant \underbrace{\min_{a \in A_h} \nu(a, X^\ell)}_{\substack{\text{минимизация} \\ \text{эмпирического риска}}} + \underbrace{\sqrt{\frac{h}{\ell} \ln\left(\frac{2e\ell}{h}\right) + \frac{1}{\ell} \ln \frac{4}{9\eta}}}_{\text{штраф за сложность}} \to \min_h$$

**Недостатки СМР:**

- $h^*$ может оказаться заниженной из-за завышенности $Q_\varepsilon$.
- На практике эмпирический CV предпочтительнее этих оценок.

## Причины завышенности оценок Вапника-Червоненкиса

- **Оценка равномерного отклонения** сильно завышена, когда бóльшая часть алгоритмов имеет исчезающе малую вероятность быть результатом обучения.

  На практике распределение

  $$q(a) = \mathsf{P}\bigl[\mu(X^\ell) = a\bigr], \quad a \in A$$

  как правило, существенно неравномерно!

  Будем называть это эффектом расслоения семейства $A$.

- **Неравенство Буля** сильно завышено, когда среди бинарных векторов ошибок есть много похожих.

  Будем называть это эффектом сходства алгоритмов.

## Оценка «бритва Оккама»

Мы не можем знать распределение $q(a) = \mathsf{P}\bigl[\mu(X^\ell) = a\bigr]$,
но можем попробовать его «угадать».

**Теорема (Дж. Лангфорд, 2002)**

Для произвольной нормированной функции, $p(a)$, $\sum\limits_{a\in A} p(a) = 1$,
любого $\eta \in (0,1)$, любого $a \in A$ с вероятностью не менее $1-\eta$

$$\nu(a, X^k) \leqslant \nu(a, X^\ell) + \sqrt{\frac{1}{\ell}\ln\frac{1}{p(a)} + \frac{1}{\ell}\ln\frac{2}{3\eta}}.$$

**Утв 1.** Если угадали, $p(a) = q(a)$, то $\mathsf{E}\ln\frac{1}{p(\mu(X^\ell))}$ минимально.

**Утв 2.** Бритва Оккама может учитывать эффект расслоения,
но не учитывает эффект сходства, поэтому тоже завышена.

## Оценка «бритва Оккама»: как задать $p(a)$?

**Пример 1.**

*Равномерное распределение* $p(a) = \frac{1}{|A|}$
даёт оценку Вапника–Червоненкиса:
для любых $a \in A$, $\eta \in (0,1)$ с вероятностью не менее $1-\eta$

$$\nu(a, X^k) \leqslant \nu(a, X^\ell) + \sqrt{\frac{1}{\ell}\ln|A| + \frac{1}{\ell}\ln\frac{2}{3\eta}}.$$

## Оценка «бритва Оккама»: как задать $p(a)$?

**Пример 2.**
Задача классификации на 2 класса $Y = \{-1, +1\}$,
$A$ — линейные классификаторы в $\mathbb{R}^n$:

$$a(x) = \operatorname{sign}(w_1 x^1 + \cdots + w_n x^n), \quad x = (x^1, \ldots, x^n) \in X.$$

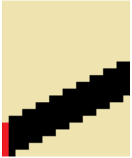
*Гауссовское распределение*: веса $w \in \mathbb{R}^n$ — независимые с.в., с нулевым ожиданием и равными дисперсиями $\sigma^2$:

$$p(a) = Z \exp\left(-\frac{1}{2\sigma^2}\|w\|^2\right).$$

Подставляя в «бритву Оккама», получаем... регуляризацию!

$$\nu(a, X^k) \leqslant \nu(a, X^\ell) + \sqrt{\frac{\|w\|^2}{2\ell\sigma^2} + \frac{1}{\ell}\ln\frac{2}{3\eta Z}}.$$

# Эксперимент: четыре семейства алгоритмов, заданных матрицами ошибок; лучший алгоритм у всех одинаков

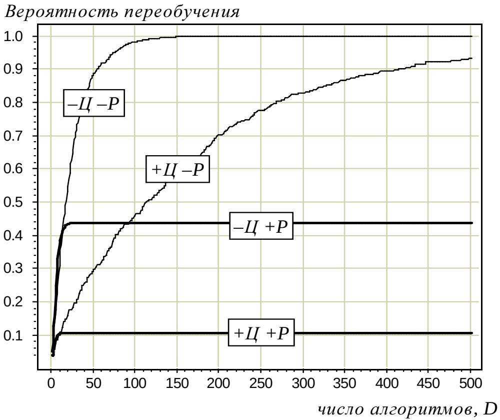
| | с расслоением по числу ошибок | без расслоения по числу ошибок |
|---|---|---|
| каждая пара соседних алгоритмов отличается только на одном объекте (образуется *цепь*) |  |  |
| соседние алгоритмы существенно различны, (*цепь* не образуется) |  |  |

Постепенно добавляя алгоритмы в $\{a_1, \ldots, a_D\}$, построим зависимости вероятности переобучения $Q_\varepsilon$ от числа $D$.

