Обобщающая способность
Методы отбора признаков

К. В. Воронцов, А.В. Зухба
vokov@forecsys.ru
a__l@mail.ru

октябрь 2015

Содержание

Задача выбора модели (model selection)

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, $t = 1, \ldots, T$;
$\mu_t \colon (X \times Y)^\ell \to A_t$ — методы обучения, $t = 1, \ldots, T$.

Найти: метод μ_t, наиболее адекватный (т. е. обладающий наилучшей обобщающей способностью) для данной задачи.

Частные случаи:

- выбор наиболее адекватной модели A_t;
- выбор метода обучения μ_t для заданной модели A (в частности, подбор значения гиперпараметра);
- выбор признаков.

Задача отбора признаков (features selection)

$\mathscr{F} = \{f_j : X \to D_j : j = 1, \ldots, n\}$ — множество признаков;
$\mu_{\mathscr{G}}$ — метод обучения, использующий «урезанные» описания объектов, включающие только признаки из $\mathscr{G} \subseteq \mathscr{F}$.

Задача отбора признаков:

- Какие признаки лишние (шумовые)?
- Какие признаки дублируют друг друга?

Основная проблема:
Непустых наборов признаков: $2^n - 1$.
Задача выбора подмножества в общем случае NP-трудна.

Функционал качества алгоритма на выборке:

$$Q(a, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathscr{L}(a, x_i).$$

Внешние и внутренние критерии

Внутренний критерий — это качество на обучении X^ℓ:

$$Q_{\text{int}}(\mu, X^\ell) = Q\big(\mu(X^\ell), X^\ell\big).$$

Внешний критерий — это качество на новых данных.
Строгого определения нет. Известно много разновидностей.

Например, *ошибка на отложенных данных* (hold-out error):

$$Q_{\text{ext}}(\mu, X^L) = Q\big(\mu(X^\ell), X^k\big),$$

где X^k — контрольная выборка, $X^L = X^\ell \sqcup X^k$, $L = \ell + k$.

Недостатки hold-out:

1) результат зависит от способа разбиения выборки;
2) уменьшается длина обучения $\ell \ll L$;
3) при малых k слишком велика дисперсия оценки.

