Логические алгоритмы классификации

Воронцов Константин Вячеславович
vokov@forecsys.ru
http://www.MachineLearning.ru/wiki?title=User:Vokov

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

Видеолекции: http://shad.yandex.ru/lectures

ШАД Яндекс • 21 февраля 2017

Содержание

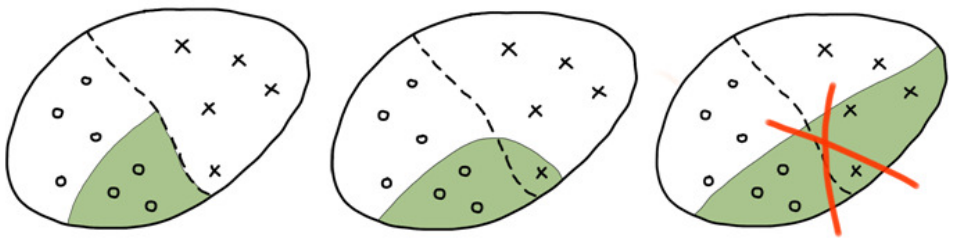
Логическая закономерность

$X^\ell = (x_i, y_i)_{i=1}^\ell \subset X \times Y$ — обучающая выборка, $y_i = y(x_i)$.

Логическая закономерность (правило, rule) — это предикат $R\colon X \to \{0,1\}$, удовлетворяющий двум требованиям:

1. *интерпретируемость*:
   1) R записывается на естественном языке;
   2) R зависит от небольшого числа признаков (1–7);
2. *информативность* относительно одного из классов $c \in Y$:
   $p_c(R) = \#\{x_i\colon R(x_i)=1 \text{ и } y_i=c\} \to \max$;
   $n_c(R) = \#\{x_i\colon R(x_i)=1 \text{ и } y_i\neq c\} \to \min$;

Если $R(x) = 1$, то говорят «R выделяет x» (R covers x).



Требование интерпретируемости

1) $R(x)$ записывается на естественном языке;
2) $R(x)$ зависит от небольшого числа признаков (1–7);

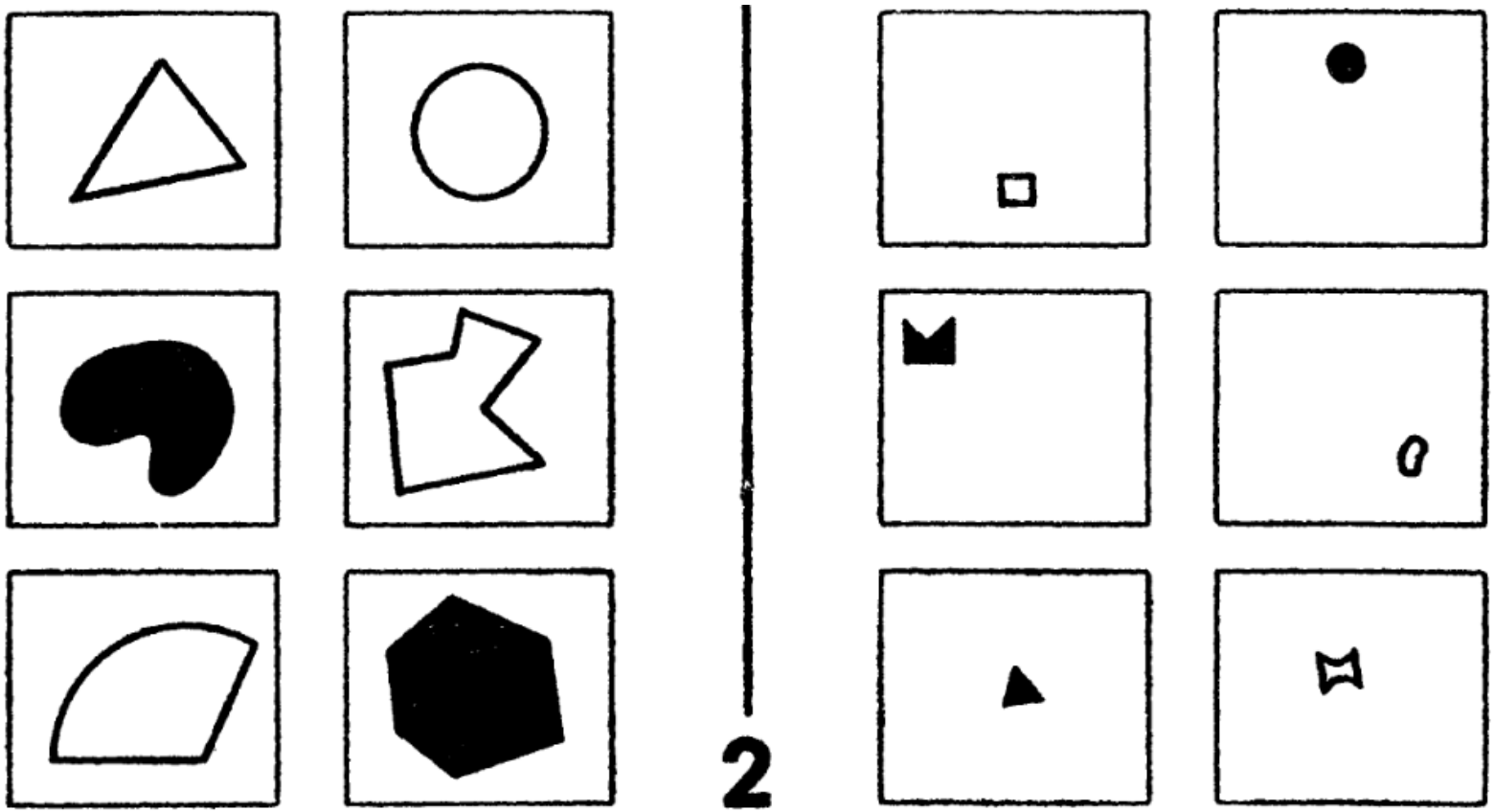
Пример (из области медицины)

Если *«возраст* $>$ *60»* ***и*** *«пациент ранее перенёс инфаркт»,*
то *операцию не делать, риск отрицательного исхода 60%.*

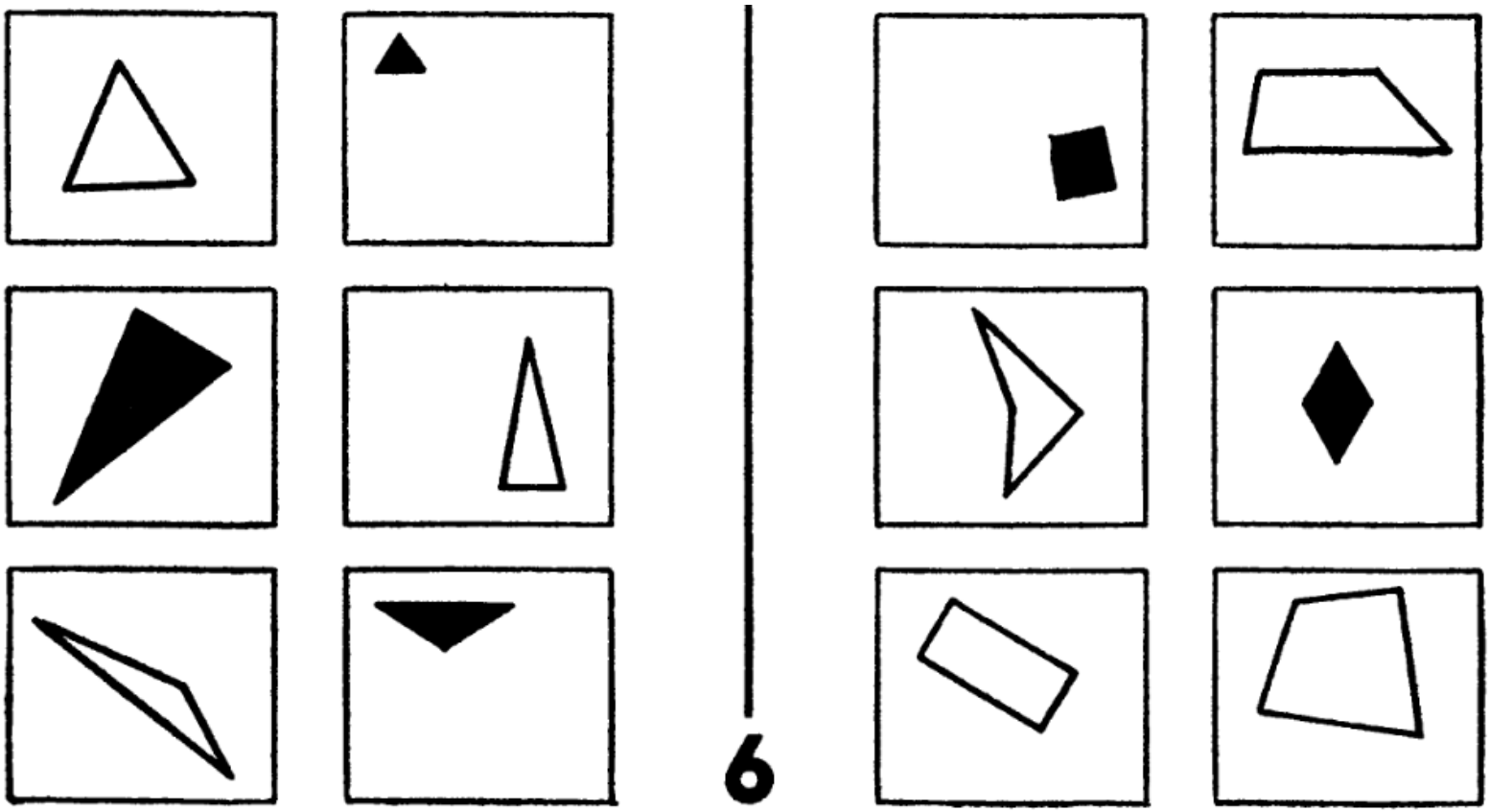
Пример (из области кредитного скоринга)