# Расслоение и связность снижают переобучение

**Эксперимент** с четырьмя семействами алгоритмов, $\ell = k = 100$, $\varepsilon = 0.05$:

+Ц — цепь;
−Ц — не цепь;
+Р — с расслоением;
−Р — без расслоения;




**Вывод:** получение точных оценок вероятности переобучения невозможно без учёта эффектов расслоения и связности.

## Граф расслоения–связности множества алгоритмов

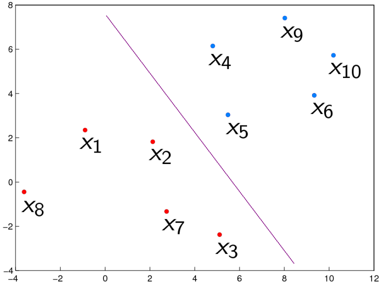
**Определим** бинарные отношения на множестве алгоритмов $A$:
*частичный порядок* $a \leqslant b$: $I(a,x) \leqslant I(b,x)$ для всех $x \in X^L$;
*предшествование* $a \prec b$: $a \leqslant b$ и $\|b - a\| = 1$.

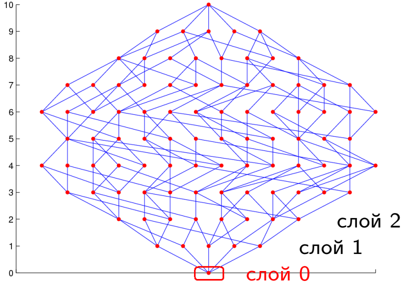
**Опр.** *Граф расслоения–связности* $\langle A, E\rangle$:
$A$ — множество попарно различных векторов ошибок;
$E = \bigl\{(a, b)\colon a \prec b\bigr\}$.

**Свойства графа расслоения–связности:**

- это подграф графа Хассе отношения порядка $\leqslant$ на $A$;
- каждому ребру $(a, b)$ соответствует объект $x_{ab} \in X^L$, такой, что $I(a, x_{ab}) = 0$, $\quad I(b, x_{ab}) = 1$;
- граф является многодольным со слоями
$A_m = \bigl\{a \in A\colon m(a, X^L) = m\bigr\}, \quad m = 0, \dots, L$;

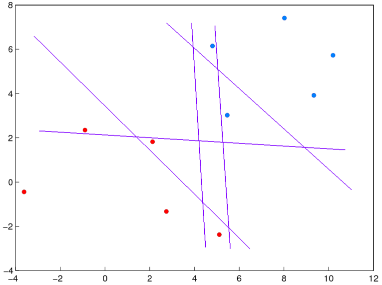
# Пример. Семейство линейных алгоритмов классификации

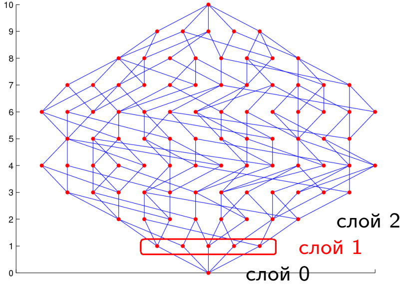



| | слой 0 |
|---|---|
| $x_1$ | 0 |
| $x_2$ | 0 |
| $x_3$ | 0 |
| $x_4$ | 0 |
| $x_5$ | 0 |
| $x_6$ | 0 |
| $x_7$ | 0 |
| $x_8$ | 0 |
| $x_9$ | 0 |
| $x_{10}$ | 0 |

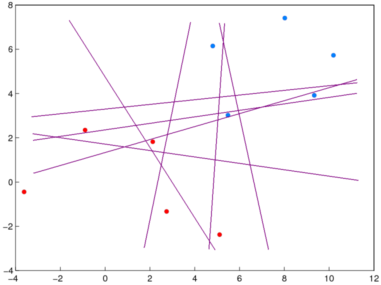
# Пример. Семейство линейных алгоритмов классификации

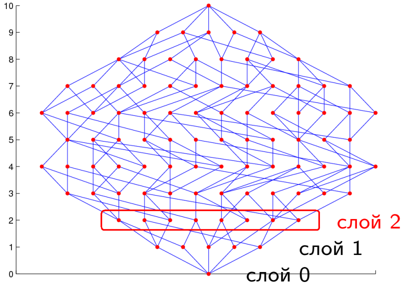



| | слой 0 | слой 1 | | | | |
|---|---|---|---|---|---|---|
| $x_1$ | 0 | 1 | 0 | 0 | 0 | 0 |
| $x_2$ | 0 | 0 | 1 | 0 | 0 | 0 |
| $x_3$ | 0 | 0 | 0 | 1 | 0 | 0 |
| $x_4$ | 0 | 0 | 0 | 0 | 1 | 0 |
| $x_5$ | 0 | 0 | 0 | 0 | 0 | 1 |
| $x_6$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $x_7$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $x_8$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $x_9$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $x_{10}$ | 0 | 0 | 0 | 0 | 0 | 0 |

# Пример. Семейство линейных алгоритмов классификации






| | слой 0 | слой 1 | | | | | слой 2 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | ... |
| $x_2$ | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| $x_3$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | ... |
| $x_4$ | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | ... |
| $x_5$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | ... |
| $x_6$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | ... |
| $x_7$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | ... |
| $x_8$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| $x_9$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| $x_{10}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |

## Характеристики расслоения и связности алгоритма $a \in A$

Множество объектов, соответствующих рёбрам, исходящим из $a$:

$$X_a = \bigl\{x_{ab} \in X^L \bigm| a \prec b\bigr\},$$

Множество объектов, соответствующих всем рёбрам на путях, ведущих в $a$:

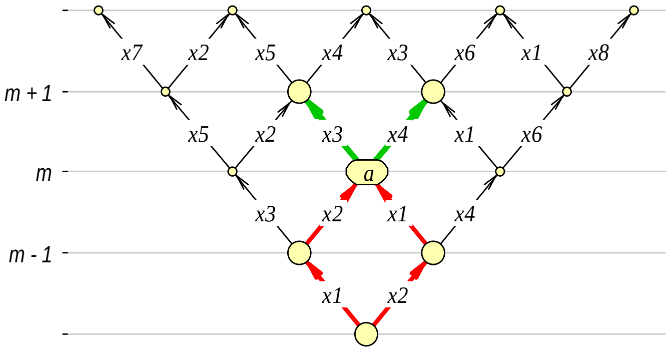
$$X'_a = \bigl\{x \in X^L \bigm| \exists b \in A\colon b < a,\ I(b,x) < I(a,x)\bigr\}.$$

**Опр.** Характеристики расслоения и связности алгоритма $a$:

$u(a) = |X_a|$ — верхняя связность алгоритма $a$;
$q(a) = |X'_a|$ — неполноценность алгоритма $a$;

# Пример: двумерная сеть алгоритмов

Верхняя связность: $u(a) = \#\{x3, x4\} = 2$;
Неполноценность: $q(a) = \#\{x1, x2\} = 2$;



## Монотонные методы обучения

**Опр.** Метод обучения $\mu$ называется *монотонным*, если

$$\mu(X^\ell) = \arg\min_{a\in A} K(a, X^\ell),$$

где $K(a, X)$ — строго монотонная функция вектора ошибок $a$:

$$\forall X \subset X^L,\ \forall a, b \in A \quad a < b \ \to\ K(a, X) < K(b, X).$$

**Пример 1.** *Взвешенный эмпирический риск* с весами $w_i$:

$$K(a, X) = \sum_{x_i\in X} w_i\, I(a, x_i), \qquad w_i > 0.$$

**Опр.** Монотонный метод $\mu$ называется *пессимистичным*, если

$$\mu(X^\ell) = \arg\max_{a\in A(X^\ell)} \delta(a, X^\ell), \quad A(X^\ell) = \operatorname{Arg}\min_{a\in A} K(a, X^\ell).$$

## Верхняя оценка расслоения–связности

### Теорема (Воронцов, 2010)

Для любого монотонного метода $\mu$, любых $X^L$, $A$ и $\varepsilon \in (0, 1)$

$$Q_\varepsilon \leqslant \sum_{a \in A} \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell} \mathcal{H}_{L-u-q}^{\ell-u,\, m-q} \left( \tfrac{\ell}{L}(m - \varepsilon k) \right)$$

где $u = |X_a|$ — верхняя связность алгоритма $a$,
$q = |X'_a|$ — неполноценность алгоритма $a$,
$m = m(a, X^L)$ — число ошибок алгоритма $a$,

$$\mathcal{H}_L^{\ell,\, m}(z) = \sum_{s=0}^{\lfloor z \rfloor} \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell}, \quad z = 0, \ldots, \ell$$

— функция гипергеометрического распределения:

## Идея доказательства

1. Пусть $\mu$ — произвольный монотонный метод обучения, $\bar\mu$ — монотонный пессимистичный метод обучения. Тогда
$$Q_\varepsilon(\mu, X^L) \leqslant Q_\varepsilon(\bar\mu, X^L).$$

2. Если $\bar\mu(X^\ell) = a$, то $\begin{cases} X_a \subseteq X^\ell & \text{в силу пессимистичности } \bar\mu, \\ X'_a \subseteq X^k & \text{в силу монотонности } \bar\mu. \end{cases}$

3. $$\mathsf{P}\Bigl[\bar\mu(X^\ell) = a\Bigr] \leqslant \mathsf{P}\bigl[\underbrace{X_a \subseteq X^\ell \text{ и } X'_a \subseteq X^k}_{S(a, X^\ell)}\bigr] = \frac{C_{L-|X_a|-|X'_a|}^{\ell-|X_a|}}{C_L^\ell} = \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell}.$$

4. По формуле полной вероятности:
$$Q_\varepsilon(\bar\mu, X^L) = \sum_{a \in A} \underbrace{\mathsf{P}\Bigl[S(a, X^\ell)\Bigr]}_{C_{L-u-q}^{\ell-u}/C_L^\ell} \cdot \underbrace{\mathsf{P}\Bigl[\delta(a, X^\ell) \geqslant \varepsilon \Bigm| S(a, X^\ell)\Bigr]}_{\mathcal{H}_{L-u-q}^{\ell-u,\, m-q}\left(\frac{\ell}{L}(m - \varepsilon k)\right)}. \quad \blacksquare$$

## Свойства оценки $Q_\varepsilon \leqslant \sum\limits_{a\in A} \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell} \mathcal{H}_{L-u-q}^{\ell-u,\, m-q} \bigl(\frac{\ell}{L}(m-\varepsilon k)\bigr)$

1. Вклад алгоритма $a \in A$ убывает экспоненциально
по $u(a) \Rightarrow$ **связные семейства меньше переобучаются**;
по $q(a) \Rightarrow$ **только нижние слои вносят вклад в $Q_\varepsilon$.**
2. Оценка обращается в равенство в случае многомерных монотонных сетей алгоритмов.
3. Вероятность получить алгоритм в результате обучения
$$\mathsf{P}\bigl[\mu(X^\ell) = a\bigr] \leqslant P_a = \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell}.$$
4. Если $q(a) > k$, то $P_a = 0$ и вклад алгоритма $a$ равен 0
$\Rightarrow$ при малых $k$ оценка вырождается.
5. $\sum\limits_{a\in A} P_a$ — оценка степени завышенности.