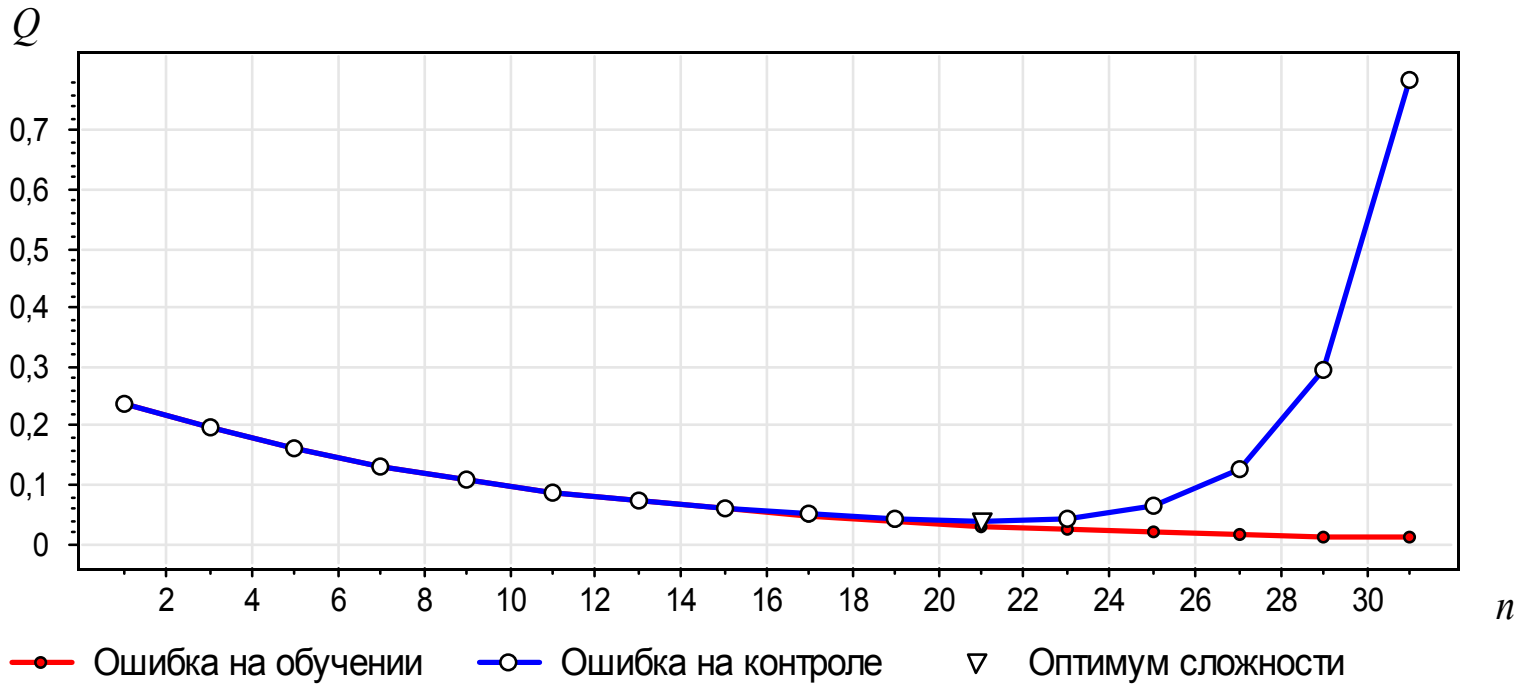
Внешние и внутренние критерии

Внутренний критерий — это качество на обучении X^ℓ:

$$Q_{\text{int}}(\mu, X^\ell) = Q\big(\mu(X^\ell), X^\ell\big).$$

Внешний критерий — это качество на новых данных.

Внешний критерий позволяет найти оптимум сложности модели.



Скользящий контроль

$X^L = X_n^\ell \sqcup X_n^k$, $n = 1, \ldots, N$ — несколько различных разбиений;
Оценка *скользящего контроля* (cross-validation error, CV):

$$\mathrm{CV}(\mu, X^L) = \frac{1}{N} \sum_{n=1}^{N} Q\big(\mu(X_n^\ell), X_n^k\big).$$

Полный скользящий контроль (complete CV): $N = C_L^\ell$.

Преимущества CCV:

- это наиболее устойчивая оценка из всех разновидностей CV;
- иногда удаётся выразить точную оценку (kNN).

Недостатки CCV:

- ресурсоёмкость — при $k > 2$ вычислить CCV нереально.

Скользящий контроль «leave-one-out»

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

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

Преимущества LOO:

- каждый объект ровно один раз участвует в контроле;
- длина обучения $\ell = L - 1$.

Недостатки LOO:

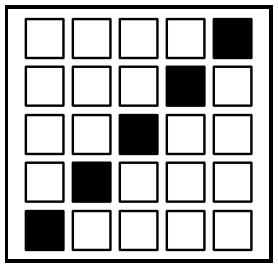
- каждый объект *лишь один раз* участвует в контроле;
- ресурсоёмкость;
- высокая дисперсия.

Скользящий контроль с разбиением по блокам

Контроль по q блокам (q-fold CV):

$X^L = X_1^{\ell_1} \sqcup \cdots \sqcup X_q^{\ell_q}, \quad \ell_1 + \cdots + \ell_q = L;$

$$Q_{\text{ext}}(\mu, X^L) = \frac{1}{q} \sum_{n=1}^{q} Q\left(\mu(X^L \setminus X_n^{\ell_n}), X_n^{\ell_n}\right).$$



Преимущества q-fold CV:

- компромисс между LOO и hold-out.

