Логические алгоритмы классификации

Воронцов Константин Вячеславович
vokov@forecsys.ru

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

сентябрь 2013

Задача обучения по прецедентам

X — множество *объектов*;
Y — множество *ответов*;
$y^*: X \to Y$ — неизвестная зависимость (target function).

Дано:
$X^\ell = \{x_1, \ldots, x_\ell\} \subset X$ — *обучающая выборка* (training sample);
$y_i = y^*(x_i), \;\; i = 1, \ldots, \ell$ — известные ответы.

Найти:
$a: X \to Y$ — алгоритм, решающую функцию (decision function), приближающую y^* на всём множестве X.

Весь курс машинного обучения — это конкретизация:

- как задаются объекты и какими могут быть ответы;
- как строится функция a;
- что значит «a приближает y^* на всём X».

Объекты и признаки

$f_j\colon X \to D_j,\ \ j = 1, \ldots, n$ — признаки объектов.

Типы признаков:

- $D_j = \{0, 1\}$ — *бинарный* признак f_j;
- $|D_j| < \infty$ — *номинальный* признак f_j;
- $|D_j| < \infty$, D_j упорядочено — *порядковый* признак f_j;
- $D_j = \mathbb{R}$ — *количественный* признак f_j.

Вектор $\big(f_1(x), \ldots, f_n(x)\big)$ — *признаковое описание* объекта x.

Данные — это матрица «*объекты–признаки*» и вектор ответов:

$$F = \big\|f_j(x_i)\big\|_{\ell \times n} = \begin{pmatrix} f_1(x_1) & \ldots & f_n(x_1) \\ \ldots & \ldots & \ldots \\ f_1(x_\ell) & \ldots & f_n(x_\ell) \end{pmatrix} \qquad y = \big\|y_i\big\|_\ell = \begin{pmatrix} y_1 \\ \ldots \\ y_\ell \end{pmatrix}$$

Ответы, типы задач, функционал качества

- $Y = \{1, \ldots, M\}$ — задача *классификации* на M непересекающихся классов.
- $Y = \{0, 1\}^M$ — задача классификации на M классов, которые могут пересекаться.
- $Y = \mathbb{R}$ — задача восстановления *регрессии*.
- Y — конечное упорядоченное множество
  — задача *ранжирования* или ранговой регрессии.

Функционалы качества:

- Для задач классификации — число ошибок:
$$Q(a, X^\ell) = \sum_{i=1}^{\ell} \big[a(x_i) \neq y_i\big] \to \min_a .$$
- Для задач регрессии — средняя квадратичная ошибка:
$$Q(a, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \big(a(x_i) - y_i\big)^2 \to \min_a .$$

Содержание

Логические алгоритмы классификации (2 лекции)

Основное понятие — «закономерность»

X — пространство объектов;
Y — множество ответов;
$f_1(x), \ldots, f_n(x)$ — признаки;
$X^\ell = (x_i, y_i)_{i=1}^\ell$ — обучающая выборка, $y_i = y(x_i)$.

Определение (пока неформальное)

Закономерность (правило, rule) — это предикат $\varphi\colon X \to \{0,1\}$, *удовлетворяющий двум требованиям:*

1) интерпретируемость (φ зависит от 1–7 признаков);

2) информативность относительно класса $c \in Y$*:*

$$p_c(\varphi, X^\ell) = \#\{x_i : \varphi(x_i) = 1 \text{ и } y_i = c\} \to \max;$$

$$n_c(\varphi, X^\ell) = \#\{x_i : \varphi(x_i) = 1 \text{ и } y_i \neq c\} \to \min;$$

Если $\varphi(x) = 1$, то говорят «φ выделяет x» (φ covers x).

Требование интерпретируемости

Пример (из области медицины)

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

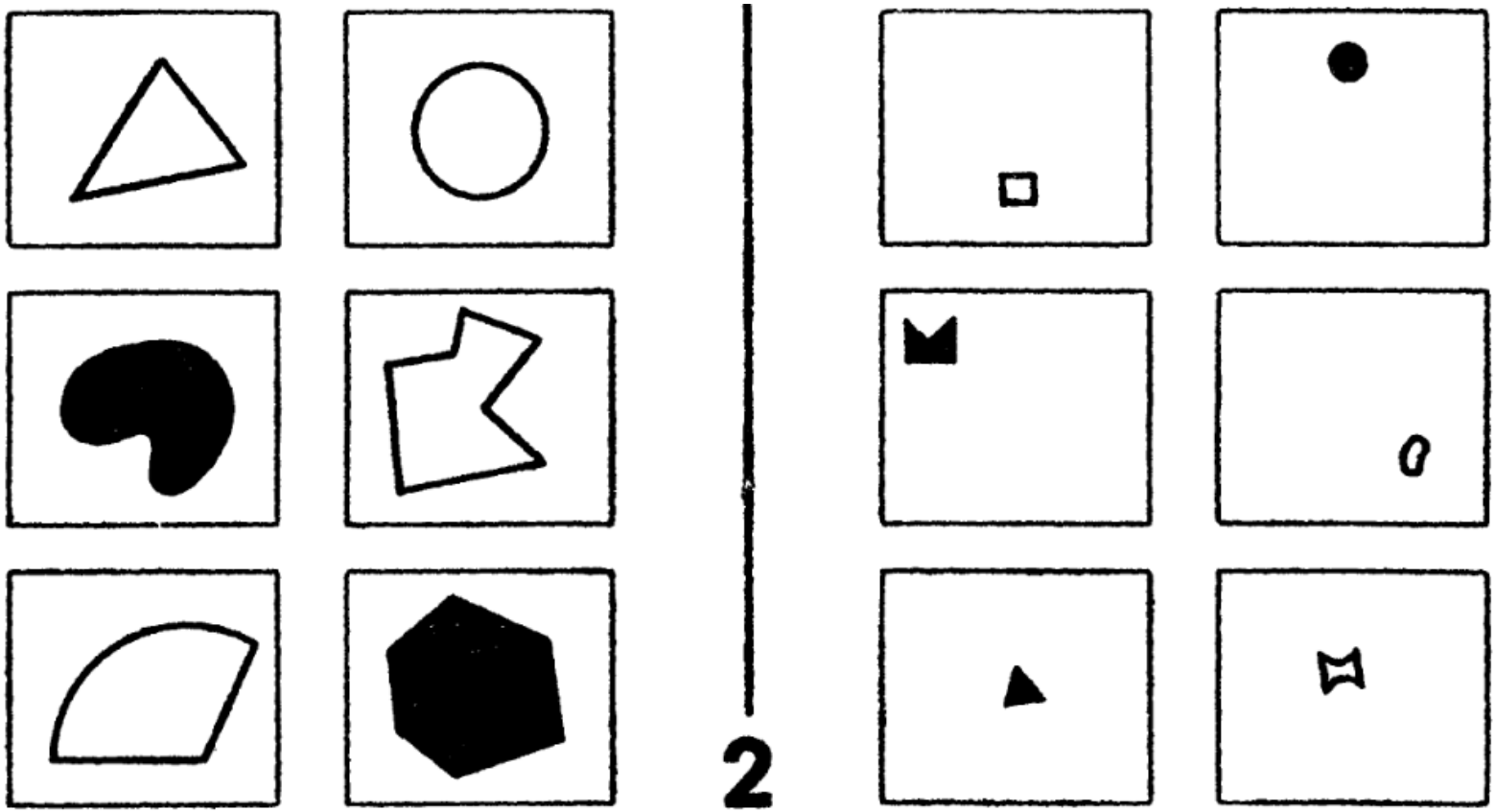
Пример (из области кредитного скоринга)