## Свойства оценки $Q_\varepsilon \leqslant \sum\limits_{a\in A} \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell} \mathcal{H}_{L-u-q}^{\ell-u,\, m-q}\left(\frac{\ell}{L}(m-\varepsilon k)\right)$

6. При $|A| = 1$ это вероятность большого отклонения частот в двух выборках:

$$Q_\varepsilon = \mathcal{H}_L^{\ell,\,m}\left(\tfrac{\ell}{L}(m-\varepsilon k)\right) \to 0 \;\text{ при }\; \ell, k \to \infty.$$

7. При $q = u = 0$ и $\ell = k$ это оценка Вапника-Червоненкиса:

$$Q_\varepsilon \leqslant \sum_{a\in A} \mathcal{H}_L^{\ell,\,m}\left(\tfrac{\ell}{L}(m-\varepsilon k)\right) \leqslant |A| \cdot \tfrac{3}{2}\exp(-\varepsilon\ell^2).$$

8. При замене неполноценности $q$ на нижнюю связность $d$ это верхняя оценка *функционала равномерного отклонения*

$$\widetilde{Q}_\varepsilon(A, X^L) = \mathsf{P}\left[\sup_{a\in A}\left(\nu(a, X^k) - \nu(a, X^\ell)\right) \geqslant \varepsilon\right],$$

который учитывает связность, но не учитывает расслоение.

## Резюме

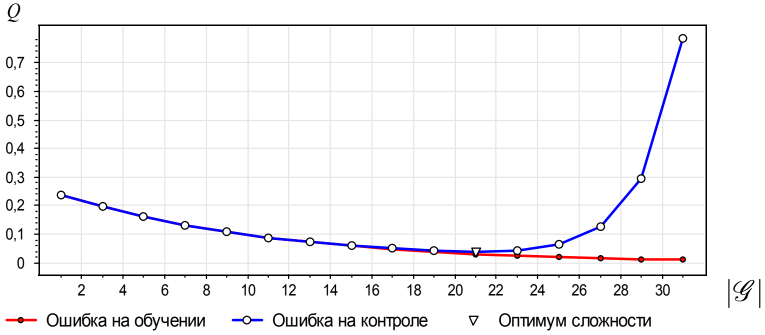
- Свойства расслоения и связности уменьшают переобучение.
- На практике семейства, как правило, ими обладают. Иначе вероятность переобучения была бы близка к 1 уже при $|A|$ порядка нескольких десятков.
- Практическое применение комбинаторных оценок:
  1) оценить $\eta = Q_\varepsilon(\mu, X^L)$ по нескольким нижним слоям;
  2) применив обращение, оценить $\varepsilon$ через $\eta$;
  3) использовать оценку $\nu(X^\ell) + \varepsilon$ как внешний критерий для выбора модели, метода или отбора признаков.
- Информацию о нижних слоях можно получать как «побочный продукт» в итерационных методах обучения.
- В теории переобучения ещё много открытых проблем.

# Для отбора признаков используются внешние критерии

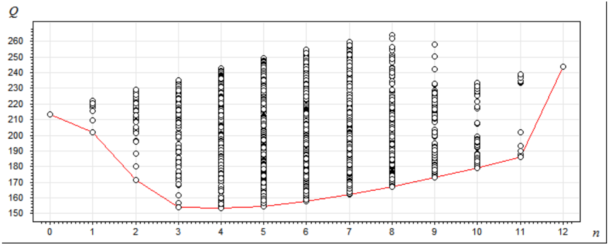
$\mathscr{F} = \{f_j\colon X \to D_j : j = 1, \dots, n\}$ — множество признаков;

$\mu_{\mathscr{G}}$ — метод обучения, использующий только признаки $\mathscr{G} \subseteq \mathscr{F}$;

$Q(\mathscr{G}) = Q(\mu_{\mathscr{G}}, X^\ell)$ — какой-либо внешний критерий.



# Алгоритм полного перебора (Full Search)




**Вход:** множество $\mathscr{F}$, критерий $Q$, параметр $d$;

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

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

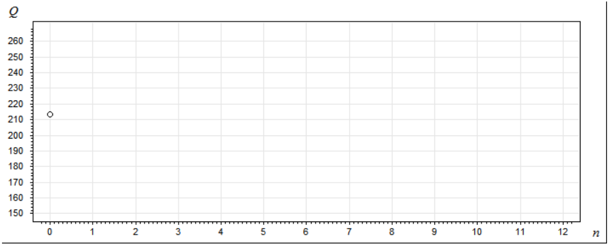
3: найти лучший набор сложности $j$:

$$\mathscr{G}_j := \mathop{\arg\min}_{\mathscr{G}\colon |\mathscr{G}|=j} Q(\mathscr{G});$$

4: **если** $Q(\mathscr{G}_j) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathscr{G}_j)$;

5: **если** $j - j^* \geqslant d$ **то вернуть** $\mathscr{G}_{j^*}$;

# Алгоритм полного перебора (Full Search)

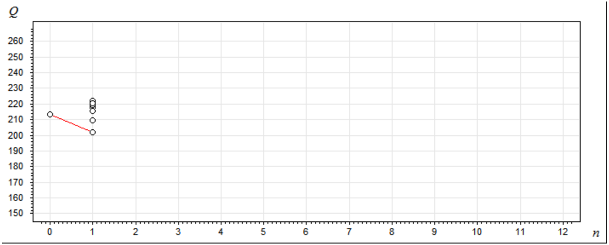



$j = 0$

**Вход:** множество $\mathscr{F}$, критерий $Q$, параметр $d$;

1: $Q^* := Q(\varnothing)$; — инициализация;
2: **для всех** $j = 1, \dots, n$, где $j$ — сложность наборов:
3: найти лучший набор сложности $j$:
$$\mathscr{G}_j := \mathop{\arg\min}_{\mathscr{G}:\, |\mathscr{G}|=j} Q(\mathscr{G});$$
4: **если** $Q(\mathscr{G}_j) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathscr{G}_j)$;
5: **если** $j - j^* \geqslant d$ **то вернуть** $\mathscr{G}_{j^*}$;