Недостатки q-fold CV:

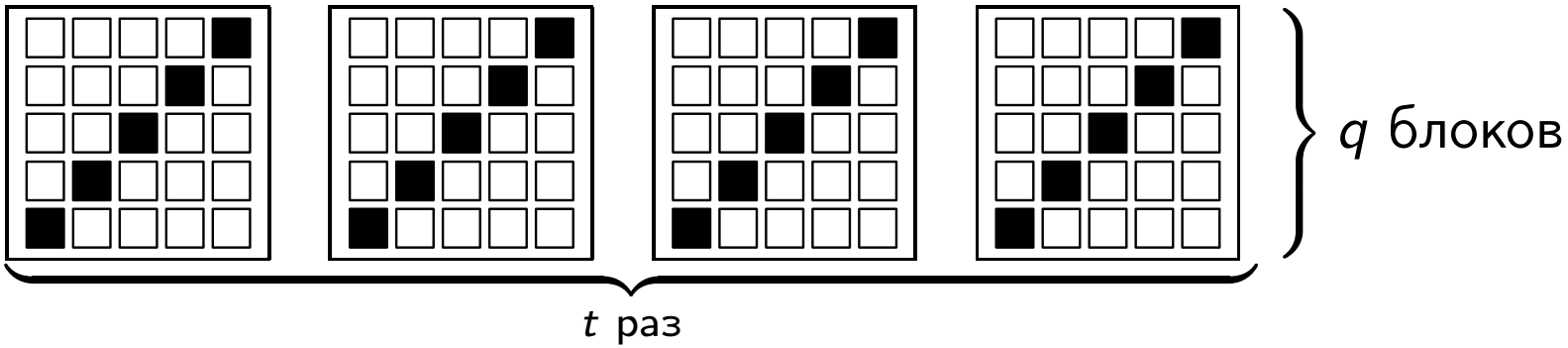
- каждый объект лишь один раз участвует в контроле.

Скользящий контроль с многократным разбиением по блокам

Контроль t раз по q блокам ($t \times q$-fold CV):

$$X^L = X_{s1}^{\ell_1} \sqcup \cdots \sqcup X_{sq}^{\ell_q}, \quad s = 1, \ldots, t, \quad \ell_1 + \cdots + \ell_q = L;$$

$$Q_{\text{ext}}(\mu, X^L) = \frac{1}{t} \sum_{s=1}^{t} \frac{1}{q} \sum_{n=1}^{q} Q\left(\mu(X^L \setminus X_{sn}^{\ell_n}), X_{sn}^{\ell_n}\right).$$



Преимущества $t \times q$-fold CV:

- компромисс между точностью и временем вычислений;
- каждый объект участвует в контроле ровно t раз;
- легко вычислять доверительные интервалы (при $t \geqslant 20$);

Критерии непротиворечивости моделей

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

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

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

Аналог CV: по N разбиениям $X^L = X_n^\ell \sqcup X_n^k$, $n = 1, \ldots, N$:

$$Q_{\text{ext}}(\mu, X^L) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{L} \sum_{i=1}^{L} \left|\mu(X_n^\ell)(x_i) - \mu(X_n^k)(x_i)\right|.$$

Недостатки:

- выборка сокращается в 2 раза;
- трудоёмкость возрастает в 2 раза;
- высокая дисперсия (сокращается при увеличении N).

Аналитические оценки и их обращение

Основная идея аналитического подхода:

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

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

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

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

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

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

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: $A = \{a(x) = \text{sign}\langle w, x\rangle\}$ — классификация,

$A = \{a(x) = \langle w, x\rangle\}$ — 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):

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

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 $\mathrm{D}\big(y_i - a(x_i)\big)$.