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

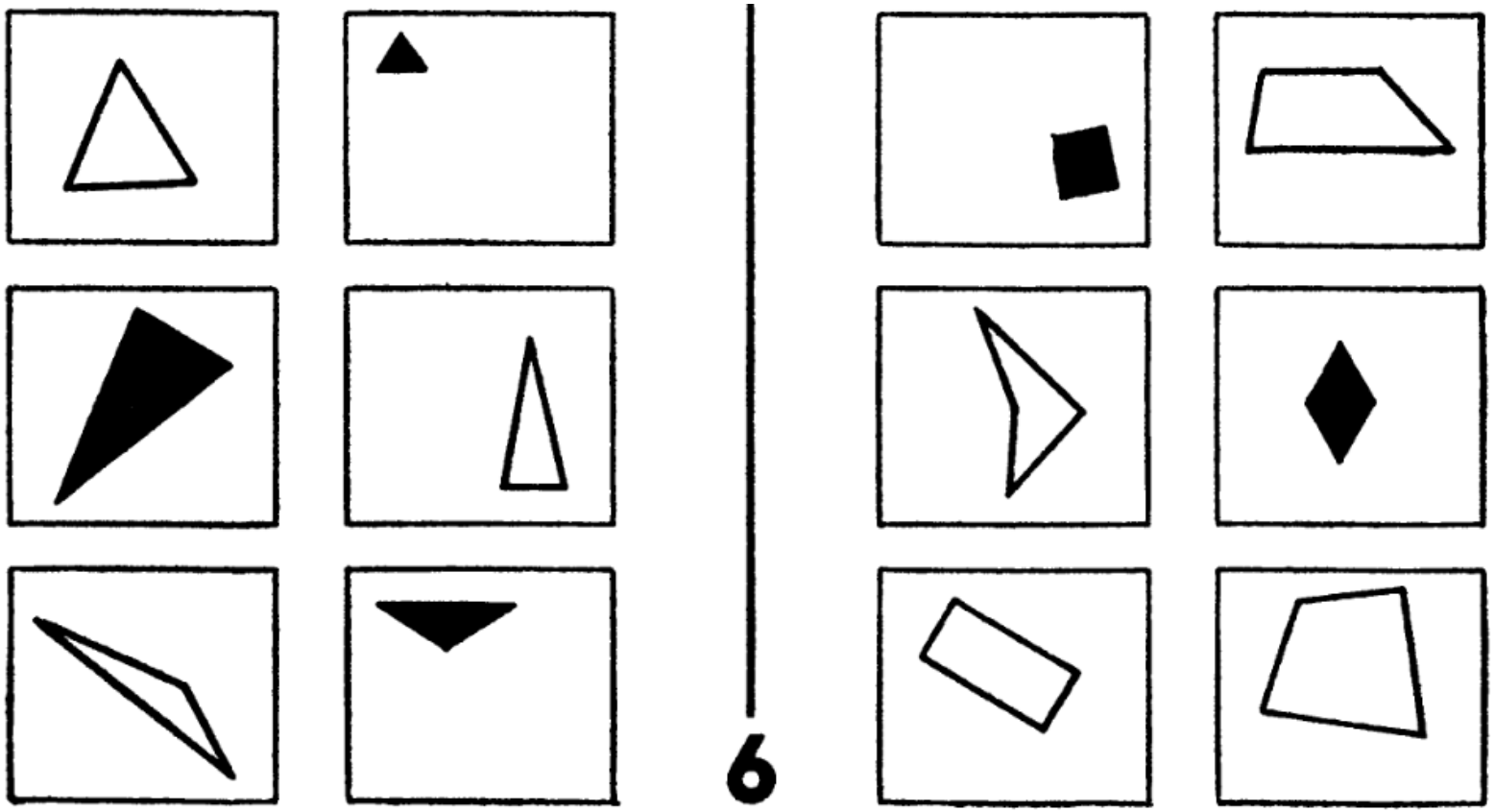
Требования интерпретируемости:

1) φ зависит от малого числа признаков;
2) формула φ выражается на естественном языке.

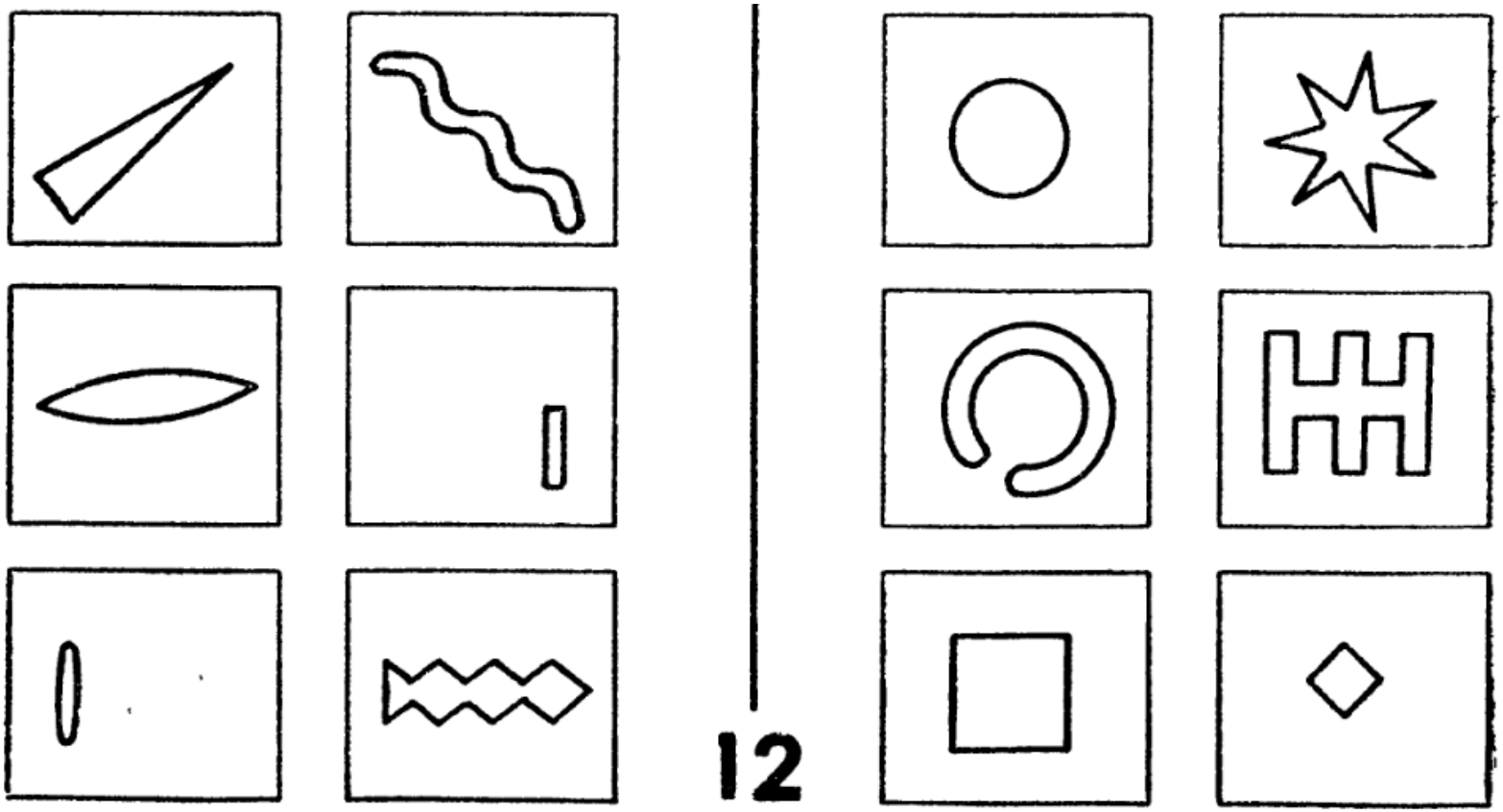
Тесты М. М. Бонгарда [Проблема узнавания, 1967]



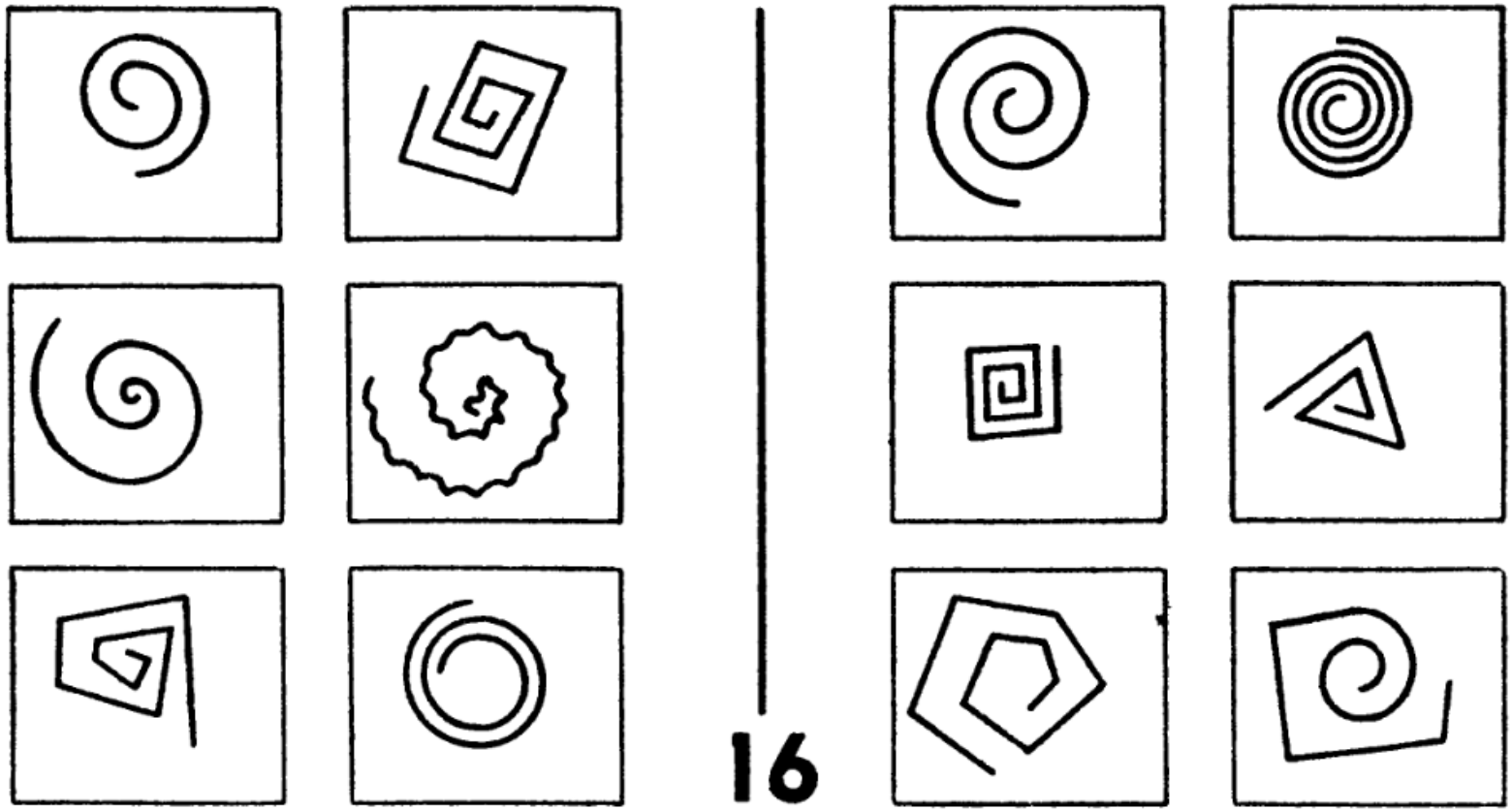
Тесты М. М. Бонгарда [Проблема узнавания, 1967]



