

Оценивание моделей и отбор 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)»

февраль 2011

Содержание

- 1 Задачи выбора модели и отбора признаков**
 - Задача выбора модели
 - Задача отбора признаков
 - Внешние и внутренние критерии
- 2 Критерии выбора модели**
 - Критерии скользящего контроля
 - Критерии непротиворечивости моделей
 - Критерии регуляризации
- 3 Методы отбора признаков**
 - Полный перебор и жадные алгоритмы
 - Поиск в глубину и в ширину
 - Стохастический поиск

Задача выбора модели (model selection)

Дано:

X — пространство объектов; Y — множество ответов;

$X^\ell = (x_i, y_i)_{i=1}^\ell$ — обучающая выборка, $y_i = y^*(x_i)$;

$A_t = \{a: X \rightarrow Y\}$ — модели алгоритмов, $t = 1, \dots, T$;

$\mu_t: (X \times Y)^\ell \rightarrow A_t$ — методы обучения, $t = 1, \dots, T$.

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)

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

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:

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

Внешние и внутренние критерии

Внутренний критерий — это качество на обучении X^ℓ :

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

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

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

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

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.

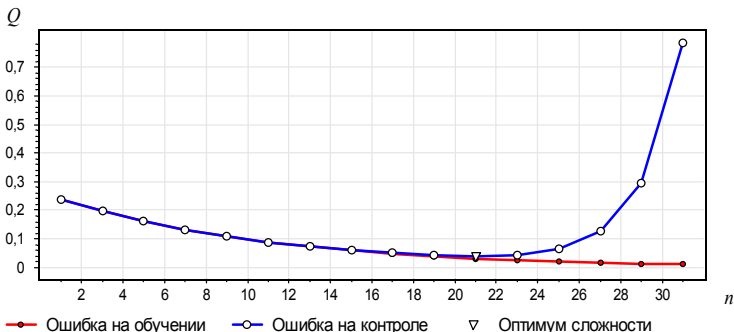
Внешние и внутренние критерии

Внутренний критерий — это качество на обучении X^ℓ :

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

Внешний критерий — это качество на новых данных.

Внешний критерий позволяет найти оптимум сложности модели.



Скольльзящий контроль

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

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

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

Преимущества CCV:

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

Недостатки CCV:

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

Скольльзящий контроль «leave-one-out»

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

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

Преимущества LOO:

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

Недостатки LOO:

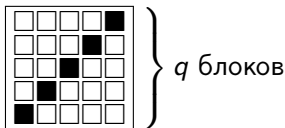
- каждый объект *лишь один раз* участвует в контроле;
- ресурсоёмкость;
- высокая дисперсия.

Скользящий контроль с разбиением по блокам

Контроль по q блокам (q -fold CV):

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

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



Преимущества q -fold CV:

- компромисс между LOO и hold-out.

Недостатки q -fold CV:

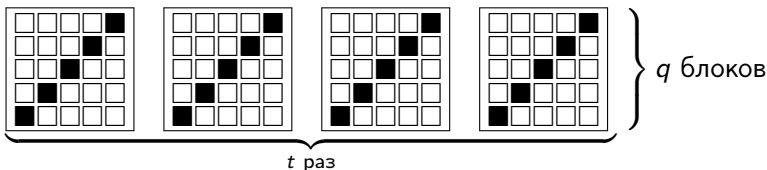
- каждый объект лишь один раз участвует в контроле.

Скользящий контроль с многократным разбиением по блокам

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

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

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



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 $t \geq 20$);

Критерии непротиворечивости моделей

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

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

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

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

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

Недостатки:

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

Критерии регуляризации

Регуляризатор — аддитивная добавка к функционалу, штраф за сложность (complexity penalty) модели A :

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

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):

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

Разновидности L_0 -регуляризации

Информационный критерий Акаике (Akaike Information Criterion):

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

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

Байесовский информационный критерий (Bayes Inform. Criterion):

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

Оценка Вапника-Червоненкиса (VC-bound):