Байесовский информационный критерий (Bayes Inform. Criterion):

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

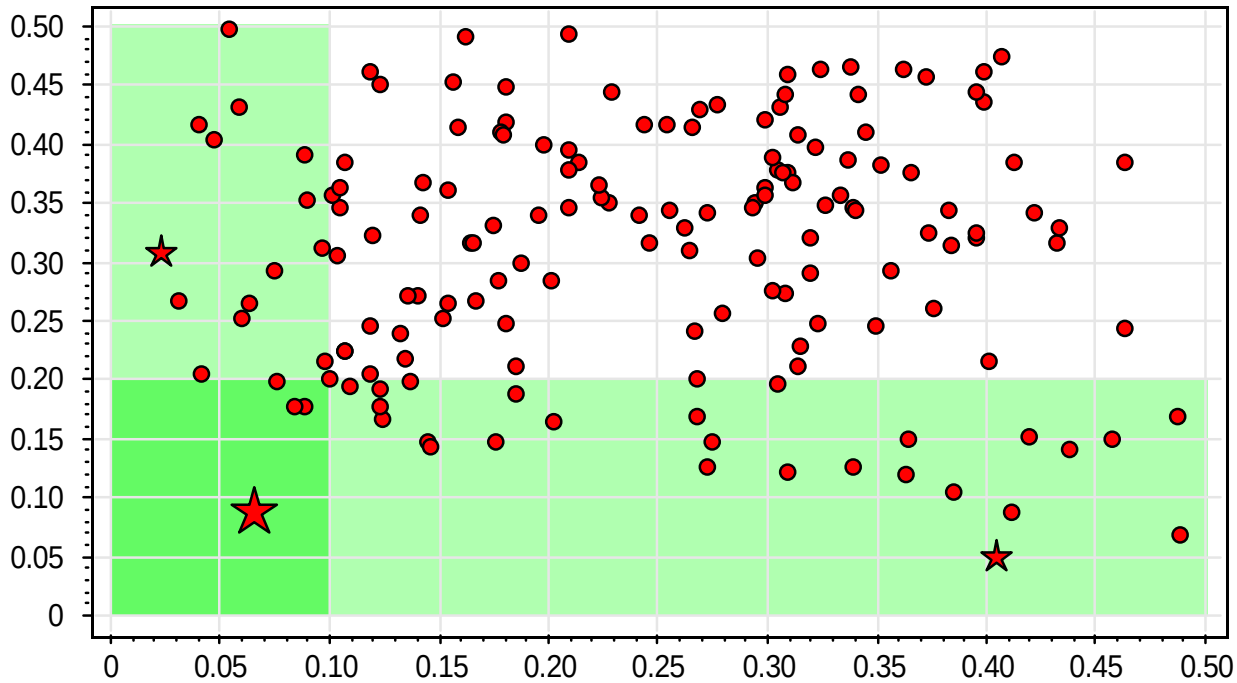
Оценка Вапника-Червоненкиса (VC-bound):

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

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

Двухступенчатый отбор по совокупности внешних критериев

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



Бинарная функция потерь. Матрица ошибок

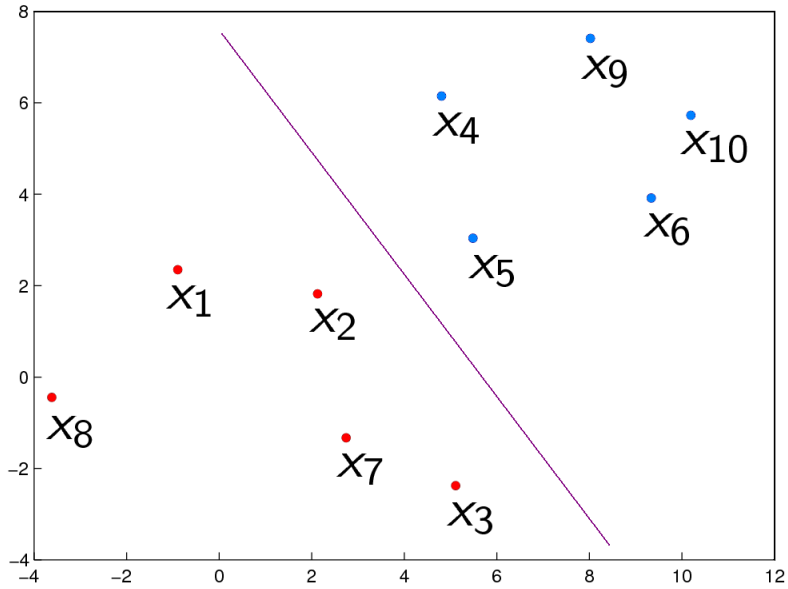
$X^L = \{x_1, \ldots, x_L\}$ — конечное *генеральное множество* объектов;
$A = \{a_1, \ldots, a_D\}$ — конечное семейство *алгоритмов*;
$\mathscr{L}(a, x) \equiv I(a, x) = [\text{алгоритм } a \text{ ошибается на объекте } x]$;