Если *«в анкете указан домашний телефон»*
и *«зарплата* $>$ *\$2000»* ***и*** *«сумма кредита* $<$ *\$5000»*
то *кредит можно выдать, риск дефолта 5%.*

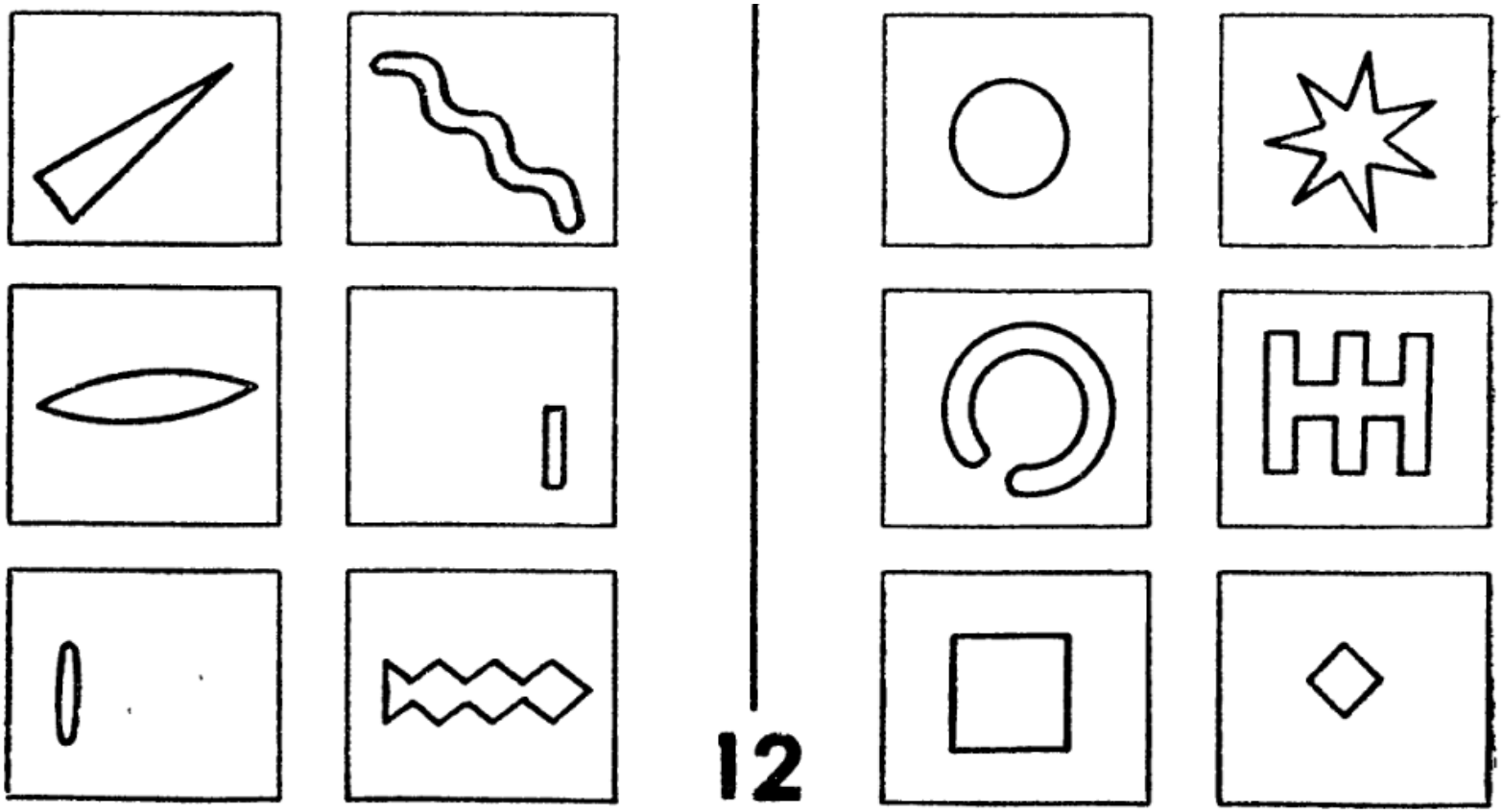
Тесты М. М. Бонгарда [Проблема узнавания, 1967]



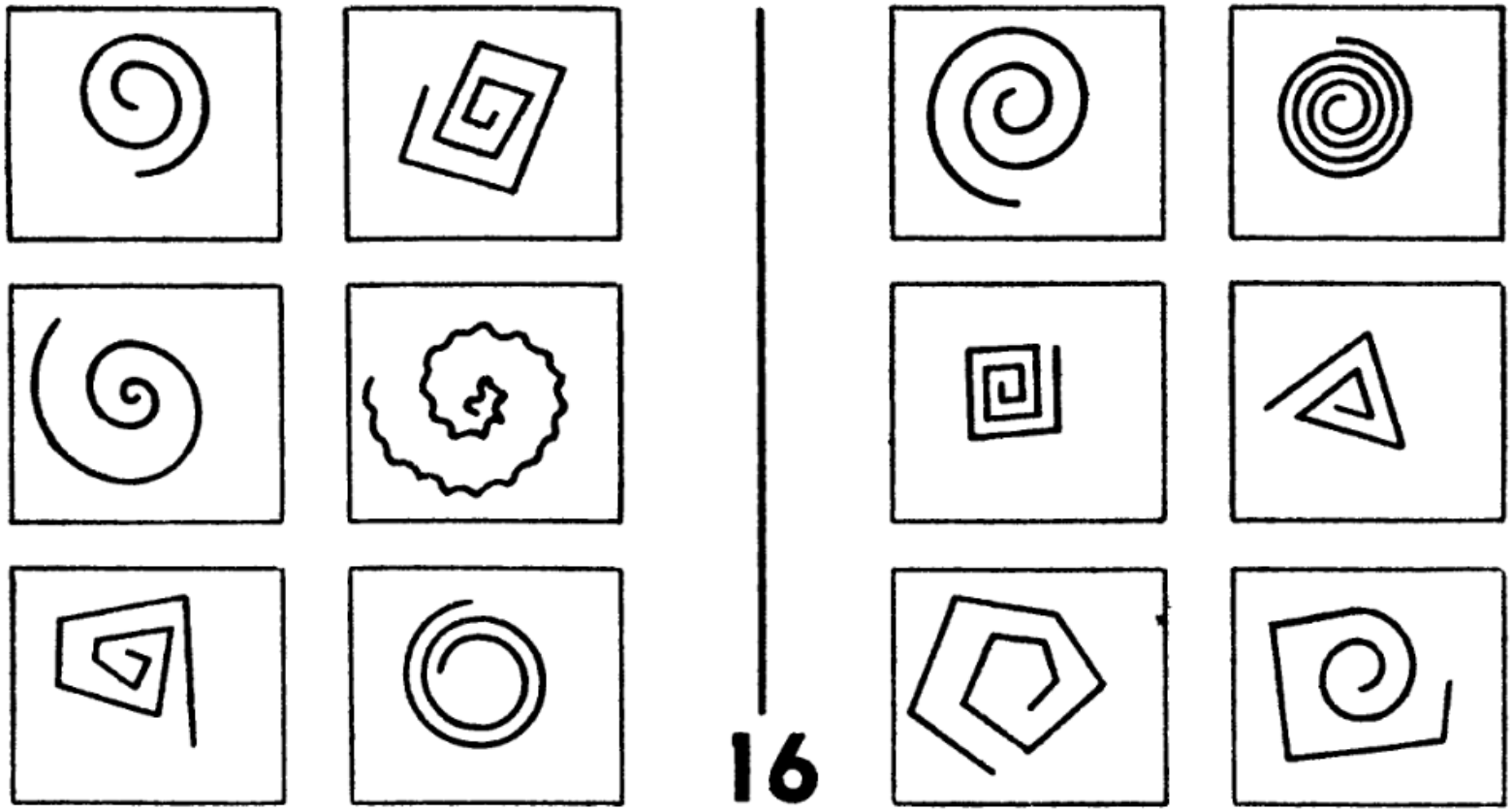
Тесты М. М. Бонгарда [Проблема узнавания, 1967]