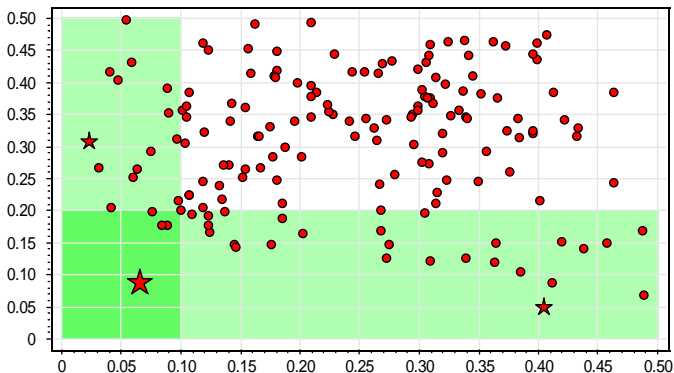
$$Q_{\text{ext}}(\mu, X^\ell) = Q_{\text{int}}(\mu, X^\ell) + \sqrt{\frac{h}{\ell} \left(\ln \frac{2\ell}{h} + 1 \right) - \frac{\ln \eta}{\ell}},$$

h — VC-размерность; для линейных, опять-таки, $h = |\mathcal{G}|$;

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.



Алгоритм полного перебора (Full Search)

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

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

- 1: $Q^* := +\infty$; — инициализация;
- 2: **для всех** $j = 1, \dots, n$, где j — сложность наборов:
- 3: найти лучший набор сложности j :
$$\mathcal{G}_j := \arg \min_{\mathcal{G} \subseteq \mathcal{F}: |\mathcal{G}|=j} Q(\mathcal{G});$$
- 4: **если** $Q(\mathcal{G}_j) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathcal{G}_j)$;
- 5: **если** $j - j^* \geq d$ **то вернуть** \mathcal{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 оооооочень долго — $O(2^n)$.

Способы устранения:

- эвристические методы сокращённого перебора.

Алгоритм жадного добавления

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

- 1: $\mathcal{G}_0 := \emptyset$; $Q^* := +\infty$; — инициализация;
- 2: **для всех** $j = 1, \dots, n$, где j — сложность наборов:
- 3: найти признак, наиболее выгодный для добавления:
$$f^* := \arg \min_{f \in \mathcal{F} \setminus \mathcal{G}_{j-1}} Q(\mathcal{G}_{j-1} \cup \{f\});$$
- 4: добавить этот признак в набор:
$$\mathcal{G}_j := \mathcal{G}_{j-1} \cup \{f^*\};$$
- 5: **если** $Q(\mathcal{G}_j) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathcal{G}_j)$;
- 6: **если** $j - j^* \geq d$ **то вернуть** \mathcal{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

1: $\mathcal{G}_0 := \emptyset$; $t := 0$; — инициализация;

2: **повторять**

3: $Q^* := +\infty$; — начать добавления Add

4: **пока** $|\mathcal{G}_t| < n$

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

6: $f^* := \arg \min_{f \in \mathcal{F} \setminus \mathcal{G}_{t-1}} Q(\mathcal{G}_{t-1} \cup \{f\})$; $\mathcal{G}_t := \mathcal{G}_{t-1} \cup \{f^*\}$;

7: **если** $Q(\mathcal{G}_t) < Q^*$ **то** $t^* := t$; $Q^* := Q(\mathcal{G}_t)$;

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

9: $Q^* := +\infty$; — начать добавления Add

10: **пока** $|\mathcal{G}_t| > 0$

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

12: $f^* := \arg \min_{f \in \mathcal{G}_{t-1}} Q(\mathcal{G}_{t-1} \setminus \{f\})$; $\mathcal{G}_t := \mathcal{G}_{t-1} \setminus \{f^*\}$;

13: **если** $Q(\mathcal{G}_t) < Q^*$ **то** $t^* := t$; $Q^* := Q(\mathcal{G}_t)$;

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

15: **пока** значения критерия $Q(\mathcal{G}_{t^*})$ уменьшаются;

16: **вернуть** \mathcal{G}_{t^*} ;

Алгоритм поочерёдного добавления и удаления

Преимущества:

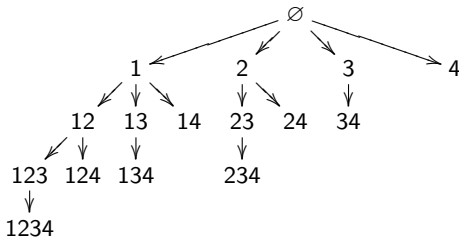
- как правило, лучше, чем Add и Del по отдельности;
- возможны быстрые инкрементные алгоритмы (пример — *шаговая регрессия*).