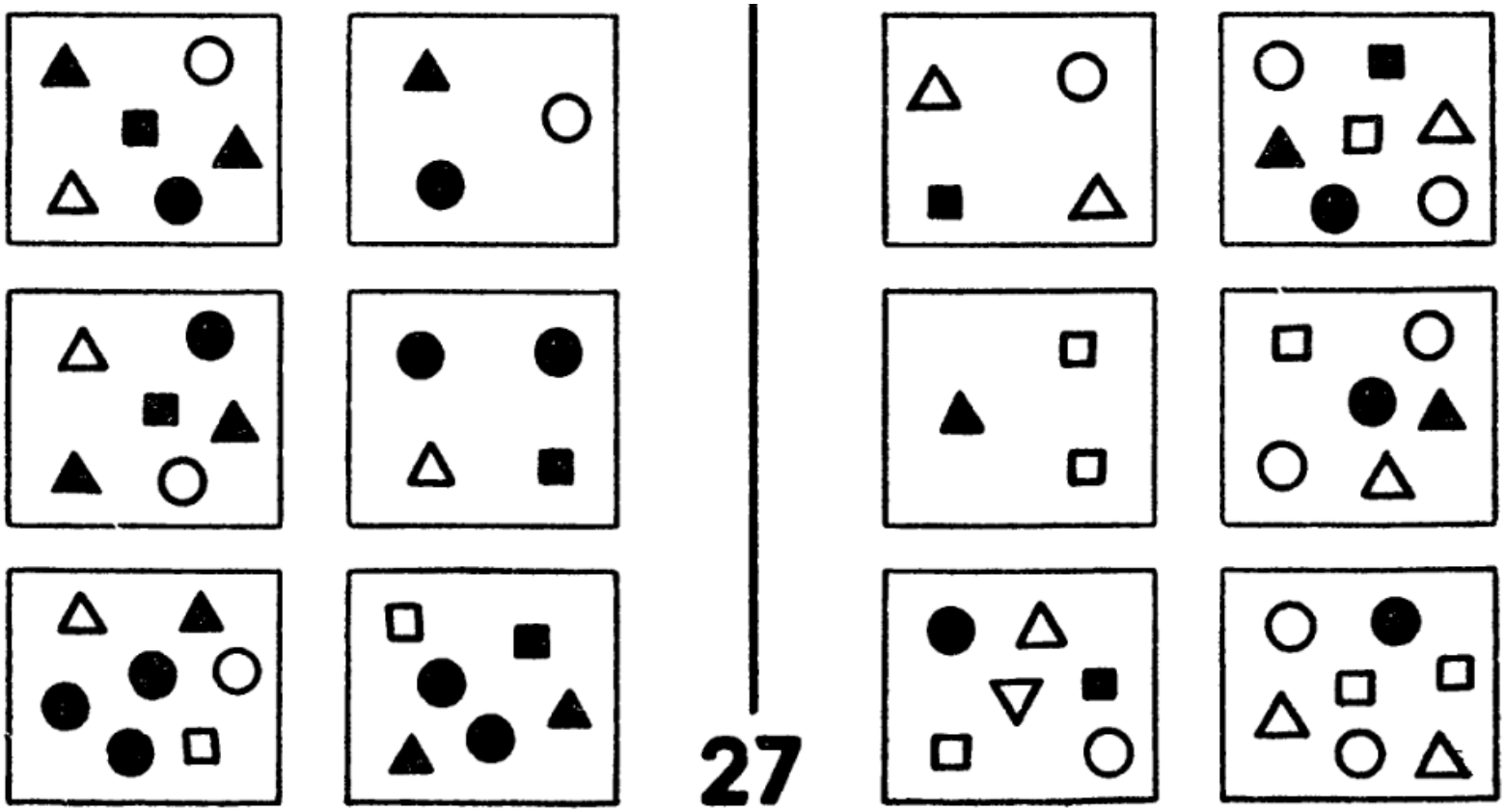
Тесты М. М. Бонгарда [Проблема узнавания, 1967]



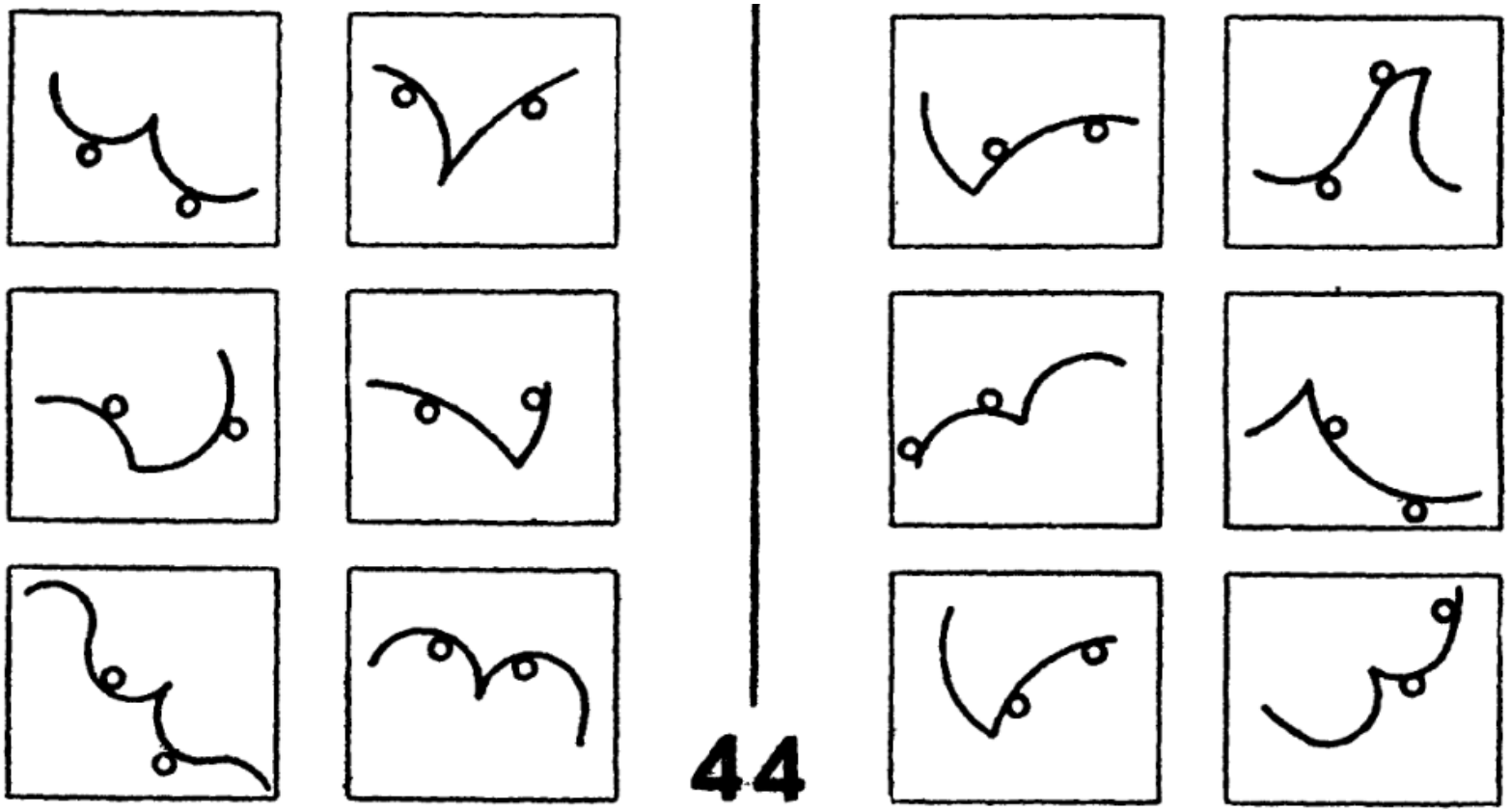
Тесты М. М. Бонгарда [Проблема узнавания, 1967]