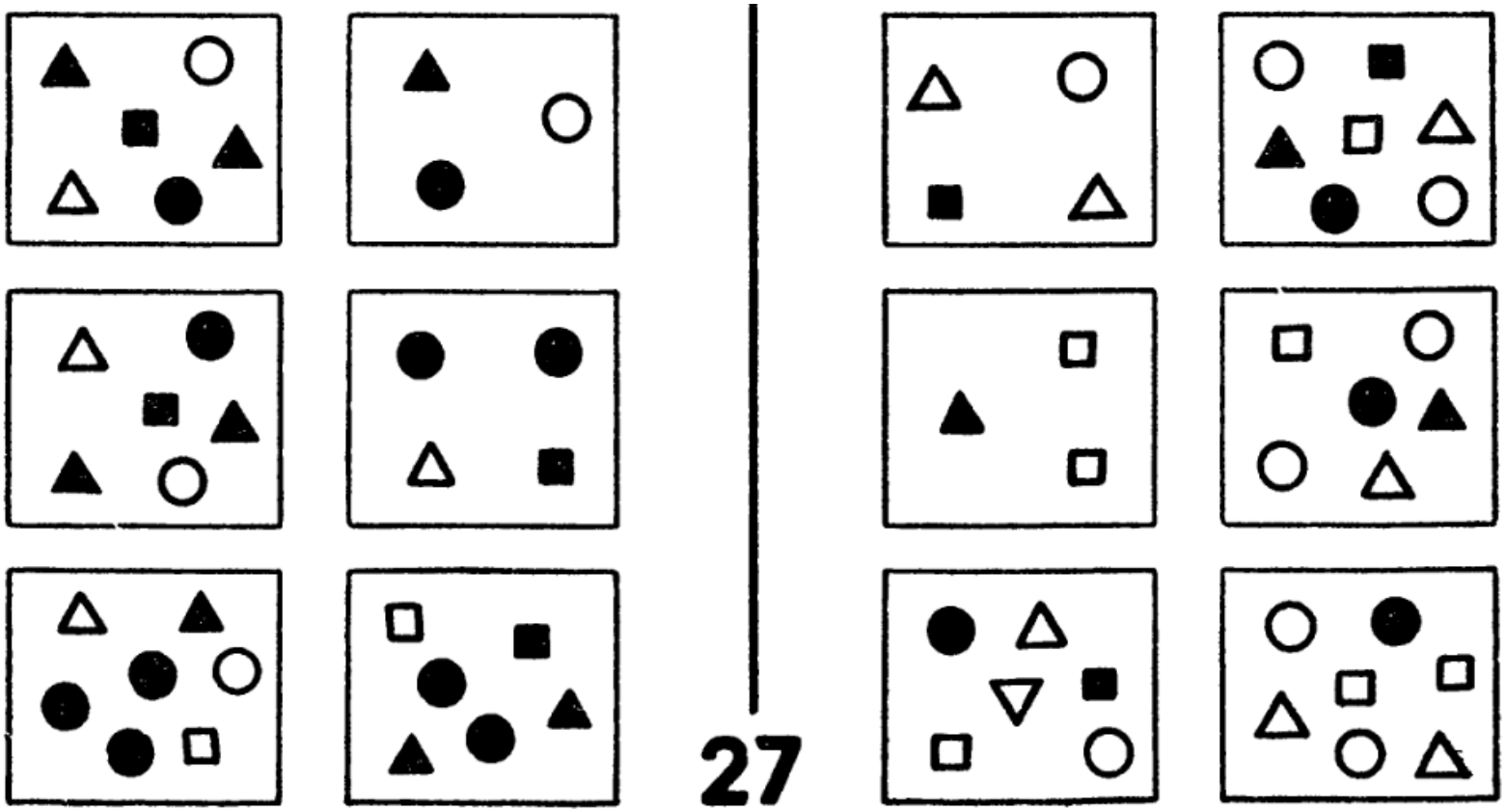
Тесты М. М. Бонгарда [Проблема узнавания, 1967]



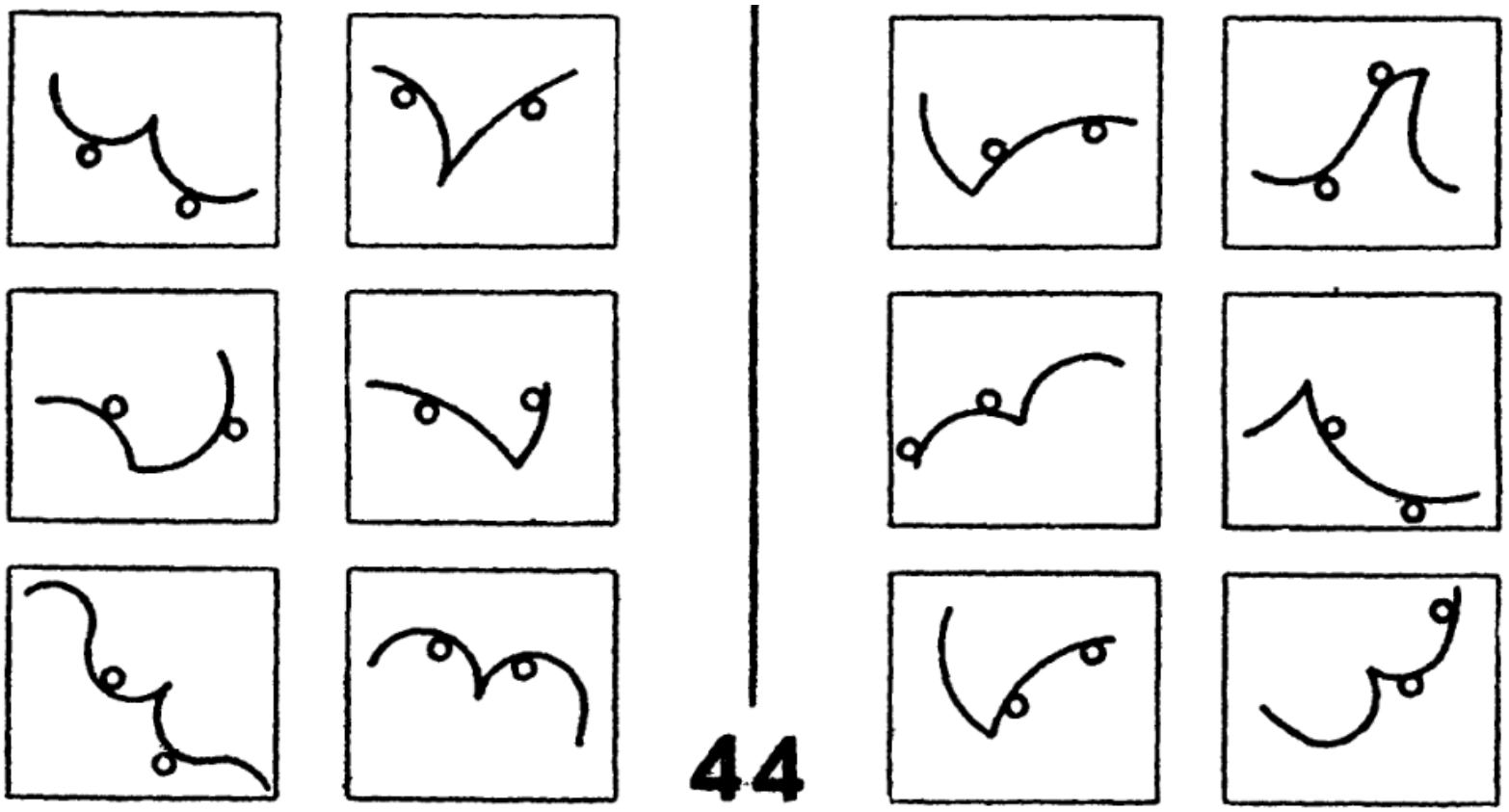
Тесты М. М. Бонгарда [Проблема узнавания, 1967]