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 \mathcal{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 \mathcal{G} :
набор \mathcal{G} не наращивается, если

$$\exists j: \quad Q(\mathcal{G}) \geq \kappa Q_j^* \quad \text{и} \quad |\mathcal{G}| \geq j + d,$$

$d \geq 0$ — целочисленный параметр,

$\kappa \geq 1$ — вещественный параметр.

Чем меньше d и κ , тем сильнее сокращается перебор.

Поиск в глубину (метод ветвей и границ)

Вход: множество \mathcal{F} , критерий Q , параметры d и κ ;

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

- 6: Инициализация массива лучших значений критерия:
 $Q_j^* := +\infty$ для всех $j = 1, \dots, n$;
- 7: Упорядочить признаки по убыванию информативности;
- 8: Нарастить (\emptyset);
- 9: **вернуть** \mathcal{G} , для которого $Q(\mathcal{G}) = \min_{j=1, \dots, n} Q_j^*$;

Поиск в ширину

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

Философский принцип *неокончателных 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:

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

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

Поиск в ширину

Вход: множество \mathcal{F} , критерий Q , параметры d, B ;

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

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 $I_j(f)$:

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

Генетический алгоритм поиска (идея и терминология)

$\mathcal{G} \subseteq \mathcal{F}$ — индивид (в МГУА «модель»);

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

$\beta = (\beta_j)_{j=1}^n$, $\beta_j = [f_j \in \mathcal{G}]$ — хромосома, кодирующая \mathcal{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 \mathcal{F} , критерий Q , параметры: d, p_m ,
 B — размер популяции, T — число поколений;

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

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:

- 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*:

$$R_{t+1} := \{\sim \mathcal{G}, \dots, \sim \mathcal{G} \mid \mathcal{G} \in R_t\} \cup R_t;$$

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 \mathcal{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 $\{p_1, \dots, p_n\}$:
 $R_{jt} := \{\mathcal{G}_{jt}^1, \dots, \mathcal{G}_{jt}^r\}$, $|\mathcal{G}_{jt}^1| = \dots = |\mathcal{G}_{jt}^r| = j$;
- 5: $\mathcal{G}_{jt}^{\min} := \arg \min_{\mathcal{G} \in R_{jt}} Q(\mathcal{G})$; — лучший из r наборов;
- 6: $\mathcal{G}_{jt}^{\max} := \arg \max_{\mathcal{G} \in R_{jt}} Q(\mathcal{G})$; — худший из r наборов;
- 7: $H := 0$; наказание для всех $f_s \in \mathcal{G}_{jt}^{\max}$:
 $\Delta p_s := \min\{p_s, h\}$; $p_s := p_s - \Delta p_s$; $H := H + \Delta p_s$;
- 8: поощрение для всех $f_s \in \mathcal{G}_{jt}^{\min}$: $p_s := p_s + H/j$;
- 9: $\mathcal{G}_j := \arg \min_{\mathcal{G} \in R_{j1}, \dots, R_{jT}} Q(\mathcal{G})$; — лучший набор сложности j ;
- 10: **если** $Q(\mathcal{G}_j) < Q^*$ **то** $j^* := j$; $Q^* := Q(\mathcal{G}_j)$;
- 11: **если** $j - j^* \geq d$ **то вернуть** \mathcal{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;

$h \approx \frac{1}{m}$ — 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

$$Q(\mathcal{G}) \rightarrow \min_{\mathcal{G} \subseteq \mathcal{F}}$$

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