$L \times D$-*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^ℓ — наблюдаемая
$\ldots$	0	0	0	**0**	1	1	$\cdots$	1	(обучающая) выборка
x_ℓ	0	0	1	**0**	0	0	$\cdots$	0	длины ℓ
$x_{\ell+1}$	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;

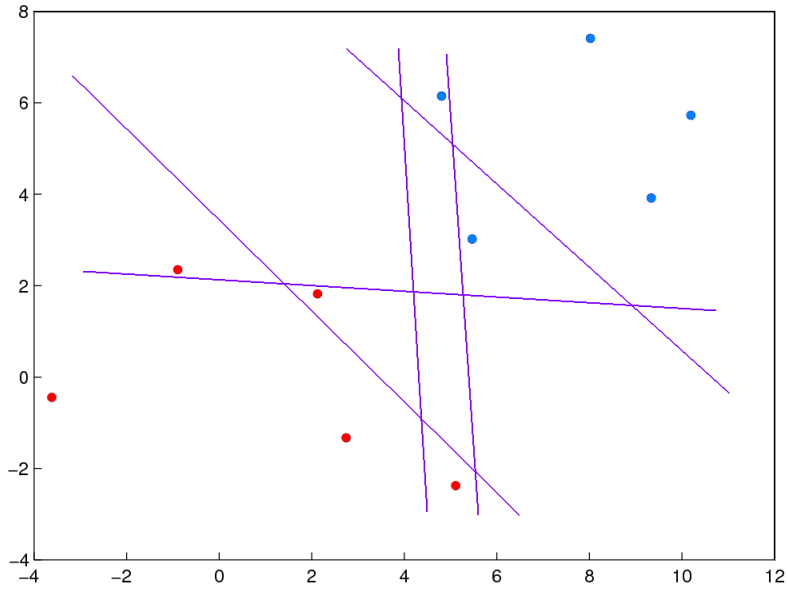
Пример. Матрица ошибок линейных классификаторов



1 вектор с 0 ошибками

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. Матрица ошибок линейных классификаторов

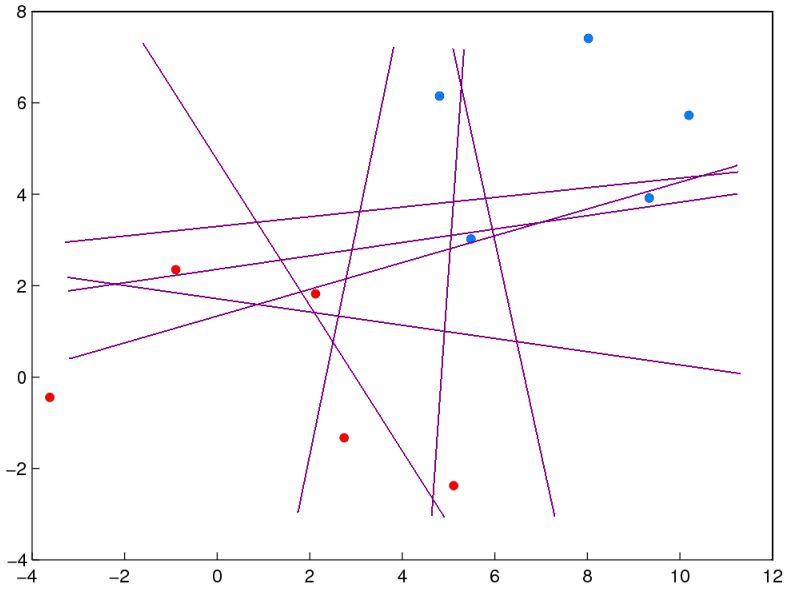


1 вектор с 0 ошибками

5 векторов с 1 ошибкой

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. Матрица ошибок линейных классификаторов



1 вектор с 0 ошибками
5 векторов с 1 ошибкой
8 векторов с 2 ошибками
и т. д...