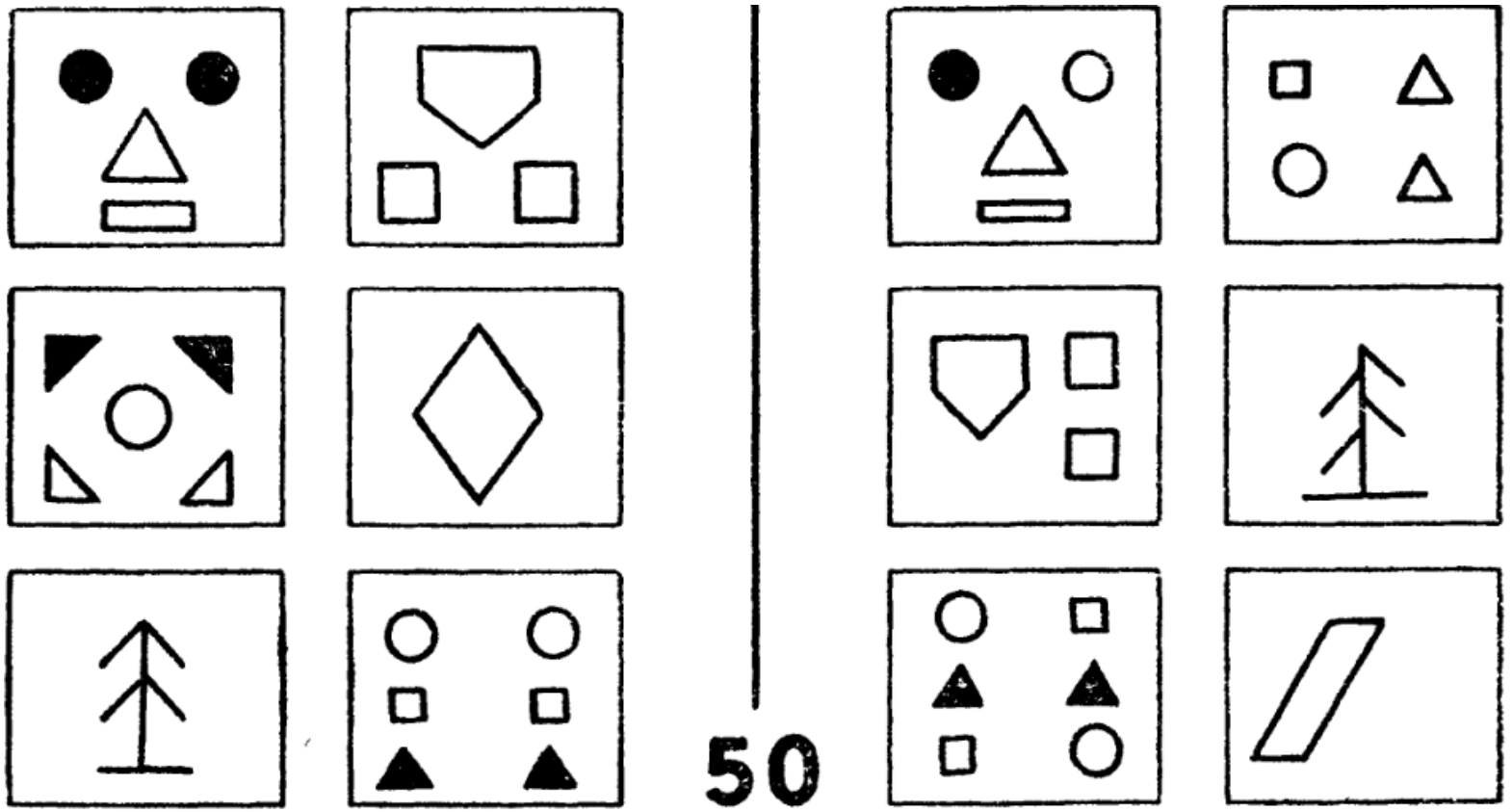
Тесты М. М. Бонгарда [Проблема узнавания, 1967]



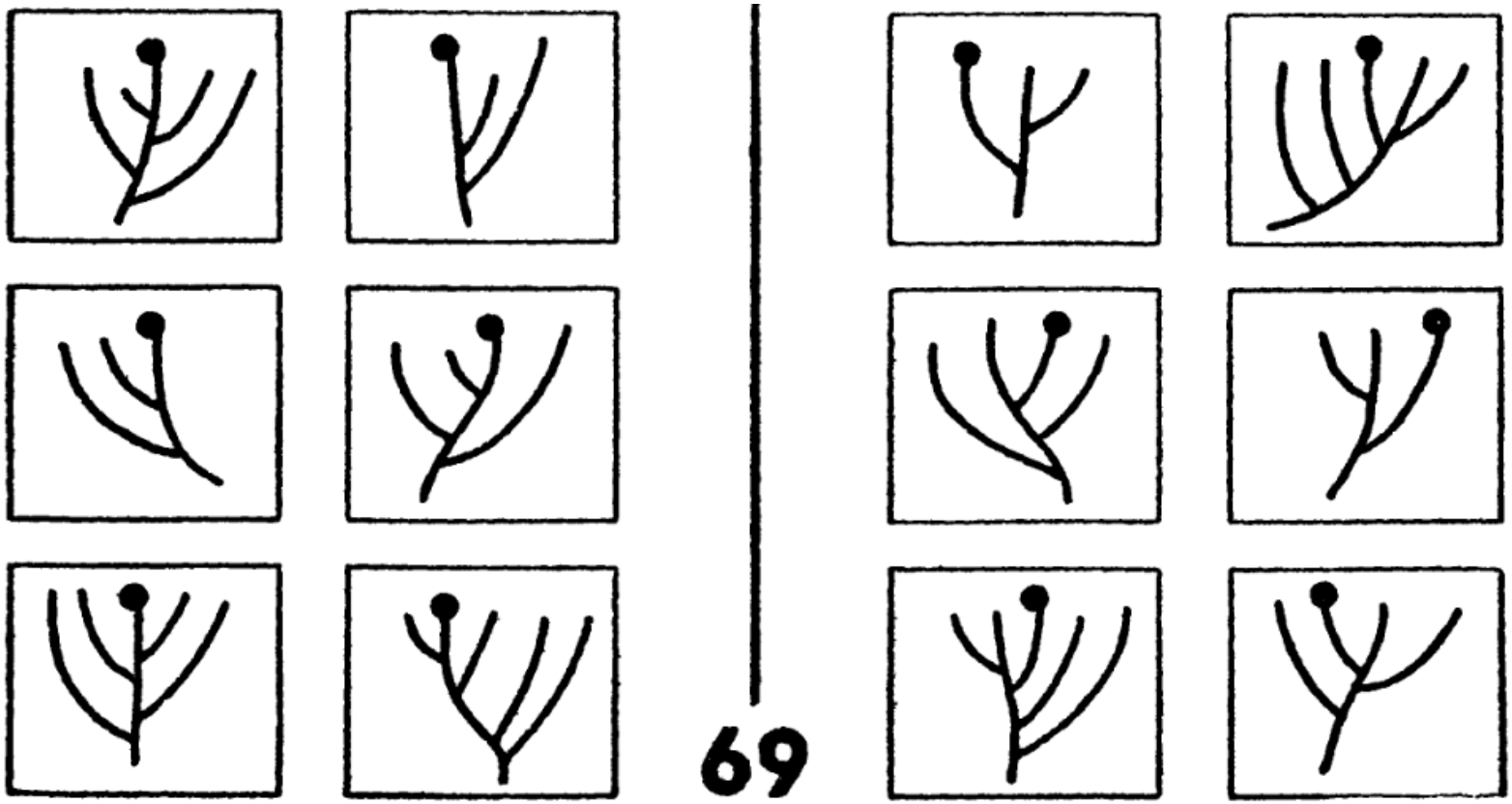
Тесты М. М. Бонгарда [Проблема узнавания, 1967]



Тесты М. М. Бонгарда [Проблема узнавания, 1967]



Тесты М. М. Бонгарда [Проблема узнавания, 1967]



Основные вопросы построения логических алгоритмов

1. Как изобретать признаки $f_1(x), \ldots, f_n(x)$?
   — не наука, а искусство (размышления, озарения, эксперименты, консультации, мозговые штурмы,...)
2. Какого вида закономерности $R(x)$ нам нужны?
   — простые формулы от малого числа признаков
3. Как определять информативность?
   — так, чтобы одновременно $p \to \max$, $n \to \min$
4. Как искать закономерности?
   — перебором подмножеств признаков
5. Как объединять закономерности в алгоритм?
   — любым классификатором ($R(x)$ — новые признаки)

Закономерность — интерпретируемый высокоинформативный одноклассовый классификатор с отказами.