## Алгоритм полного перебора (Full Search)

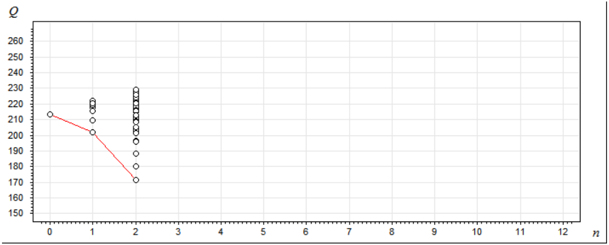



$j = 1$
$j^* = 1$

**Вход:** множество $\mathscr{F}$, критерий $Q$, параметр $d$;

1: $Q^* := Q(\varnothing)$; — инициализация;
2: **для всех** $j = 1, \ldots, n$, где $j$ — сложность наборов:
3: найти лучший набор сложности $j$:
$\mathscr{G}_j := \arg\min\limits_{\mathscr{G}:\, |\mathscr{G}|=j} Q(\mathscr{G})$;
4: **если** $Q(\mathscr{G}_j) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathscr{G}_j)$;
5: **если** $j - j^* \geqslant d$ **то вернуть** $\mathscr{G}_{j^*}$;

# Алгоритм полного перебора (Full Search)



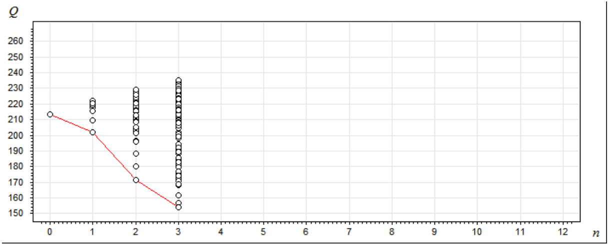

$j = 2$
$j^* = 2$

**Вход:** множество $\mathscr{F}$, критерий $Q$, параметр $d$;

---

$Q^* := Q(\varnothing)$; — инициализация;
**для всех** $j = 1, \dots, n$, где $j$ — сложность наборов:
  найти лучший набор сложности $j$:
  $\mathscr{G}_j := \arg\min\limits_{\mathscr{G}:\, |\mathscr{G}|=j} Q(\mathscr{G})$;
  **если** $Q(\mathscr{G}_j) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathscr{G}_j)$;
  **если** $j - j^* \geqslant d$ **то вернуть** $\mathscr{G}_{j^*}$;

# Алгоритм полного перебора (Full Search)

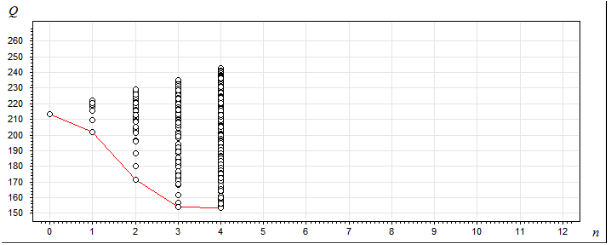



$j = 3$
$j^* = 3$

**Вход:** множество $\mathscr{F}$, критерий $Q$, параметр $d$;

1: $Q^* := Q(\varnothing)$; — инициализация;
2: **для всех** $j = 1, \dots, n$, где $j$ — сложность наборов:
3: найти лучший набор сложности $j$:
$$\mathscr{G}_j := \arg\min_{\mathscr{G}:\, |\mathscr{G}|=j} Q(\mathscr{G});$$
4: **если** $Q(\mathscr{G}_j) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathscr{G}_j)$;
5: **если** $j - j^* \geqslant d$ **то вернуть** $\mathscr{G}_{j^*}$;

# Алгоритм полного перебора (Full Search)

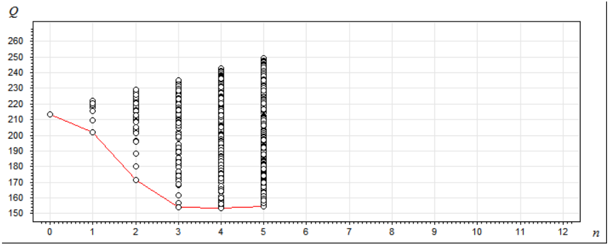



$j = 4$
$j^* = 4$

**Вход:** множество $\mathscr{F}$, критерий $Q$, параметр $d$;

1: $Q^* := Q(\varnothing)$; — инициализация;
2: **для всех** $j = 1, \dots, n$, где $j$ — сложность наборов:
3: найти лучший набор сложности $j$:
$$\mathscr{G}_j := \arg\min_{\mathscr{G}:\, |\mathscr{G}|=j} Q(\mathscr{G});$$
4: **если** $Q(\mathscr{G}_j) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathscr{G}_j)$;
5: **если** $j - j^* \geqslant d$ **то вернуть** $\mathscr{G}_{j^*}$;