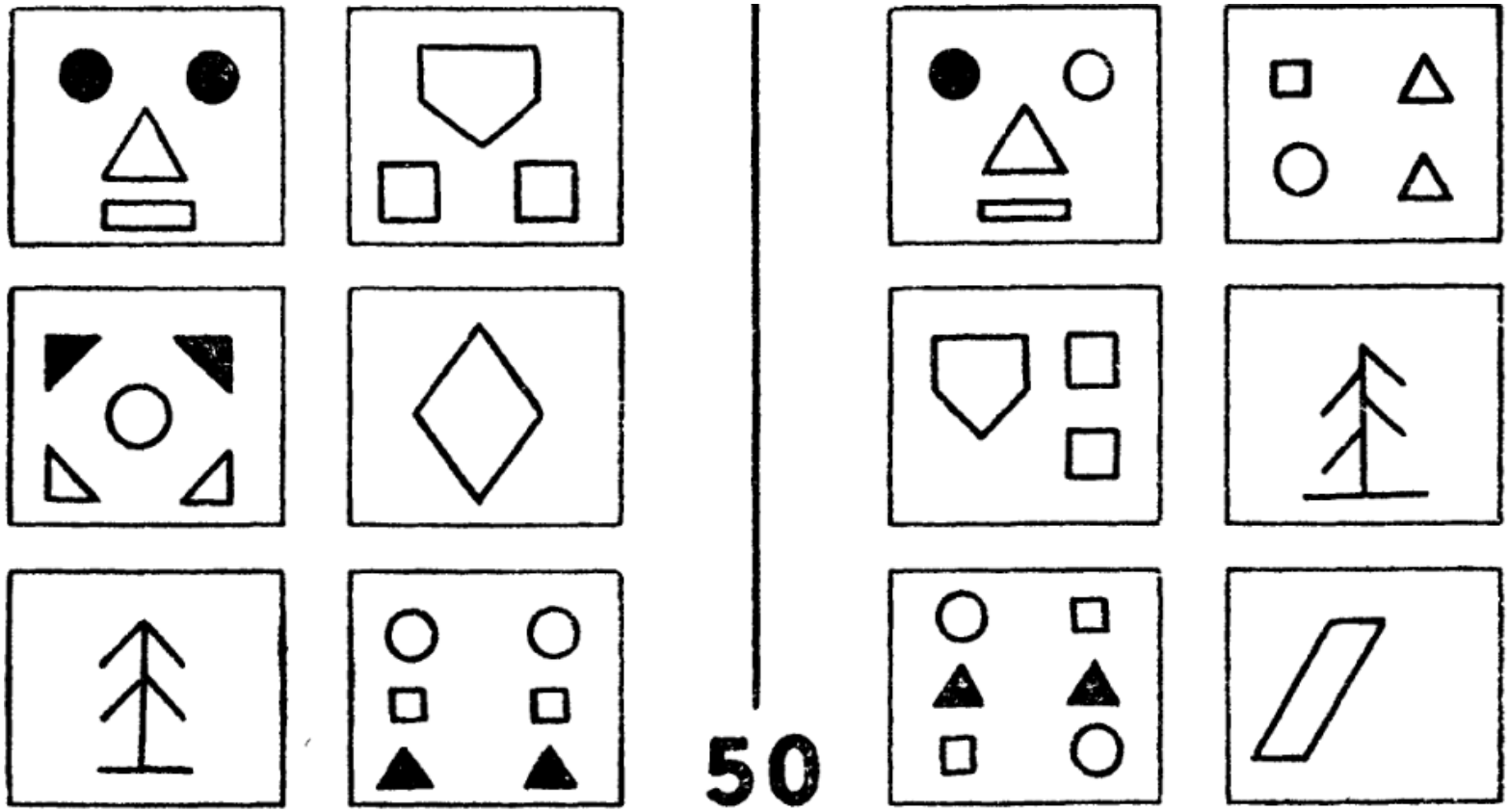
Тесты М. М. Бонгарда [Проблема узнавания, 1967]



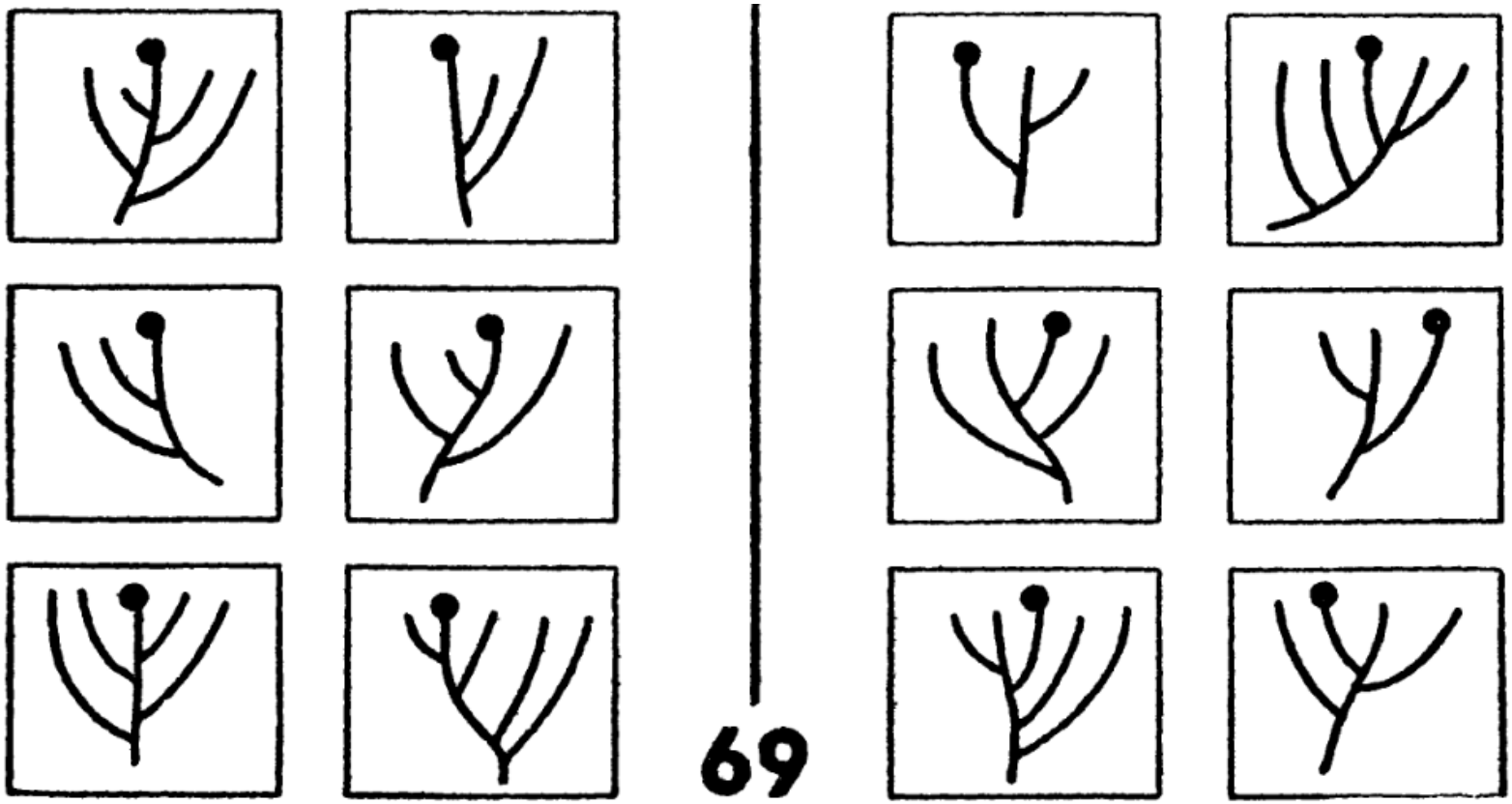
Тесты М. М. Бонгарда [Проблема узнавания, 1967]



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



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



Основные вопросы построения логических алгоритмов

1. Как изобретать признаки?
   — не наука, а искусство (озарения, мозговые штурмы,...)
2. Какого вида закономерности нам нужны?
   — простые формулы от малого числа признаков
3. Как определять информативность?
   — так, чтобы одновременно $p_c \to \max$, $n_c \to \min$
4. Как строить отдельные закономерности?
   — методами отбора признаков
5. Как объединять закономерности в алгоритм?
   — методами построения композиций классификаторов

Закономерность — это хорошо интерпретируемый одноклассовый классификатор с отказами.

Виды интерпретируемых закономерностей

Параметрическое семейство *конъюнкций пороговых условий*:

$$\varphi(x) = \bigwedge_{j \in J} \left[\alpha_j \leqslant f_j(x) \leqslant \beta_j \right].$$

Параметрическое семейство *синдромных правил*:

$$\varphi(x) = \left[\sum_{j \in J} \left[\alpha_j \leqslant f_j(x) \leqslant \beta_j \right] \geqslant K \right].$$

Параметрическое семейство *шаров*:

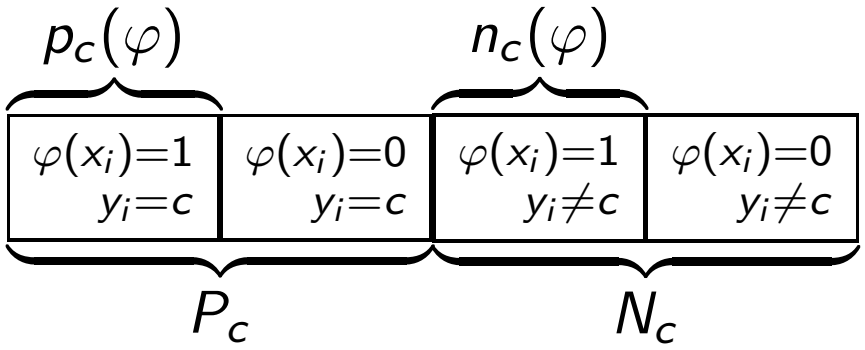
$$\varphi(x) = \left[\sum_{j \in J} \alpha_j |f_j(x) - f_j(x_0)|^\gamma \leqslant R^\gamma \right].$$

Параметрическое семейство *полуплоскостей*:

$$\varphi(x) = \left[\sum_{j \in J} \alpha_j f_j(x) \geqslant \alpha_0 \right].$$

Основная проблема — отбор признаков $J \subseteq \{1, \ldots, n\}$.