Часто используемые виды закономерностей

1. Пороговое условие (решающий пень, decision stump):

$$R(x) = \big[f_j(x) \leqslant a_j\big] \text{ или } \big[a_j \leqslant f_j(x) \leqslant b_j\big].$$

2. *Конъюнкция* пороговых условий:

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

3. *Синдром* — выполнение не менее d условий из $|J|$,
(при $d = |J|$ это конъюнкция, при $d = 1$ — дизъюнкция):

$$R(x) = \left[\sum_{j \in J} \big[a_j \leqslant f_j(x) \leqslant b_j\big] \geqslant d\right],$$

Параметры J, a_j, b_j, d настраиваются по обучающей выборке путём оптимизации *критерия информативности*.

Часто используемые виды закономерностей

4. *Полуплоскость* — линейная пороговая функция:

$$R(x) = \Big[\sum_{j\in J} w_j f_j(x) \geqslant w_0\Big].$$

5. *Шар* — пороговая функция близости:

$$R(x) = \big[\rho(x, x_0) \leqslant w_0\big],$$

АВО — алгоритмы вычисления оценок [Ю. И. Журавлёв, 1971]:

$$\rho(x, x_0) = \max_{j\in J} w_j \big|f_j(x) - f_j(x_0)\big|.$$

SCM — машины покрывающих множеств [M. Marchand, 2001]:

$$\rho(x, x_0) = \sum_{j\in J} w_j \big|f_j(x) - f_j(x_0)\big|^{\gamma}.$$

Параметры J, w_j, w_0, x_0 настраиваются по обучающей выборке путём оптимизации *критерия информативности*.

Схема локального поиска информативных закономерностей

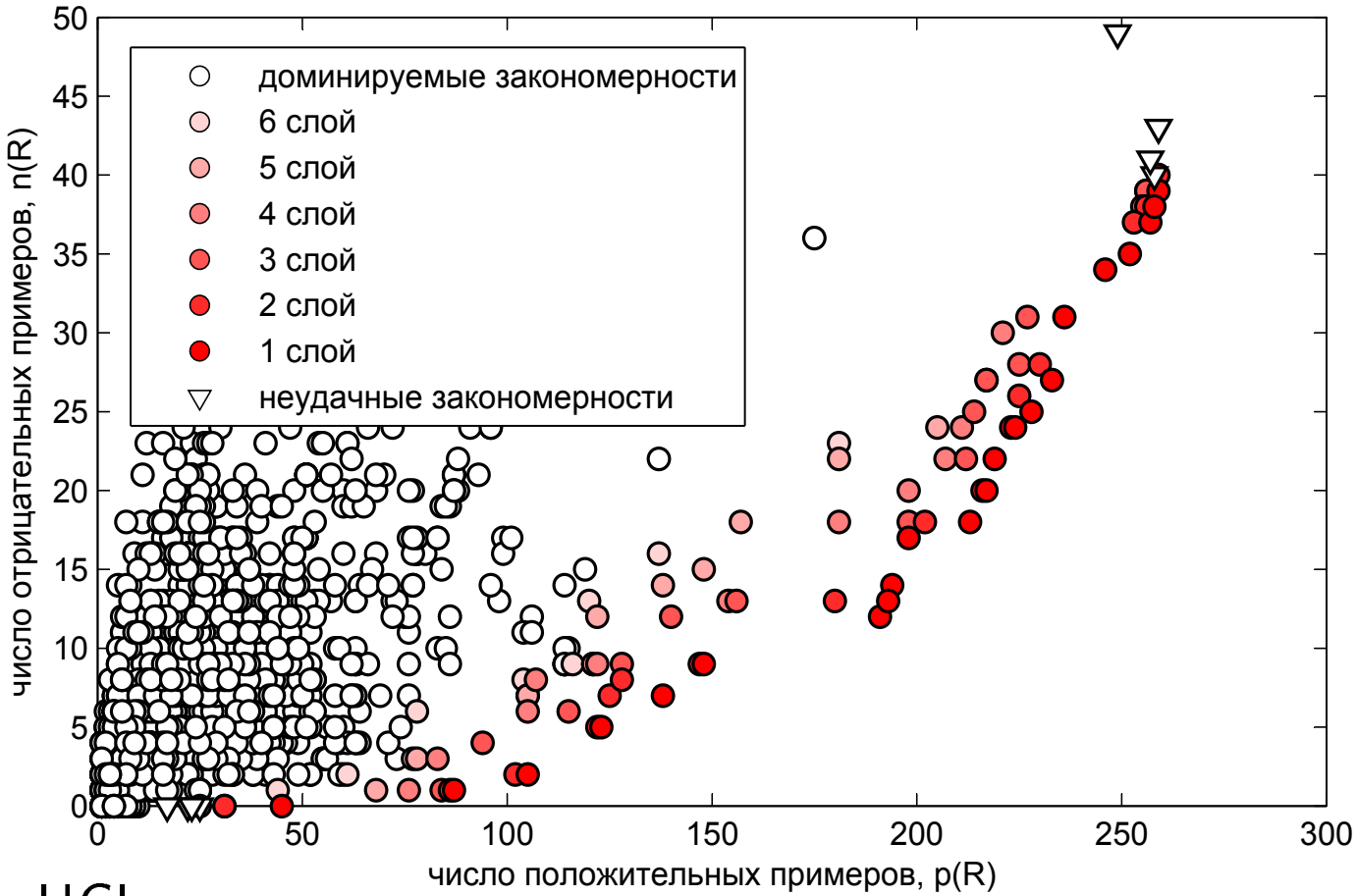
Вход: выборка X^ℓ;
Выход: множество закономерностей Z;

1: начальное множество правил Z;
2: **повторять**
3: $\quad Z' :=$ множество модификаций правил $R \in Z$;
4: $\quad$ удалить слишком похожие правила из $Z \cup Z'$;
5: $\quad$ оценить информативность всех правил $R \in Z'$;
6: $\quad Z :=$ наиболее информативные правила из $Z \cup Z'$;
7: **пока** правила продолжают улучшаться
8: **вернуть** Z.

Частные случаи:
— стохастический локальный поиск (SLS)
— генетические (эволюционные) алгоритмы
— метод ветвей и границ

Отбор закономерностей по информативности в (p, n)-плоскости

Парето-фронт — множество недоминируемых закономерностей (точка недоминируема, если правее и ниже неё точек нет)



задача UCI:german

Определение решающего дерева (Decision Tree)

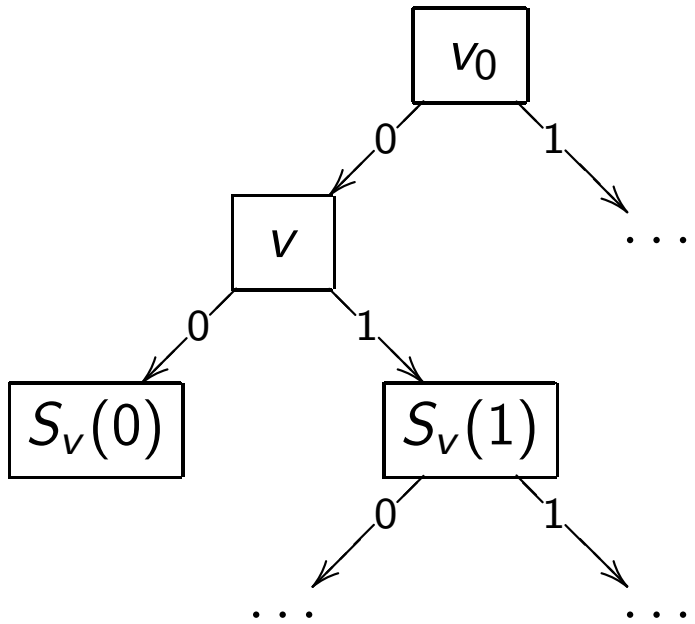
Решающее дерево — алгоритм классификации $a(x)$, задающийся *деревом* (связным ациклическим графом):

1) $V = V_{\text{внутр}} \sqcup V_{\text{лист}}$, $v_0 \in V$ — корень дерева;
2) $v \in V_{\text{внутр}}$: функции $f_v : X \to D_v$ и $S_v : D_v \to V$, $|D_v| < \infty$;
3) $v \in V_{\text{лист}}$: метка класса $y_v \in Y$.