# Алгоритм полного перебора (Full Search)

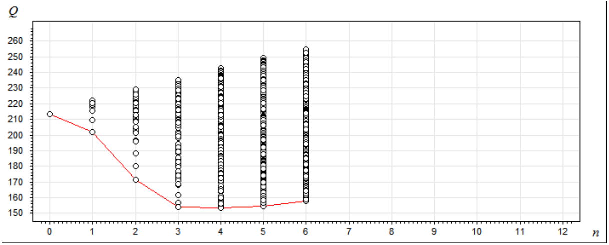



$j = 5$
$j^* = 4$

**Вход:** множество $\mathscr{F}$, критерий $Q$, параметр $d$;

1: $Q^* := Q(\varnothing)$; — инициализация;
2: **для всех** $j = 1, \ldots, n$, где $j$ — сложность наборов:
3: найти лучший набор сложности $j$:
$\mathscr{G}_j := \arg\min\limits_{\mathscr{G}\colon |\mathscr{G}|=j} Q(\mathscr{G})$;
4: **если** $Q(\mathscr{G}_j) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathscr{G}_j)$;
5: **если** $j - j^* \geqslant d$ **то вернуть** $\mathscr{G}_{j^*}$;

## Алгоритм полного перебора (Full Search)




$j = 6$
$j^* = 4$

**Вход:** множество $\mathscr{F}$, критерий $Q$, параметр $d$;

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

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

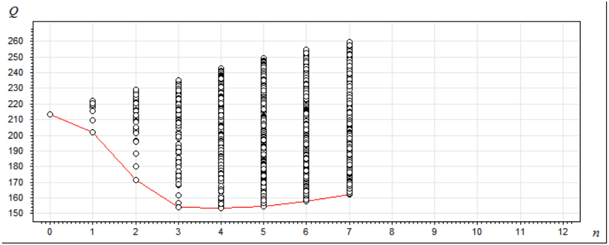
3: найти лучший набор сложности $j$:

$$\mathscr{G}_j := \mathop{\arg\min}_{\mathscr{G}\colon |\mathscr{G}|=j} Q(\mathscr{G});$$

4: **если** $Q(\mathscr{G}_j) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathscr{G}_j)$;

5: **если** $j - j^* \geqslant d$ **то вернуть** $\mathscr{G}_{j^*}$;

# Алгоритм полного перебора (Full Search)




$j = 7$
$j^* = 4$

**Вход:** множество $\mathscr{F}$, критерий $Q$, параметр $d$;

---

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

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

найти лучший набор сложности $j$:

$$\mathscr{G}_j := \arg\min_{\mathscr{G}\colon |\mathscr{G}|=j} Q(\mathscr{G});$$

**если** $Q(\mathscr{G}_j) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathscr{G}_j)$;

**если** $j - j^* \geqslant d$ **то вернуть** $\mathscr{G}_{j^*}$;

# Алгоритм полного перебора (Full Search)

**Преимущества:**

- простота реализации;
- гарантированный результат;
- неплохой выбор, когда информативных признаков $\lesssim 5$;
- неплохой выбор, когда всего признаков $\lesssim 20$.

**Недостатки:**

- в остальных случаях ооооооочень долго — $O(2^n)$.

**Способы устранения:**

- эвристические методы сокращённого перебора.

## Алгоритм жадного добавления

**Вход:** множество $\mathscr{F}$, критерий $Q$, параметр $d$;

---

$\mathscr{G}_0 := \varnothing$; $Q^* := Q(\varnothing)$; — инициализация;

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

найти признак, наиболее выгодный для добавления:

$f^* := \arg\min\limits_{f \in \mathscr{F} \setminus \mathscr{G}_{j-1}} Q\bigl(\mathscr{G}_{j-1} \cup \{f\}\bigr)$;

добавить этот признак в набор:

$\mathscr{G}_j := \mathscr{G}_{j-1} \cup \{f^*\}$;

**если** $Q(\mathscr{G}_j) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathscr{G}_j)$;

**если** $j - j^* \geqslant d$ **то вернуть** $\mathscr{G}_{j^*}$;

# Алгоритм жадного добавления

**Преимущества:**

- работает быстро — $O(n^2)$, точнее $O(n(j^* + d))$;
- возможны быстрые инкрементные алгоритмы;

**Недостатки:**

- Add склонен включать в набор лишние признаки.

**Способы устранения:**

- Add-Del — чередование добавлений и удалений;
- расширение поиска.

# Алгоритм поочерёдного добавления и удаления

$\mathscr{G}_0 := \varnothing$; $Q^* := Q(\varnothing)$; $t := 0$; — инициализация;
**повторять**

---

  **пока** $|\mathscr{G}_t| < n$ добавлять признаки (ADD):
    $t := t + 1$; — началась следующая итерация;
    $f^* := \arg\min\limits_{f \in \mathscr{F} \setminus \mathscr{G}_{t-1}} Q\bigl(\mathscr{G}_{t-1} \cup \{f\}\bigr)$; $\mathscr{G}_t := \mathscr{G}_{t-1} \cup \{f^*\}$;
    **если** $Q(\mathscr{G}_t) < Q^*$ **то** $t^* := t$; $Q^* := Q(\mathscr{G}_t)$;
    **если** $t - t^* \geqslant d$ **то прервать цикл**;

---

  **пока** $|\mathscr{G}_t| > 0$ удалять признаки (DEL):
    $t := t + 1$; — началась следующая итерация;
    $f^* := \arg\min\limits_{f \in \mathscr{G}_{t-1}} Q\bigl(\mathscr{G}_{t-1} \setminus \{f\}\bigr)$; $\mathscr{G}_t := \mathscr{G}_{t-1} \setminus \{f^*\}$;
    **если** $Q(\mathscr{G}_t) < Q^*$ **то** $t^* := t$; $Q^* := Q(\mathscr{G}_t)$;
    **если** $t - t^* \geqslant d$ **то прервать цикл**;

---

**пока** значения критерия $Q(\mathscr{G}_{t^*})$ уменьшаются;
**вернуть** $\mathscr{G}_{t^*}$;

## Алгоритм поочерёдного добавления и удаления

**Преимущества:**

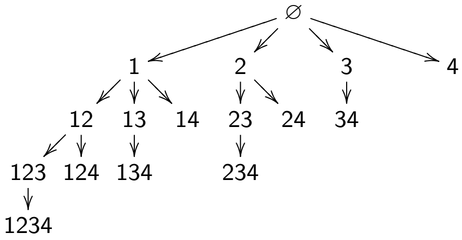
- как правило, лучше, чем Add и Del по отдельности;
- возможны быстрые инкрементные алгоритмы (пример — *шаговая регрессия*).