Логический (эвристический) критерий закономерности



Определение

Предикат $\varphi(x)$ — логическая ε,δ-закономерность класса $c \in Y$

$$E_c(\varphi, X^\ell) = \frac{n_c(\varphi)}{p_c(\varphi) + n_c(\varphi)} \leqslant \varepsilon;$$

$$D_c(\varphi, X^\ell) = \frac{p_c(\varphi)}{\ell} \geqslant \delta.$$

Проблема: хотелось бы иметь один скалярный критерий.

Нетривиальность проблемы свёртки двух критериев

Пример.

Претенденты на звание «Критерий информативности» при $P = 200$, $N = 100$ и различных p и n.

p	n	$p - n$	$p - 5n$	$\frac{p}{P} - \frac{n}{N}$	$\frac{p}{n+1}$	I_c	IGain_c	$\sqrt{p} - \sqrt{n}$
50	0	**50**	50	0.25	50	22.65	23.70	7.07
100	50	**50**	−150	0	1.96	2.33	1.98	2.93
50	9	41	**5**	0.16	**5**	7.87	7.94	4.07
5	0	5	**5**	0.03	**5**	2.04	3.04	2.24
100	0	**100**	100	**0.5**	100	52.18	53.32	10.0
140	20	**120**	40	**0.5**	6.67	37.09	37.03	7.36

Статистический критерий информативности

Точный тест Фишера. Пусть X — в.п., выборка X^ℓ — i.i.d. Гипотеза H_0: $y(x)$ и $\varphi(x)$ — независимые случайные величины. Тогда вероятность реализации пары (p, n) описывается *гипергеометрическим распределением*:

$$\mathrm{P}(p,n) = \frac{C_P^p C_N^n}{C_{P+N}^{p+n}}, \quad 0 \leqslant p \leqslant P, \quad 0 \leqslant n \leqslant N,$$

где $C_N^n = \frac{N!}{n!(N-n)!}$ — биномиальные коэффициенты.

Определение

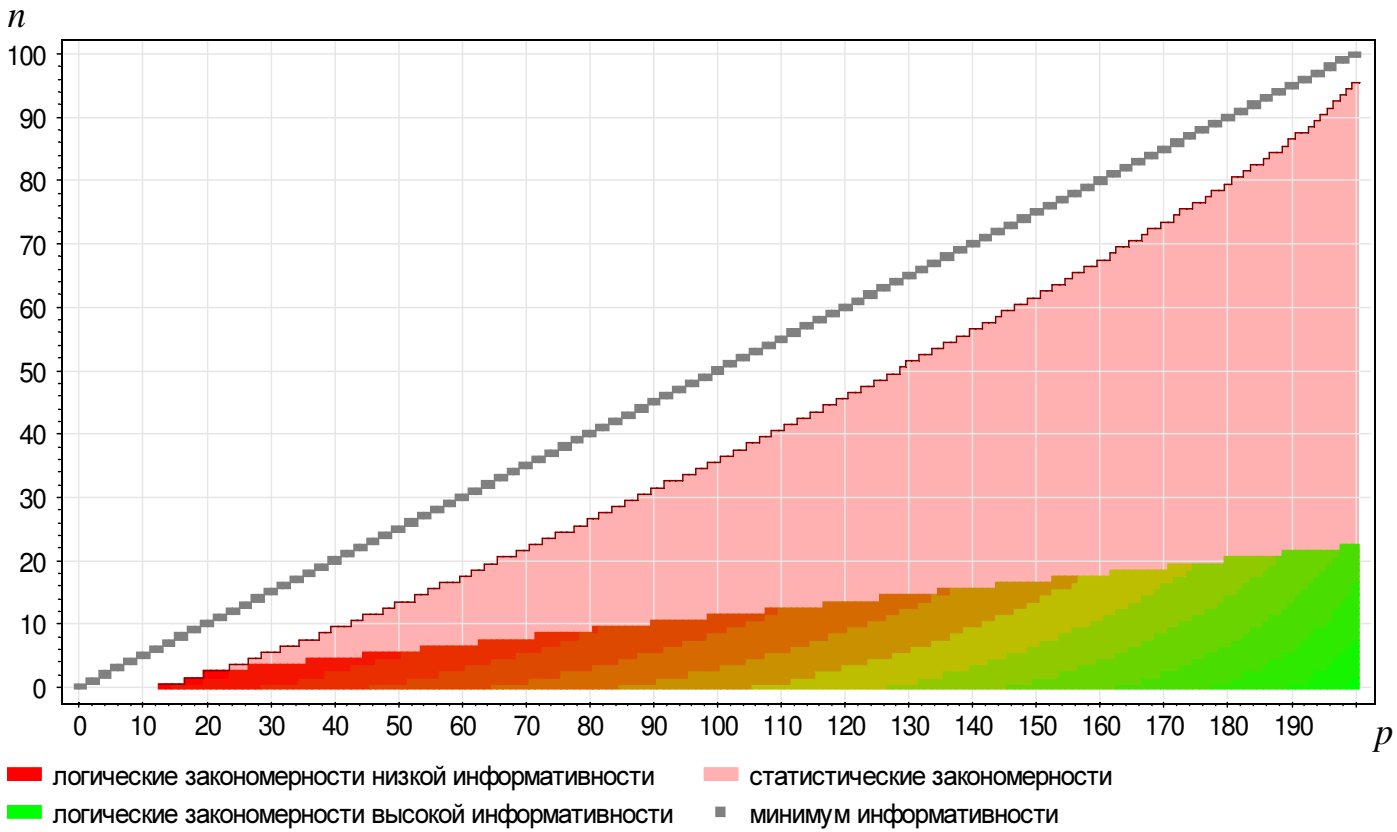
Информативность предиката $\varphi(x)$ относительно класса $c \in Y$:

$$I_c(\varphi, X^\ell) = -\ln \frac{C_{P_c}^{p_c(\varphi)} C_{N_c}^{n_c(\varphi)}}{C_{P_c+N_c}^{p_c(\varphi)+n_c(\varphi)}},$$

$I_c(\varphi, X^\ell) \geqslant I_0$ *— статистическая закономерность класса $c \in Y$.*

Соотношение логического и статистического критериев

Области логических ($\varepsilon = 0.1$) и статистических ($I_0 = 5$) закономерностей в координатах (p, n) при $P = 200$, $N = 100$.



Философский вопрос: закономерность == неслучайность?

Энтропийный критерий информативности

Пусть ω_0, ω_1 — два исхода с вероятностями q и $1-q$.
Количество информации: $I_0 = -\log_2 q$, $I_1 = -\log_2(1-q)$.
Энтропия — математическое ожидание количества информации:

$$h(q) = -q\log_2 q - (1-q)\log_2(1-q).$$

Энтропия выборки X^ℓ, если исходы — это классы $y{=}c$, $y{\neq}c$:

$$H(y) = h\left(\frac{P}{\ell}\right).$$

Энтропия выборки X^ℓ после получения информации φ:

$$H(y|\varphi) = \frac{p+n}{\ell} h\left(\frac{p}{p+n}\right) + \frac{\ell-p-n}{\ell} h\left(\frac{P-p}{\ell-p-n}\right).$$

Информационный выигрыш (Information gain, IGain):

$$\mathrm{IGain}_c(\varphi, X^\ell) = H(y) - H(y|\varphi).$$

Соотношение статистического и энтропийного критериев

Определение

Предикат φ — закономерность по энтропийному критерию, если $\mathrm{IGain}_c(\varphi, X^\ell) > G_0$ при некотором G_0.

Теорема

Энтропийный критерий IGain_c асимптотически эквивалентен статистическому I_c:

$$\mathrm{IGain}_c(\varphi, X^\ell) \to \frac{1}{\ell \ln 2} I_c(\varphi, X^\ell) \quad \text{при } \ell \to \infty.$$

Доказательство:

применить формулу Стирлинга к статистическому критерию.

Задача перебора конъюнкций

Пусть $\mathscr{B}$ — конечное множество *элементарных предикатов*, например, вида $\beta(x) = \big[\alpha_j \leqslant f_j(x) \leqslant \beta_j\big]$.

Множество конъюнкций с ограниченным числом термов из $\mathscr{B}$:

$$\mathscr{K}_K[\mathscr{B}] = \big\{\varphi(x) = \beta_1(x) \wedge \cdots \wedge \beta_k(x) \;\big|\; \beta_1, \ldots, \beta_k \in \mathscr{B},\ k \leqslant K\big\}.$$

Число допустимых конъюнкций: $O(|\mathscr{B}|^K)$ — ооооочень много!