1: $v := v_0$;
2: **пока** $v \in V_{\text{внутр}}$
3: $\quad v := S_v(f_v(x))$;
4: **вернуть** y_v;

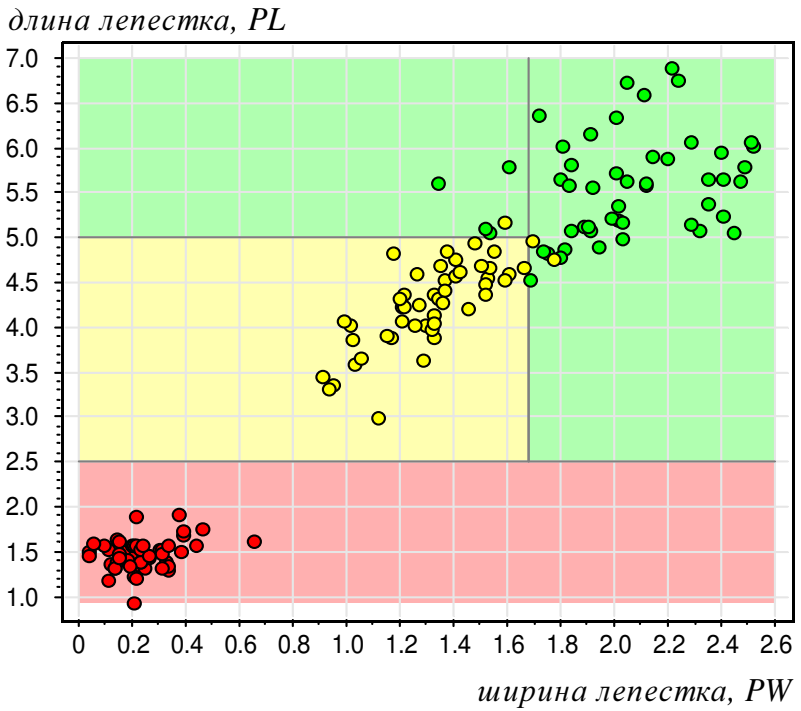
Частный случай: $D_v \equiv \{0, 1\}$ — бинарное решающее дерево

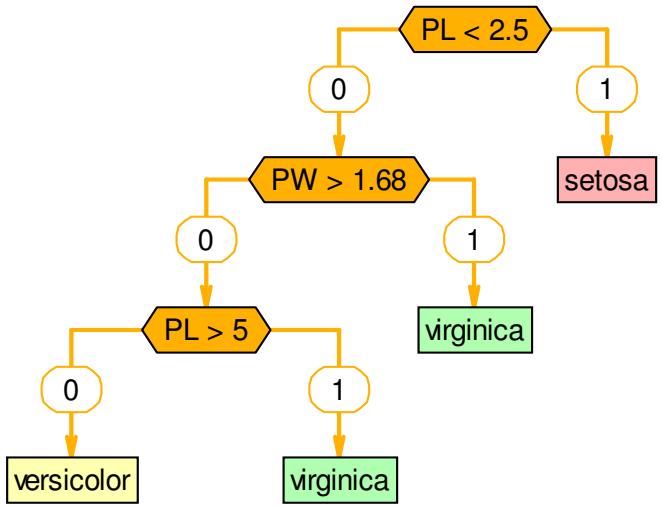
Пример: $f_v(x) = [f_j(x) \geqslant \theta_j]$



Пример решающего дерева

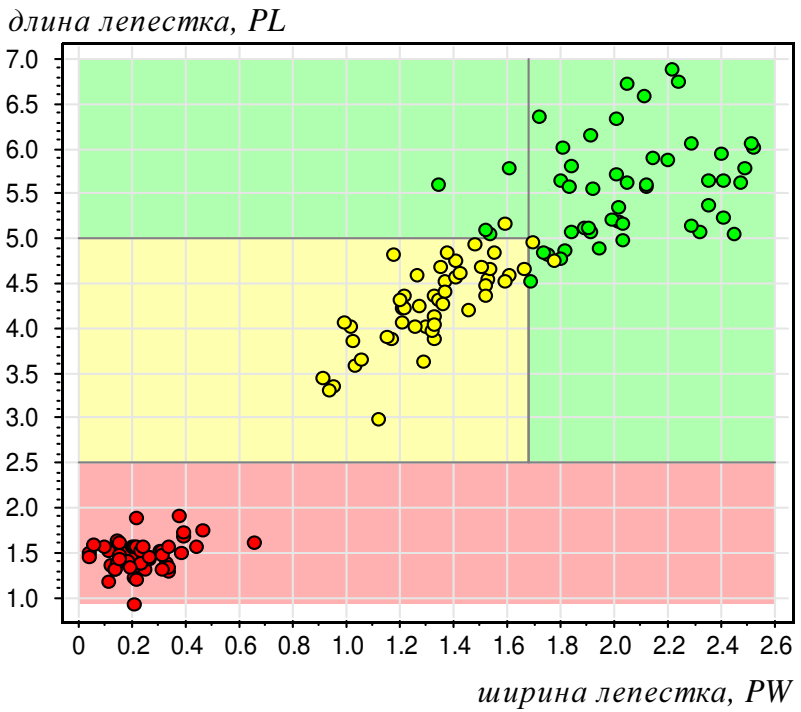
Задача Фишера о классификации цветков ириса на 3 класса, в выборке по 50 объектов каждого класса, 4 признака.

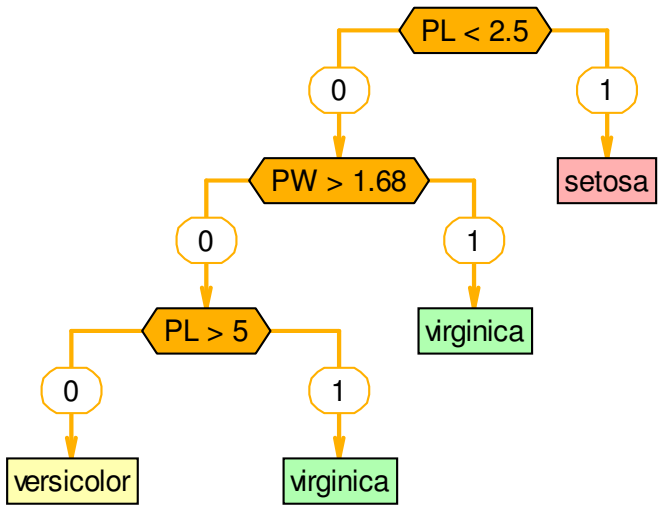




На графике: в осях двух самых информативных признаков (из 4) два класса разделились без ошибок, на третьем 3 ошибки.

Решающее дерево → покрывающий набор конъюнкций





setosa	$r_1(x) = [PL \leqslant 2.5]$
virginica	$r_2(x) = [PL > 2.5] \wedge [PW > 1.68]$
virginica	$r_3(x) = [PL > 5] \wedge [PW \leqslant 1.68]$
versicolor	$r_4(x) = [PL > 2.5] \wedge [PL \leqslant 5] \wedge [PW < 1.68]$

Обучение решающего дерева: стратегия «разделяй и властвуй»

$v_0 := \text{TreeGrowing}\ (X^\ell)$;

1: **ФУНКЦИЯ** TreeGrowing $(U \subseteq X^\ell) \mapsto$ корень дерева v;
2: **если** StopCriterion (U) **то**
3: **вернуть** новый лист v, взяв $y_v := \text{Major}\,(U)$;
4: найти признак, наиболее выгодный для ветвления дерева:
$f_v := \arg\max\limits_{f \in F} \text{Gain}\,(f, U)$;
5: **если** $\text{Gain}\,(f_v, U) < G_0$ **то**
6: **вернуть** новый лист v, взяв $y_v := \text{Major}\,(U)$;
7: создать новую внутреннюю вершину v с функцией f_v;
8: **для всех** $k \in D_v$
$U_k := \{x \in U\colon f_v(x) = k\}$, $S_v(k) := \text{TreeGrowing}\ (U_k)$,
9: **вернуть** v;

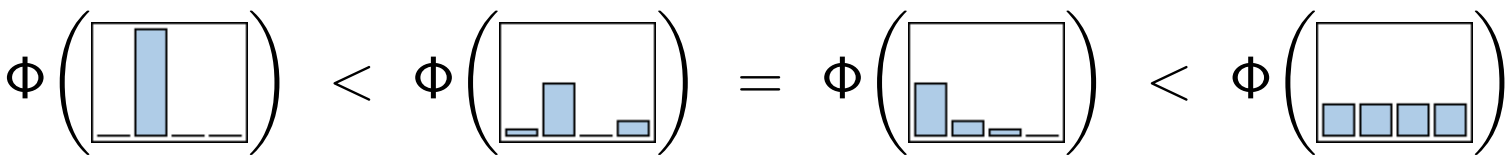
Мажоритарное правило: $\text{Major}\,(U) := \arg\max\limits_{y \in Y} P(y|U)$.