**Недостатки:**

- работает дольше, оптимальность не гарантирует.

# Поиск в глубину (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 $\mathscr{G}$ бесперспективен, то больше не пытаться его наращивать.

# Поиск в глубину (метод ветвей и границ)

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

**Оценка бесперспективности** набора признаков $\mathscr{G}$:
набор $\mathscr{G}$ не наращивается, если

$$\exists j: \quad Q(\mathscr{G}) \geqslant \varkappa Q_j^* \quad \text{и} \quad |\mathscr{G}| \geqslant j + d,$$

$d \geqslant 0$ — целочисленный параметр,
$\varkappa \geqslant 1$ — вещественный параметр.

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

# Поиск в глубину (метод ветвей и границ)

**Вход:** множество $\mathscr{F}$, критерий $Q$, параметры $d$ и $\varkappa$;

---

1: **ПРОЦЕДУРА** Нарастить $(\mathscr{G})$;
2: **если** найдётся $j \leqslant |\mathscr{G}| - d$ такое, что $Q(\mathscr{G}) \geqslant \varkappa Q_j^*$, **то**
3: **выход**;
4: $Q_{|\mathscr{G}|}^* := \min\bigl\{Q_{|\mathscr{G}|}^*, Q(\mathscr{G})\bigr\}$;
5: **для всех** $f_s \in \mathscr{F}$ таких, что $s > \max\{t \mid f_t \in \mathscr{G}\}$
Нарастить $(\mathscr{G} \cup \{f_s\})$;

---

6: Инициализация массива лучших значений критерия:
$Q_j^* := Q(\varnothing)$ для всех $j = 1, \dots, n$;
7: Упорядочить признаки по убыванию информативности;
8: Нарастить $(\varnothing)$;
9: **вернуть** $\mathscr{G}$, для которого $Q(\mathscr{G}) = \min\limits_{j=1,\dots,n} Q_j^*$;

# Поиск в ширину

Он же *многорядный итерационный алгоритм МГУА* (МГУА — метод группового учёта аргументов).

Философский принцип *неокончательных решений* Габора: принимая решения, следует оставлять максимальную свободу 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*:

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

где $B_j \leqslant B$ — параметр *ширины поиска*.

# Поиск в ширину

**Вход:** множество $\mathscr{F}$, критерий $Q$, параметры $d$, $B$;

---

1: первый ряд состоит из всех наборов длины 1:
$R_1 := \bigl\{\{f_1\}, \ldots, \{f_n\}\bigr\}$; $Q^* = Q(\varnothing)$;
2: **для всех** $j = 1, \ldots, n$, где $j$ — сложность наборов:
3: отсортировать ряд $R_j = \bigl\{\mathscr{G}_j^1, \ldots, \mathscr{G}_j^{B_j}\bigr\}$
по возрастанию критерия: $Q\bigl(\mathscr{G}_j^1\bigr) \leqslant \ldots \leqslant Q\bigl(\mathscr{G}_j^{B_j}\bigr)$;
4: **если** $B_j > B$ **то**
5: $R_j := \bigl\{\mathscr{G}_j^1, \ldots, \mathscr{G}_j^B\bigr\}$; — $B$ лучших наборов ряда;
6: **если** $Q(\mathscr{G}_j^1) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathscr{G}_j^1)$;
7: **если** $j - j^* \geqslant d$ **то вернуть** $\mathscr{G}_{j^*}^1$;
8: породить следующий ряд:
$R_{j+1} := \bigl\{\mathscr{G} \cup \{f\} \bigm| \mathscr{G} \in R_j,\ f \in \mathscr{F} \backslash \mathscr{G}\bigr\}$;

## Поиск в ширину

- **Трудоёмкость:**
  $O(n^2)$, точнее $O(Bn(j^* + d))$.
- **Проблема дубликатов:**
  после сортировки (шаг 3) проверить на совпадение только соседние наборы с равными значениями внутреннего и внешнего критерия.
- **Адаптивный отбор признаков:**
  на шаге 8 добавлять к $j$-му ряду только признаки $f$ с наибольшей информативностью $I_j(f)$:

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

## Генетический алгоритм поиска (идея и терминология)

$\mathscr{G} \subseteq \mathscr{F}$ — *индивид* (в МГУА «модель»);
$R_t := \{\mathscr{G}_t^1, \ldots, \mathscr{G}_t^{B_t}\}$ — *поколение* (в МГУА — «ряд»);
$\beta = (\beta_j)_{j=1}^n$, $\beta_j = [f_j \in \mathscr{G}]$ — *хромосома*, кодирующая $\mathscr{G}$;

Бинарная операция *скрещивания* $\beta = \beta' \times \beta''$:

$$\beta_j = \begin{cases} \beta'_j, & \text{с вероятностью } 1/2; \\ \beta''_j, & \text{с вероятностью } 1/2; \end{cases}$$

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

$$\beta_j = \begin{cases} 1 - \beta'_j, & \text{с вероятностью } p_m; \\ \beta'_j, & \text{с вероятностью } 1 - p_m; \end{cases}$$

где параметр $p_m$ — *вероятность мутации*.