Семейство методов локального поиска

Окрестность $V(\varphi)$ — все конъюнкции, получаемые из φ добавлением, изъятием или модификацией одного из термов.

Основная идея: на t-й итерации

$$\varphi_t := \operatorname*{arg\,max}_{\varphi \in V(\varphi_{t-1})} I_c(\varphi, X^\ell).$$

Обобщённый алгоритм локального поиска

Вход: выборка X^ℓ; класс $c \in Y$;
начальное приближение φ_0; параметры $t_{\max}$, d, ε;
Выход: конъюнкция φ;

1: $I^* := I_c(\varphi_0, X^\ell)$; $\varphi^* := \varphi_0$;
2: **для всех** $t = 1, \ldots, t_{\max}$
3: $\varphi_t := \underset{\varphi \in V(\varphi_{t-1})}{\arg\max} I_c(\varphi, X^\ell)$ — перспективная конъюнкция;
4: $\varphi_t^* := \underset{\substack{\varphi \in V(\varphi_{t-1}) \\ E_c(\varphi) < \varepsilon}}{\arg\max} I_c(\varphi, X^\ell)$ — лучшая конъюнкция;
5: **если** $I_c(\varphi_t^*) > I^*$ **то** $t^* := t$; $\varphi^* := \varphi_t^*$; $I^* := I_c(\varphi^*)$;
6: **если** $t - t^* > d$ **то выход**;
7: **вернуть** φ^*;

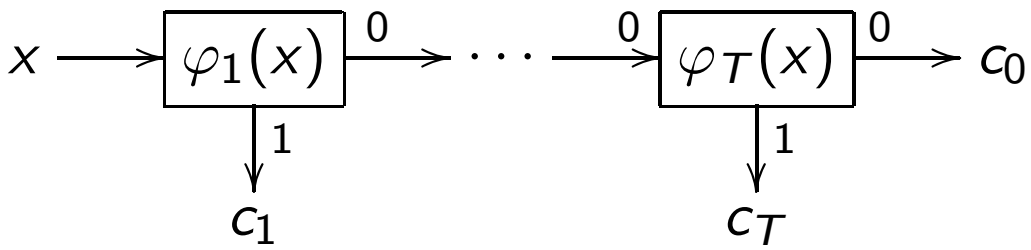
Частные случаи и модификации

- **жадный алгоритм:**
  $V(\varphi)$ — только добавления термов; $\varphi_0 = \varnothing$;
- **стохастический локальный поиск (SLS):**
  $V(\varphi)$ — случайное подмножество всевозможных добавлений, удалений, модификаций термов; $\varphi_0 = \varnothing$;
- **стабилизация:**
  $V(\varphi)$ — удаления термов или изменение параметров в термах; $\varphi_0 \neq \varnothing$;
- **редукция:**
  $V(\varphi)$ — только удаления термов; $\varphi_0 \neq \varnothing$;
  $I_c(\varphi, X^k)$ оценивается по контрольной выборке X^k;
- **поиск в ширину:**
  на каждой итерации строится множество конъюнкций $\Phi_t = \{\varphi_t\}$.

Определение решающего списка

Решающий список (decision list, DL)
— алгоритм классификации $a\colon X \to Y$, который задаётся закономерностями $\varphi_1(x), \ldots, \varphi_T(x)$ классов $c_1, \ldots, c_T \in Y$:

1: **для всех** $t = 1, \ldots, T$
2: **если** $\varphi_t(x) = 1$ **то**
3: **вернуть** c_t;
4: **вернуть** c_0.



«Особый ответ» c_0 — отказ от классификации объекта x.

Построение решающего списка

Вход: выборка X^ℓ; семейство предикатов Φ;
параметры: $T_{\max}$, $I_{\min}$, $E_{\max}$, ℓ_0;
Выход: решающий список $\{\varphi_t, c_t\}_{t=1}^{T}$;

1: $U := X^\ell$;
2: **для всех** $t := 1, \ldots, T_{\max}$
3: $c := c_t$ — выбрать класс из Y;
4: $\Phi' = \{\varphi \in \Phi : E_c(\varphi, U) \leqslant E_{\max}\}$;
5: $\varphi_t := \arg\max\limits_{\varphi \in \Phi'} I_c(\varphi, U)$;
6: **если** $I_c(\varphi_t, U) < I_{\min}$ **то выход**;
7: исключить из выборки объекты, выделенные правилом φ_t:
$U := \{x \in U : \varphi_t(x) = 0\}$;
8: **если** $|U| \leqslant \ell_0$ **то выход**;

Замечания к алгоритму построения решающего списка

- Параметр $E_{\max}$ позволяет управлять сложностью списка:
  $E_{\max} \downarrow \quad \Rightarrow \quad p(\varphi_t) \downarrow,\ T \uparrow.$
- Стратегии выбора класса c_t:
  1) все классы по очереди;
  2) на каждом шаге определяется оптимальный класс:
  $$(\varphi_t, c_t) := \arg\max_{\varphi \in \Phi', c \in Y} I_c(\varphi, U);$$
- Простой обход проблемы пропусков в данных.
- Другие названия:
  — комитет с логикой старшинства;
  — голосование по старшинству;
  — машина покрывающих множеств (SCM);

Решающие списки: достоинства и недостатки

Достоинства:

- Интерпретируемость и простота классификации.
- Универсальность: можно использовать любое семейство Ф.
- Допустимы разнотипные данные и данные с пропусками.
- Правила получаются различными. Можно построить несколько списков и по ним проголосовать.

Недостатки:

- При неудачном выборе Ф список может не построиться, будет много отказов от классификации.
- Список плохо интерпретируется, если он длинный и/или правила различных классов следуют вперемежку.
- Качество классификации обычно ниже, чем у голосования, когда правила могут компенсировать ошибки друг друга.

Резюме в конце лекции

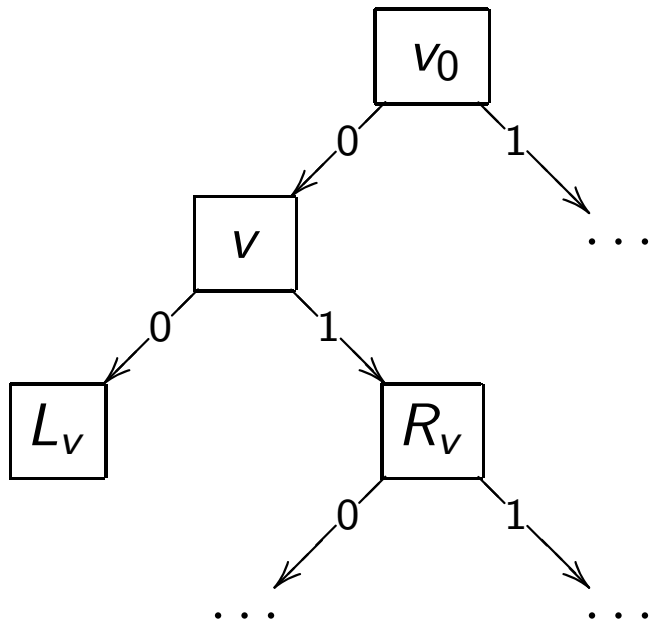
- *Правило* — это интерпретируемый предикат $X \to \{0, 1\}$.
- *Закономерность* — это информативное правило.
- Существует много критериев *информативности*:
  - *статистический* критерий — для поиска закономерностей,
  - *энтропийный* критерий — его асимптотический вариант,
  - *логический* ε, δ-критерий — для отбора закономерностей.
- Как строить отдельные закономерности:
  - *отбор признаков* по критерию информативности,
  - *локальный поиск* — простой жадный алгоритм,
  - *бинаризация* — предварительный этап сокращения поиска.
- Как строить композицию закономерностей:
  - решающий список,
  - решающее дерево,
  - взвешенное голосование.

Определение бинарного решающего дерева

Бинарное решающее дерево — алгоритм классификации $a(x)$, задающийся бинарным деревом:

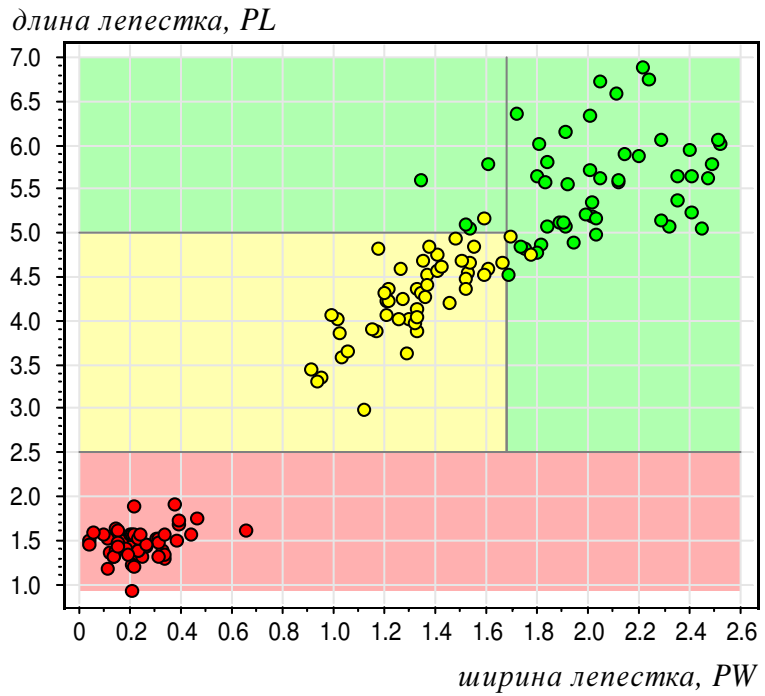
1) $\forall v \in V_{\text{внутр}} \;\rightarrow\;$ предикат $\beta_v : X \to \{0,1\}$, $\beta \in \mathscr{B}$
2) $\forall v \in V_{\text{лист}} \;\rightarrow\;$ имя класса $c_v \in Y$.