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	...

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^ℓ:

$$\delta(\mu, X^\ell) = \nu\big(\mu(X^\ell), X^k\big) - \nu\big(\mu(X^\ell), X^\ell\big).$$

Переобучение — это событие $\delta(\mu, X^\ell) \geqslant \varepsilon$.

Основная задача — оценить *вероятность переобучения*:

$$Q_\varepsilon(\mu, X^L) = \mathsf{P}\big[\delta(\mu, X^\ell) \geqslant \varepsilon\big].$$

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:

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

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 $\mathcal{H}_L^{\ell, m}(z) = \sum_{s=0}^{\lfloor z \rfloor} \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell}$ — 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 ℓ шаров наугад.
Какова вероятность того, что s из них чёрные?

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

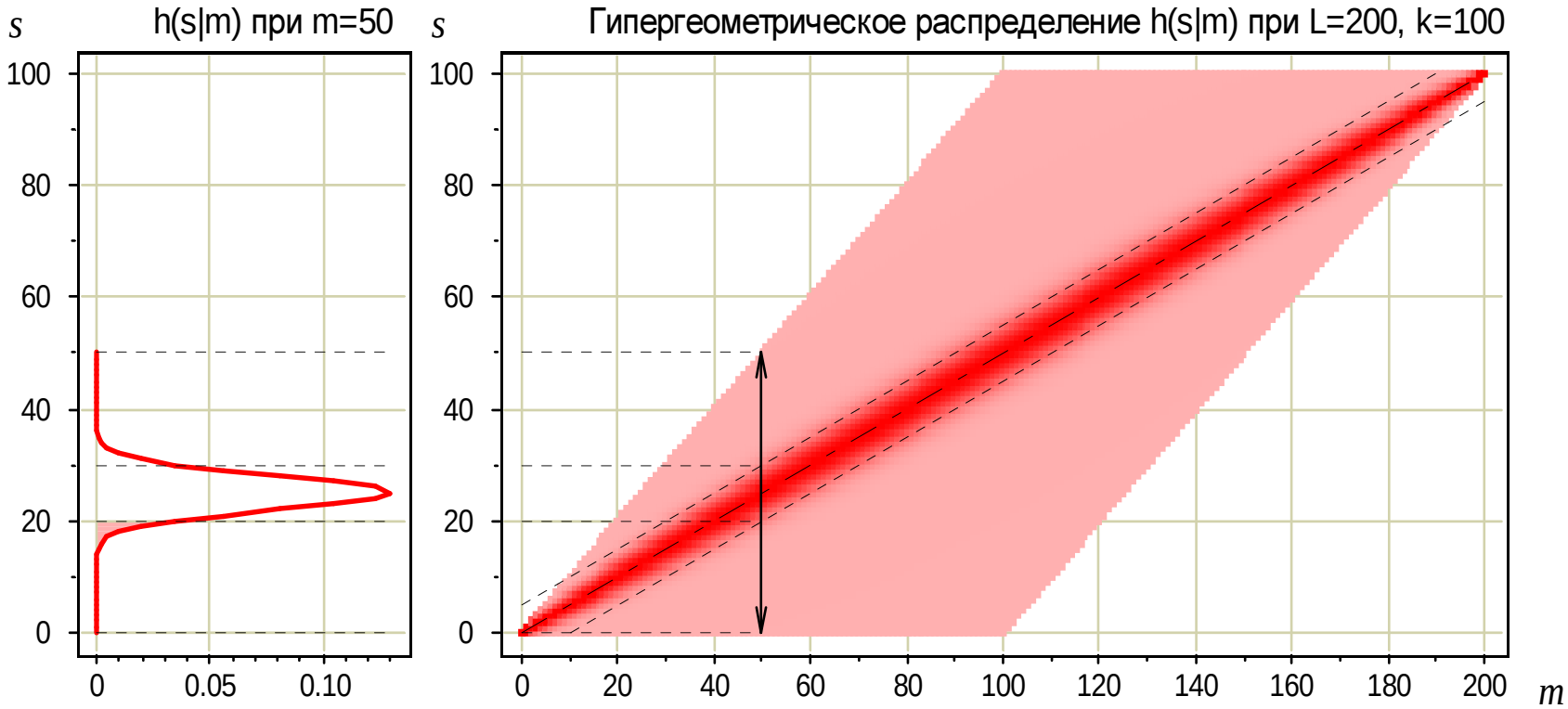
3. Распишем Q_ε, подставив $\nu(a, X^k) = \frac{m-s}{k}$, $\nu(a, X^\ell) = \frac{s}{\ell}$:

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

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

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

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, μ

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

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

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

Теория Вапника–Червоненкиса

Таким образом, доказали важную теорему:

Теорема

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

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

где $m = n(a, X^L)$.

Следствие (Вапник и Червоненкис, 1968)

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

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

Алгоритм полного перебора (Full Search)

Пусть $Q(\mathscr{G})$ — какой-либо внешний критерий.

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

1: $Q^* := +\infty$; — инициализация;
2: **для всех** $j = 1, \ldots, n$, где j — сложность наборов:
3: найти лучший набор сложности j:
$\mathscr{G}_j := \underset{\mathscr{G} \subseteq \mathscr{F}:\ |\mathscr{G}|=j}{\arg\min} Q(\mathscr{G})$;
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)

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;

1: $\mathscr{G}_0 := \varnothing$; $Q^* := +\infty$; — инициализация;
2: **для всех** $j = 1, \ldots, n$, где j — сложность наборов:
3: найти признак, наиболее выгодный для добавления:
$f^* := \arg\min\limits_{f \in \mathscr{F} \setminus \mathscr{G}_{j-1}} Q\left(\mathscr{G}_{j-1} \cup \{f\}\right)$;
4: добавить этот признак в набор:
$\mathscr{G}_j := \mathscr{G}_{j-1} \cup \{f^*\}$;
5: **если** $Q(\mathscr{G}_j) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathscr{G}_j)$;
6: **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

$\mathscr{G}_0 := \varnothing$; $t := 0$; — инициализация;
повторять

$Q^* := +\infty$; — начать добавления Add
пока $|\mathscr{G}_t| < n$
  $t := t + 1$; — началась следующая итерация;
  $f^* := \underset{f \in \mathscr{F} \setminus \mathscr{G}_{t-1}}{\arg\min} Q\big(\mathscr{G}_{t-1} \cup \{f\}\big)$; $\mathscr{G}_t := \mathscr{G}_{t-1} \cup \{f^*\}$;
  если $Q(\mathscr{G}_t) < Q^*$ **то** $t^* := t$; $Q^* := Q(\mathscr{G}_t)$;
  если $t - t^* \geqslant d$ **то прервать цикл**;

$Q^* := +\infty$; — начать добавления Add
пока $|\mathscr{G}_t| > 0$
  $t := t + 1$; — началась следующая итерация;
  $f^* := \underset{f \in \mathscr{G}_{t-1}}{\arg\min} Q\big(\mathscr{G}_{t-1} \setminus \{f\}\big)$; $\mathscr{G}_t := \mathscr{G}_{t-1} \setminus \{f^*\}$;
  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:

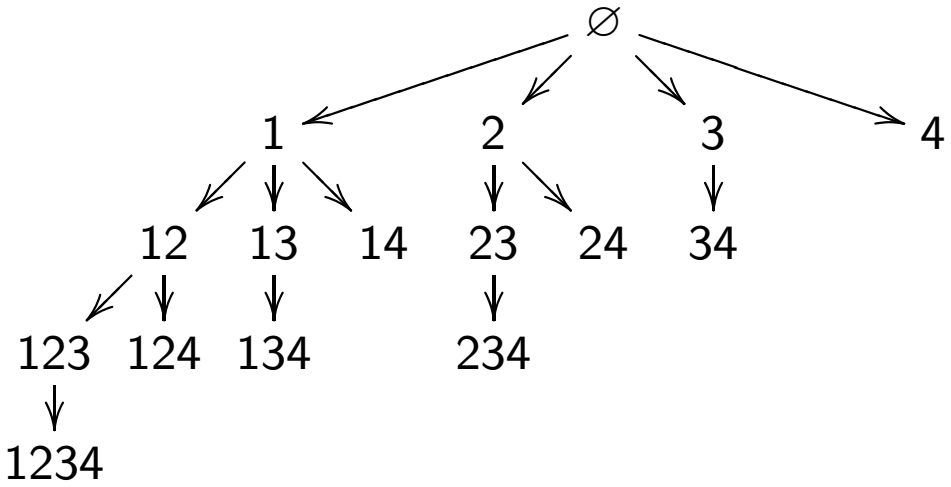
- как правило, лучше, чем Add и Del по отдельности;
- возможны быстрые инкрементные алгоритмы (пример — *шаговая регрессия*).

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: $Q^*_{|\mathscr{G}|} := \min\{Q^*_{|\mathscr{G}|}, Q(\mathscr{G})\}$;
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:
$Q_j^* := +\infty$ для всех $j = 1, \ldots, n$;
7: Упорядочить признаки по убыванию информативности;
8: Нарастить $(\varnothing)$;
9: **вернуть** $\mathscr{G}$, для которого $Q(\mathscr{G}) = \min\limits_{j=1,\ldots,n} Q_j^*$;

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:
$R_1 := \{\{f_1\}, \ldots, \{f_n\}\}$;
2: **для всех** $j = 1, \ldots, n$, где j — сложность наборов:
3: отсортировать ряд $R_j = \{\mathscr{G}_j^1, \ldots, \mathscr{G}_j^{B_j}\}$
по возрастанию критерия: $Q(\mathscr{G}_j^1) \leqslant \ldots \leqslant Q(\mathscr{G}_j^{B_j})$;
4: **если** $B_j > B$ **то**
5: $R_j := \{\mathscr{G}_j^1, \ldots, \mathscr{G}_j^B\}$; — 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:
$R_{j+1} := \{\mathscr{G} \cup \{f\} \mid \mathscr{G} \in R_j,\ f \in \mathscr{F} \setminus \mathscr{G}\}$;

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»);
$R_t := \left\{\mathscr{G}_t^1, \ldots, \mathscr{G}_t^{B_t}\right\}$ — *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:
$B_1 := B$; $R_1 := \{\mathscr{G}_1^1, \ldots, \mathscr{G}_1^{B_1}\}$; $Q^* := +\infty$;
2: **для всех** $t = 1, \ldots, T$, где t — номер поколения:
3: ранжирование индивидов: $Q(\mathscr{G}_t^1) \leqslant \ldots \leqslant Q(\mathscr{G}_t^{B_t})$;
4: **если** $B_t > B$ **то**
5: селекция: $R_t := \{\mathscr{G}_t^1, \ldots, \mathscr{G}_t^B\}$;
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:
$R_{t+1} := \left\{\sim(\mathscr{G}' \times \mathscr{G}'') \;\middle|\; \mathscr{G}', \mathscr{G}'' \in R_t\right\} \cup R_t$;

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).

Генетический (эволюционный) алгоритм

Преимущества:

- 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;

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

$h \approx \frac{1}{rn}$ — 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 регуляризации наиболее эффективны с вычислительной точки зрения.
- Для отбора признаков могут использоваться любые эвристические методы решения задачи дискретной оптимизации

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

- На практике хорошо зарекомендовали себя генетические алгоритмы.
- ГА, МГУА, СПА очень похожи — на их основе легко создавать новые «симбиотические» алгоритмы.