Мера неопределённости распределения

Частотная оценка вероятности класса y в вершине $v \in V_{\text{внутр}}$:

$$p_y \equiv P(y|x \in U) = \frac{1}{|U|} \sum_{x_i \in U} [y_i = y], \quad \forall x \in U$$

$\Phi(U)$ — мера *неопределённости* (impurity) распределения p_y:



1) минимальна, когда $p_y \in \{0, 1\}$,
2) максимальна, когда $p_y = \frac{1}{|Y|}$ для всех $y \in Y$,
3) симметрична: не зависит от перенумерации классов.

$$\Phi(U) = \sum_{y \in Y} p_y \mathscr{L}(p_y) = \frac{1}{|U|} \sum_{x_i \in U} \mathscr{L}\big(P(y_i|x_i \in U)\big) \to \min,$$

где $\mathscr{L}(p)$ убывает и $\mathscr{L}(1) = 0$, например: $-\log p$, $1-p$, $1-p^2$

Критерий ветвления

Неопределённость распределения $P(y_i|x_i \in U_{f(x_i)})$, после ветвления вершины v по признаку f и разбиения $U = \bigsqcup\limits_{k \in D_v} U_k$:

$$\Phi(U_1, \ldots, U_{|D_v|}) = \frac{1}{|U|} \sum_{k \in D_v} \sum_{x_i \in U_k} \mathscr{L}\big(P(y_i|x_i \in U_k)\big) =$$

$$= \sum_{k \in D_v} \frac{|U_k|}{|U|} \Phi(U_k)$$

Выигрыш от ветвления вершины v:

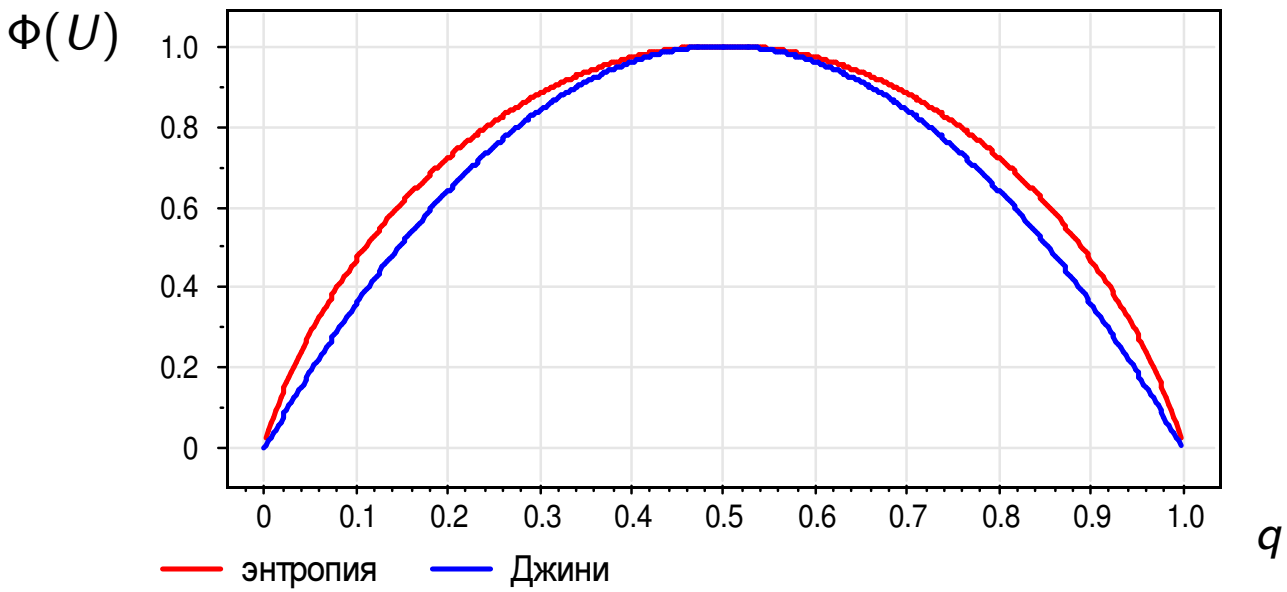
$$\text{Gain}\,(f, U) = \Phi(U) - \Phi(U_1, \ldots, U_{|D_v|}) =$$

$$= \Phi(U) - \sum_{k \in D_v} \frac{|U_k|}{|U|} \Phi(U_k) \to \max_{f \in F}$$

Критерий Джини и энтропийный критерий

Два класса, $Y = \{0, 1\}$, $P(y|x \in U) = \begin{cases} q, & y=1 \\ 1-q, & y=0 \end{cases}$

- Если $\mathscr{L}(p) = -\log_2 p$, то
  $\Phi(U) = -q \log_2 q - (1-q)\log_2(1-q)$ — *энтропия* выборки.
- Если $\mathscr{L}(p) = 2(1-p)$, то
  $\Phi(U) = 4q(1-q)$ — *неопределённость Джини* (Gini impurity).



Обработка пропущенных значений

На стадии обучения:

- $f_v(x_i)$ не определено $\Rightarrow$ x_i исключается из U для $\text{Gain}\,(f_v, U)$
- $q_{vk} = \frac{|U_k|}{|U|}$ — оценка вероятности k-й ветви, $v \in V_{\text{внутр}}$
- $P(y|x,v) = \frac{1}{|U|} \sum\limits_{x_i \in U} [y_i = y]$ для всех $v \in V_{\text{лист}}$

На стадии классификации:

- $f_v(x)$ определено $\Rightarrow$ из дочерней $s = S_v(f_v(x))$ взять $P(y|x,v) = P(y|x,s)$.
- $f_v(x)$ не определено $\Rightarrow$ *пропорциональное распределение*: $P(y|x,v) = \sum\limits_{k \in D_v} q_{vk} P(y|x, S_v(k))$.
- Окончательное решение — наиболее вероятный класс: $a(x) = \arg\max\limits_{y \in Y} P(y|x, v_0)$.

Жадная нисходящая стратегия: достоинства и недостатки

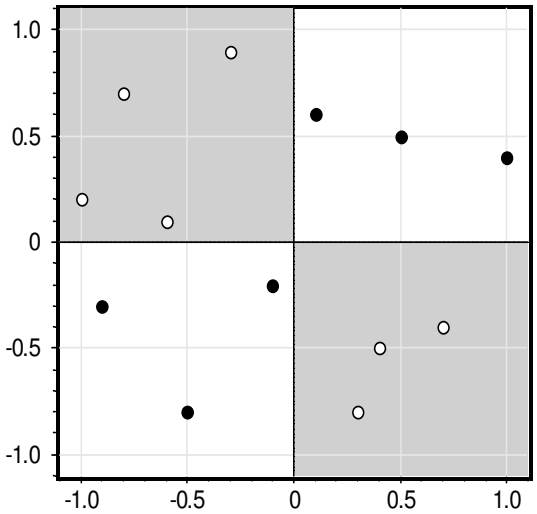
Достоинства:

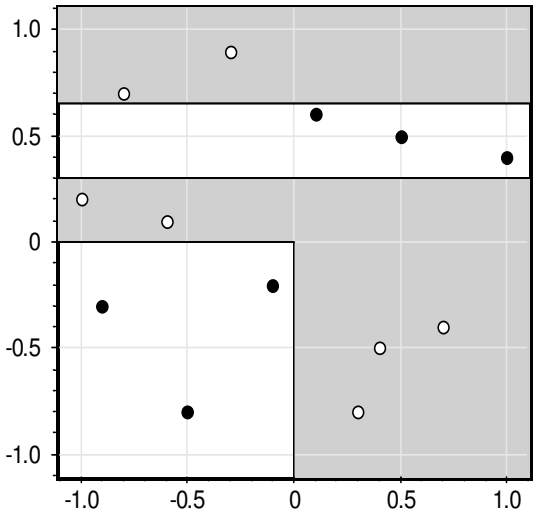
- Интерпретируемость и простота классификации.
- Гибкость: можно варьировать множество F.
- Допустимы разнотипные данные и данные с пропусками.
- Трудоёмкость линейна по длине выборки $O(|F|h\ell)$.
- Не бывает отказов от классификации.

Недостатки:

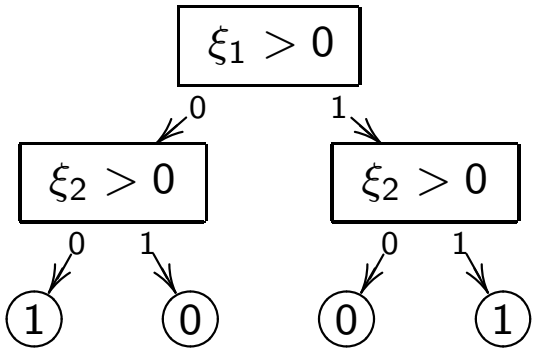
- Жадная стретегия переусложняет структуру дерева, и, как следствие, сильно переобучается.
- Фрагментация выборки: чем дальше v от корня, тем меньше статистическая надёжность выбора f_v, y_v.
- Высокая чувствительность к шуму, к составу выборки, к критерию информативности.