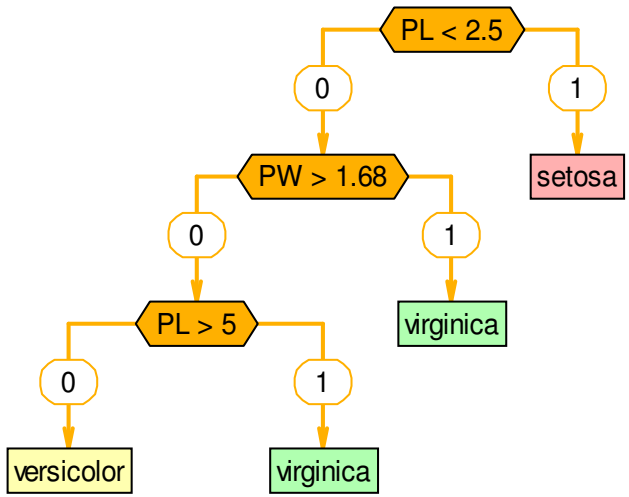
1: $v := v_0$;
2: **пока** $v \in V_{\text{внутр}}$
3: **если** $\beta_v(x) = 1$ **то**
4: переход вправо:
$v := R_v$;
5: **иначе**
6: переход влево:
$v := L_v$;
7: **вернуть** c_v.



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.

Жадный алгоритм построения дерева ID3

ПРОЦЕДУРА LearnID3 $(U \subseteq X^\ell)$;
если все объекты из U лежат в одном классе $c \in Y$ **то**
    вернуть новый лист v, $c_v := c$;
найти предикат с максимальной информативностью:
$\beta := \arg\max_{\beta \in \mathscr{B}} I(\beta, U)$;
разбить выборку на две части $U = U_0 \cup U_1$ по предикату β:
$U_0 := \{x \in U : \beta(x) = 0\}$;
$U_1 := \{x \in U : \beta(x) = 1\}$;
если $U_0 = \varnothing$ или $U_1 = \varnothing$ **то**
    вернуть новый лист v, $c_v :=$ Мажоритарный класс(U);
создать новую внутреннюю вершину v: $\beta_v := \beta$;
построить левое поддерево: $L_v :=$ LearnID3 (U_0);
построить правое поддерево: $R_v :=$ LearnID3 (U_1);
вернуть v;

Разновидности критериев ветвления

1. **Отделение одного класса** (слишком сильное ограничение):

$$I(\beta, X^\ell) = \max_{c \in Y} I_c(\beta, X^\ell).$$

2. **Многоклассовый энтропийный критерий:**

$$I(\beta, X^\ell) = \sum_{c \in Y} h\left(\frac{P_c}{\ell}\right) - \frac{p}{\ell} \sum_{c \in Y} h\left(\frac{p_c}{p}\right) - \frac{\ell - p}{\ell} \sum_{c \in Y} h\left(\frac{P_c - p_c}{\ell - p}\right),$$

где $P_c = \#\{x_i : y_i = c\}$, $p = \#\{x_i : \beta(x_i) = 1\}$, $h(z) \equiv -z \log_2 z$.

3. **Критерий Джини:**

$$I(\beta, X^\ell) = \#\big\{(x_i, x_j) : \beta(x_i) = \beta(x_j) \text{ и } y_i = y_j\big\}.$$

4. ***D*-критерий В.И.Донского:**

$$I(\beta, X^\ell) = \#\big\{(x_i, x_j) : \beta(x_i) \neq \beta(x_j) \text{ и } y_i \neq y_j\big\}.$$

Обработка пропусков

На стадии обучения:

- Если $\beta(x_i)$ не определено, то при вычислении $I(\beta, U)$ объект x_i исключается из выборки U.
- $q_v = \frac{|U_0|}{|U|}$ — оценка вероятности левой ветви, $\forall v \in V_{\text{внутр}}$.

На стадии классификации:

- $\hat{P}_v(y|x) = \beta_v(x)\hat{P}_{L_v}(y|x) + \big(1-\beta_v(x)\big)\hat{P}_{R_v}(y|x)$;
- $\beta_v(x)$ не определено $\Rightarrow$ *пропорциональное распределение*:

$$\hat{P}_v(y|x) = \begin{cases} [y = c_v], & v \in V_{\text{лист}}; \\ q_v\hat{P}_{L_v}(y|x) + \big(1-q_v\big)\hat{P}_{R_v}(y|x), & v \in V_{\text{внутр}}. \end{cases}$$

- Окончательное решение — байесовское правило:
$y = \arg\max\limits_{y \in Y} \hat{P}_{v_0}(y|x)$.

Решающие деревья ID3: достоинства и недостатки

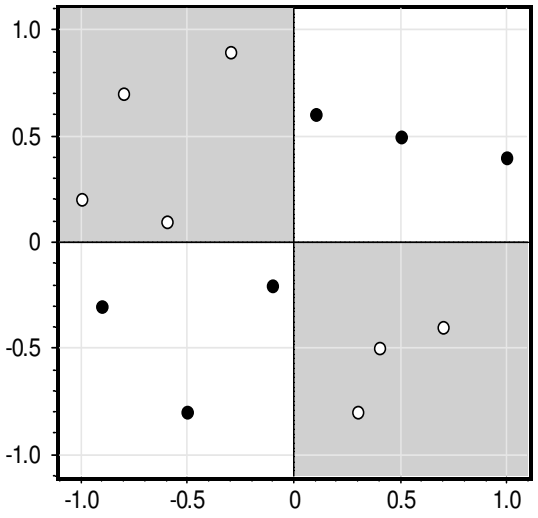
Достоинства:

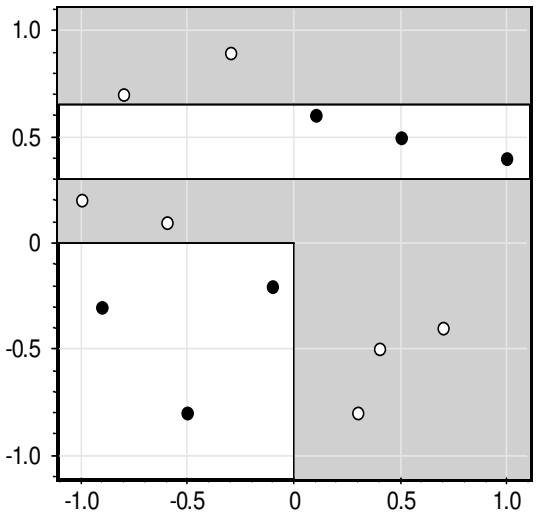
- Интерпретируемость и простота классификации.
- Гибкость: можно варьировать множество $\mathscr{B}$.
- Допустимы разнотипные данные и данные с пропусками.
- Трудоёмкость линейна по длине выборки $O(|\mathscr{B}|h\ell)$.
- Не бывает отказов от классификации.

Недостатки:

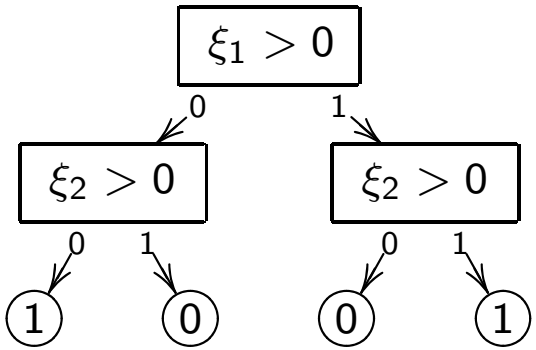
- Жадный ID3 переусложняет структуру дерева, и, как следствие, сильно переобучается.
- Фрагментация выборки: чем дальше v от корня, тем меньше статистическая надёжность выбора β_v, c_v.
- Высокая чувствительность к шуму, к составу выборки, к критерию информативности.