## Генетический (эволюционный) алгоритм

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

---

1: инициализировать случайную популяцию из $B$ наборов:
$B_1 := B$; $R_1 := \{\mathscr{G}_1^1, \dots, \mathscr{G}_1^{B_1}\}$; $Q^* := Q(\varnothing)$;
2: **для всех** $t = 1, \dots, T$, где $t$ — номер поколения:
3: ранжирование индивидов: $Q(\mathscr{G}_t^1) \leqslant \dots \leqslant Q(\mathscr{G}_t^{B_t})$;
4: **если** $B_t > B$ **то**
5: селекция: $R_t := \{\mathscr{G}_t^1, \dots, \mathscr{G}_t^B\}$;
6: **если** $Q(\mathscr{G}_t^1) < Q^*$ **то** $t^* := t$; $Q^* := Q(\mathscr{G}_t^1)$;
7: **если** $t - t^* \geqslant d$ **то вернуть** $\mathscr{G}_{t^*}^1$;
8: породить $t{+}1$-е поколение путём скрещиваний и мутаций:
$R_{t+1} := \{\sim(\mathscr{G}' \times \mathscr{G}'') \mid \mathscr{G}', \mathscr{G}'' \in R_t\} \cup R_t$;

## 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).

# Генетический (эволюционный) алгоритм

**Преимущества:**

- it is fun!
- возможность введения различных эвристик;
- решает задачи даже с очень большим числом признаков.

**Недостатки:**

- относительно медленная сходимость;
- отсутствие теории;
- подбор параметров — непростое искусство;

## Случайный поиск — упрощенный генетический алгоритм

**Модификация:** шаг 8

- породить $t+1$-е поколение путём многократных *мутаций*:
  $R_{t+1} := \left\{\sim\mathscr{G},\ldots,\sim\mathscr{G} \;\middle|\; \mathscr{G}\in R_t\right\} \cup R_t;$

**Недостатки:**

- ничем не лучше ГА;
- очень медленная сходимость.

**Способ устранения:**

- СПА — случайный поиск с адаптацией.

**Основная идея адаптации:**

- увеличивать вероятность появления тех признаков, которые часто входят в наилучшие наборы,
- одновременно уменьшать вероятность появления признаков, которые часто входят в наихудшие наборы.

# Случайный поиск с адаптацией (СПА)

**Вход:** множество $\mathscr{F}$, критерий $Q$, параметры $d$, $j_0$, $T$, $r$, $h$;

1: $p_1 = \dots = p_n := 1/n;$ — равные вероятности признаков;
2: **для всех** $j = j_0, \dots, n$, где $j$ — сложность наборов:
3: **для всех** $t = 1, \dots, T$, где $t$ — номер итерации:
4: $r$ случайных наборов признаков из распределения $\{p_1, \dots, p_n\}$:
$R_{jt} := \{\mathscr{G}^1_{jt}, \dots, \mathscr{G}^r_{jt}\}$, $|\mathscr{G}^1_{jt}| = \dots = |\mathscr{G}^r_{jt}| = j$;
5: $\mathscr{G}^{\min}_{jt} := \arg\min\limits_{\mathscr{G} \in R_{jt}} Q(\mathscr{G})$; — лучший из $r$ наборов;
6: $\mathscr{G}^{\max}_{jt} := \arg\max\limits_{\mathscr{G} \in R_{jt}} Q(\mathscr{G})$; — худший из $r$ наборов;
7: $H := 0$; наказание для всех $f_s \in \mathscr{G}^{\max}_{jt}$:
$\Delta p_s := \min\{p_s, h\}$; $p_s := p_s - \Delta p_s$; $H := H + \Delta p_s$;
8: поощрение для всех $f_s \in \mathscr{G}^{\min}_{jt}$: $p_s := p_s + H/j$;
9: $\mathscr{G}_j := \arg\min\limits_{\mathscr{G} \in R_{j1}, \dots, R_{jT}} Q(\mathscr{G})$; — лучший набор сложности $j$;
10: **если** $Q(\mathscr{G}_j) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathscr{G}_j)$;
11: **если** $j - j^* \geqslant d$ **то вернуть** $\mathscr{G}_{j^*}$;

# Случайный поиск с адаптацией (СПА)

**Рекомендации по выбору параметров $r, T, h$:**

$T \approx 10..50$ — число итераций;

$r \approx 20..100$ — число наборов, создаваемых на каждой итерации;

$h \approx \frac{1}{r\,n}$ — скорость адаптации;

**Преимущества:**

- трудоёмкость порядка $O(Tr(j^* + d))$ операций;
- меньшее число параметров, по сравнению с генетикой;
- довольно быстрая сходимость.

**Недостатки:**

- при большом числе признаков СПА малоэффективен.

## Резюме в конце лекции

- Отбор признаков надо вести по внешним критериям.
- Критерии, основанные на оценках обобщающей способности, наиболее вычислительно эффективны.
- Для отбора признаков могут использоваться любые эвристические методы дискретной оптимизации

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

- Большинство эвристик основаны на двух идеях:
  — среди признаков есть наиболее информативные;
  — $Q(\mathscr{G})$ изменяется не сильно при небольшом изменении $\mathscr{G}$.
- МГУА, ГА, СПА очень похожи — на их основе легко создавать различные «симбиотические» алгоритмы.