Жадная стратегия переусложняет структуру дерева

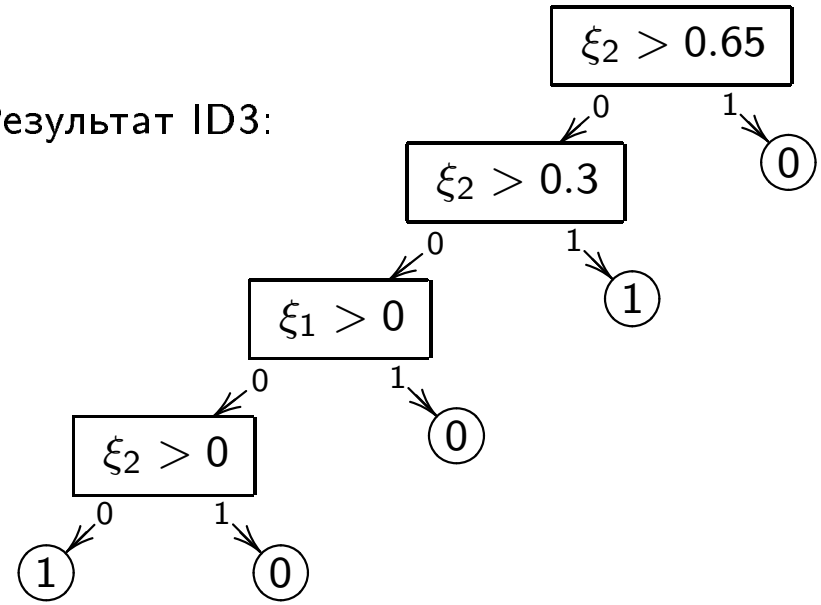




Оптимальное дерево для задачи XOR:



Результат ID3:



Усечение дерева (pruning)

X^q — независимая контрольная выборка, $q \approx 0.5\ell$.

1: **для всех** $v \in V_{\text{внутр}}$
2: $X_v^q :=$ подмножество объектов X^q, дошедших до v;
3: **если** $X_v^q = \varnothing$ **то**
4: **вернуть** новый лист v, $y_v := \text{Major}\,(U)$;
5: число ошибок при классификации X_v^q разными способами:
$\text{Err}(v)$ — поддеревом, растущим из вершины v;
$\text{Err}_k(v)$ — дочерним поддеревом $S_v(k)$, $k \in D_v$;
$\text{Err}_c(v)$ — классом $c \in Y$.
6: в зависимости от того, какое из них минимально:
сохранить поддерево v;
заменить поддерево v дочерним $S_v(k)$;
заменить поддерево v листом, $y_v := \arg\min\limits_{c \in Y} \text{Err}_c(v)$.

CART: деревья регрессии и классификации

Обобщение на случай регрессии: $Y = \mathbb{R}$, $y_v \in \mathbb{R}$.

U — множество объектов x_i, дошедших до вершины v

Мера неопределённости — среднеквадратичная ошибка

$$\Phi(U) = \min_{y \in Y} \sum_{x_i \in U} (y - y_i)^2$$

Значения в терминальных вершинах — МНК-решение:

$$y_v = \frac{1}{|U|} \sum_{x_i \in U} y_i$$

Дерево регрессии $a(x)$ — это кусочно-постоянная функция.

CART: критерий Minimal Cost-Complexity Pruning

Среднеквадратичная ошибка со штрафом за сложность дерева

$$C_\alpha = \sum_{i=1}^{\ell} \big(a(x_i) - y_i\big)^2 + \alpha |V_{\text{лист}}| \to \min$$

При увеличении α дерево последовательно упрощается.
Причём последовательность вложенных деревьев единственна.

Из этой последовательности выбирается дерево с минимальной ошибкой на тестовой выборке (Hold-Out).

Для случая классификации используется аналогичная стратегия усечения, с критерием Джини.

Небрежные решающие деревья (Oblivious Decision Tree, ODT)

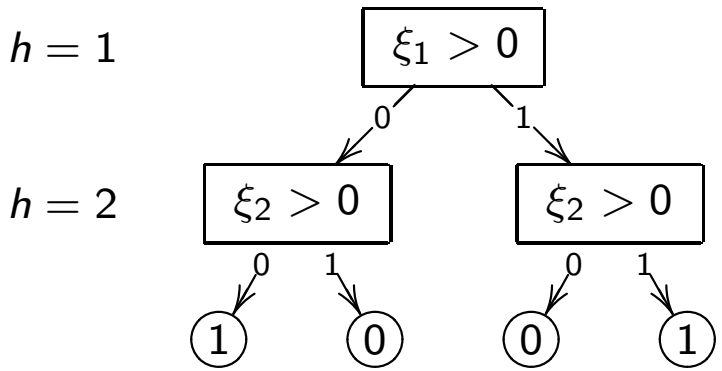
Решение проблемы фрагментации:

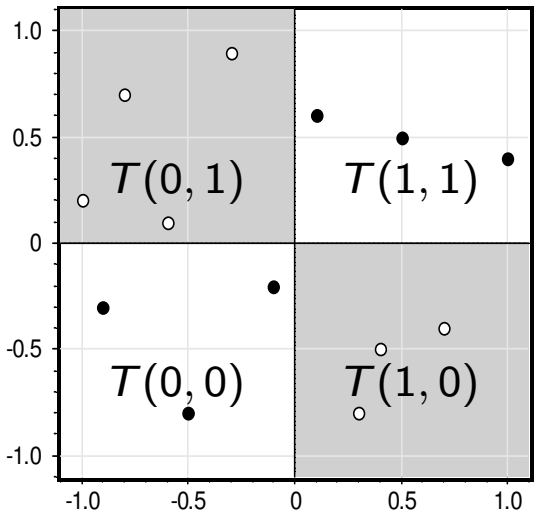
строится сбалансированное дерево глубины H, $D_v = \{0,1\}$;
для всех узлов уровня h условие ветвления $f_h(x)$ *одинаково*;
на уровне h ровно 2^{h-1} вершин; X делится на 2^H ячеек.

Классификатор задаётся *таблицей решений* $T\colon \{0,1\}^H \to Y$:

$$a(x) = T\big(f_1(x), \ldots, f_H(x)\big).$$

Пример: задача XOR, $H = 2$.





Алгоритм обучения ODT

Вход: выборка X^ℓ; множество признаков F; глубина дерева H;
Выход: признаки f_h, $h = 1, \ldots, H$; таблица $T: \{0,1\}^H \to Y$;

1: **для всех** $h = 1, \ldots, H$
2: предикат с максимальным выигрышем определённости:
$f_h := \arg\max_{f \in F} \text{Gain}\,(f_1, \ldots, f_{h-1}, f)$;
3: классификация по мажоритарному правилу:
$T(\beta) := \text{Major}\,(U_{H\beta})$;

Выигрыш от ветвления на уровне h по всей выборке X^ℓ:

$$\text{Gain}\,(f_1, \ldots, f_h) = \Phi(X^\ell) - \sum_{\beta \in \{0,1\}^h} \frac{|U_{h\beta}|}{\ell} \Phi(U_{h\beta}),$$

$$U_{h\beta} = \left\{x_i \in X^\ell : f_s(x_i) = \beta_s,\ s = 1..h\right\},\quad \beta = (\beta_1, \ldots, \beta_h) \in \{0,1\}^h.$$

Случайный лес (Random Forest)

Голосование деревьев классификации, $Y = \{-1, +1\}$:

$$a(t) = \text{sign}\,\frac{1}{T}\sum_{t=1}^{T} b_t(x).$$

Голосование деревьев регрессии, $Y = \mathbb{R}$:

$$a(t) = \frac{1}{T}\sum_{t=1}^{T} b_t(x).$$

- каждое дерево $b_t(x)$ обучается по случайной выборке с возвращениями
- в каждой вершине признак выбирается из случайного подмножества $\sqrt{n}$ признаков ($\lfloor n/3 \rfloor$ для регрессии)
- признаки и пороги выбираются по критерию Джини
- усечений (pruning) нет

Резюме в конце лекции

- Основные требования к логическим закономерностям:
  - интерпретируемость, информативность, различность.
- Преимущества решающих деревьев:
  - интерпретируемость,
  - допускаются разнотипные данные,
  - возможность обхода пропусков;
- Недостатки решающих деревьев:
  - переобучение,
  - фрагментация,
  - неустойчивость к шуму, составу выборки, критерию;
- Способы устранения этих недостатков:
  - редукция,
  - композиции (леса) деревьев.

Yandex MatrixNet = голосование (градиентный бустинг) над ODT.