Жадный ID3 переусложняет структуру дерева

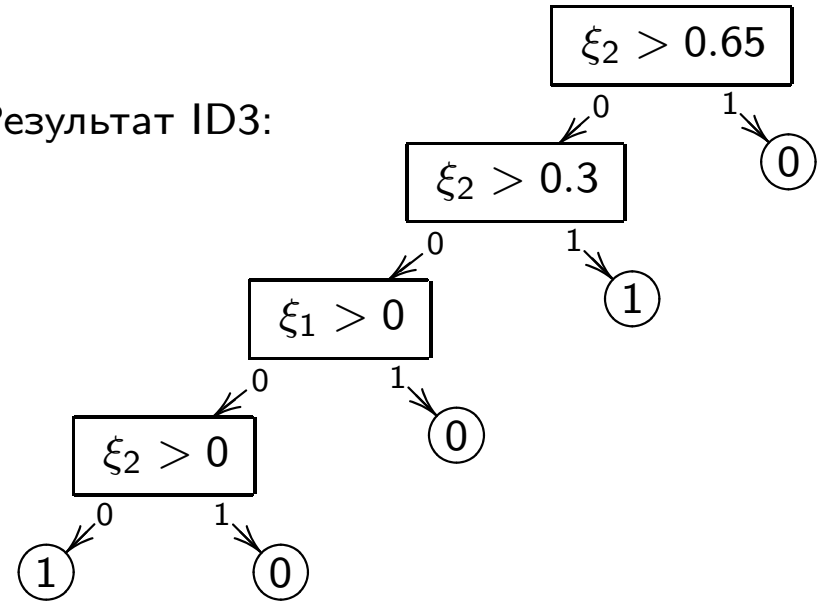




Оптимальное дерево для задачи XOR:



Результат ID3:



Стратегия пред-просмотра (look ahead)

Шаг 4:
найти предикат с максимальной информативностью:
$\beta := \arg\max\limits_{\beta \in \mathscr{B}} I(\beta, U)$;

Шаг 4 заменяется на более ресурсоёмкую процедуру:

для всех деревьев T глубины h
$r_T(U) =$ число ошибок дерева T на выборке U;
$\beta :=$ корень лучшего поддерева $\arg\min\limits_T r_T(U)$;

Достоинства:

- Задача XOR решается практически идеально.

Недостатки:

- При $h > 2$ ooooooочень долго.
- На реальных данных улучшение незначительно.

Стратегия пред-редукции (pre-pruning)

Шаг 6:

если $U_0 = \varnothing$ или $U_1 = \varnothing$ **то**

    вернуть новый лист v;

Шаг 6 заменяется на более мягкое условие:

если $I(\beta, U) \leqslant I_0$ **то**

    вернуть новый лист v;

Достоинства:

- Сразу строится более простое дерево.

Недостатки:

- Качество дерева может и не улучшиться.

Стратегия пост-редукции (post-pruning: C4.5, CART)

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

1: **для всех** $v \in V_{\text{внутр}}$
2: $S_v :=$ подмножество объектов X^k, дошедших до v;
3: **если** $S_v = \varnothing$ **то**
4: **вернуть** новый лист v, $c_v :=$ Мажоритарный класс(U);
5: число ошибок при классификации S_v четырьмя способами:
$r(v)$ — поддеревом, растущим из вершины v;
$r_L(v)$ — поддеревом левой дочерней вершины L_v;
$r_R(v)$ — поддеревом правой дочерней вершины R_v;
$r_c(v)$ — к классу $c \in Y$.
6: в зависимости от того, какое из них минимально:
сохранить поддерево v;
заменить поддерево v поддеревом L_v;
заменить поддерево v поддеревом R_v;
заменить поддерево v листом, $c_v := \arg\min\limits_{c \in Y} r_c(v)$.

Преобразование решающего дерева в список (C4.5-rules)

- Для любого бинарного решающего дерева

$$a(x) = \arg\max_{y \in Y} \sum_{v \in V_{\text{лист}}} [c_v = y] K_v(x),$$

где $K_v(x)$ — конъюнкция по всем рёбрам пути $[v_0, v]$:

$$K_v(x) = \bigwedge_{(u,R_u)} \beta_u(x) \bigwedge_{(u,L_u)} \bar{\beta}_u(x).$$

- Редукция $K_v(x)$, $\forall v \in V_{\text{лист}}$ по контрольной выборке X^k.

Достоинства:

- Переобучение, как правило, уменьшается.

Недостатки:

- Преобразование в список необратимо: это уже не дерево.

Небрежные решающие деревья — ODT (Oblivious Decision Tree) [1993]

Решение проблемы фрагментации:

строится сбалансированное дерево высоты H;

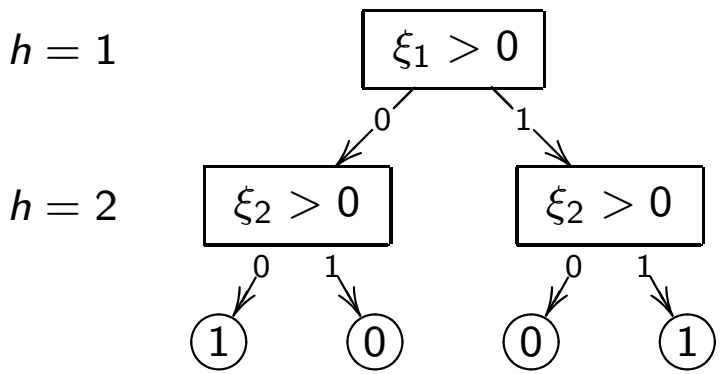
для всех узлов уровня h условие ветвления $\beta_h(x)$ *одинаково*;

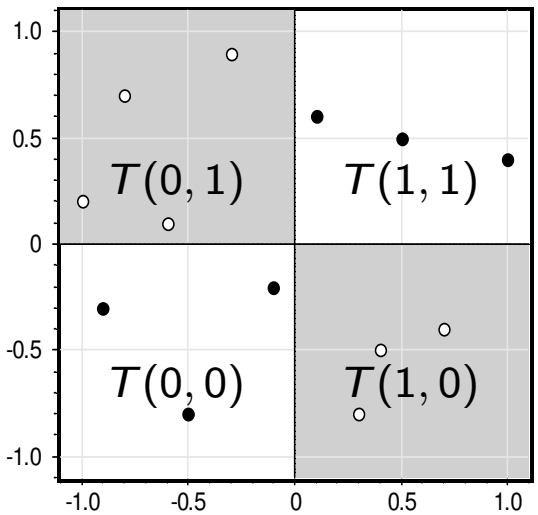
на уровне h ровно 2^{h-1} вершин; X делится на 2^H ячеек.

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

$$a(x) = T\big(\beta_1(x), \ldots, \beta_H(x)\big).$$

Пример: задача XOR, $H = 2$.





Алгоритм обучения ODT

Вход: выборка X^ℓ; семейство правил $\mathscr{B}$; глубина дерева H;
Выход: условия β_h, $h = 1, \ldots, H$; таблица $T: \{0,1\}^H \to Y$;

1: **для всех** $h = 1, \ldots, H$
2: найти предикат с максимальной информативностью:
$\beta_h := \arg\max\limits_{\beta \in \mathscr{B}} I(\beta_1, \ldots, \beta_{h-1}, \beta; X^\ell)$;
3: **для всех** $b \equiv (b_1, \ldots, b_H) \in \{0,1\}^H$
4: классификация по мажоритарному правилу:
$$T(b_1, \ldots, b_H) := \arg\max_{c \in Y} \sum_{i=1}^{\ell} [y_i = c] \prod_{h=1}^{H} \left[\beta_h(x_i) = b_h\right];$$

$$I(\beta_1, \ldots, \beta_h) = \sum_{c \in Y} h\left(\frac{P_c}{\ell}\right) - \sum_{b \in \{0,1\}^h} \frac{|X_b|}{\ell} \sum_{c \in Y} h\left(\frac{|X_b \cap X_c|}{|X_b|}\right);$$

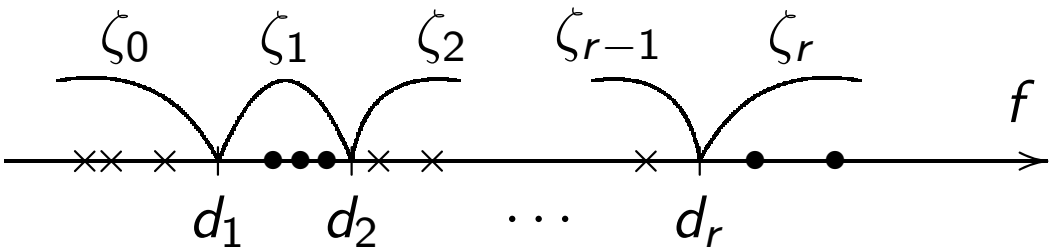
$$X_b = \left\{x_i : \beta_s(x_i) = b_s,\ s = 1, \ldots, h\right\}, \quad X^\ell = \bigsqcup_{b \in \{0,1\}^h} X_b.$$

Вспомогательная задача бинаризации вещественного признака

Цель: сократить перебор предикатов вида $\left[\alpha \leqslant f(x) \leqslant \beta\right]$.

Дано: выборка значений вещественного признака $f(x_i)$, $x_i \in X^\ell$.
Найти: наилучшее (в каком-то смысле) разбиение области значений признака на относительно небольшое число зон:

$$\zeta_0(x) = [f(x) < d_1];$$
$$\zeta_s(x) = [d_s \leqslant f(x) < d_{s+1}], \qquad s = 1, \ldots, r-1;$$
$$\zeta_r(x) = [d_r \leqslant f(x)].$$



Способы разбиения области значений признака на зоны

1. Жадная максимизация информативности путём слияний
2. Разбиение на равномощные подвыборки
3. Разбиение по равномерной сетке
4. Объединение нескольких разбиений

Повышение интерпретируемости пороговых значений

Задача: на отрезке $[a, b]$ найти значение x с минимальным числом значащих цифр.
Если таких x несколько, выбрать $\arg\min_x \left|\frac{1}{2}(a+b) - x\right|$.

Алгоритм разбиения области значений признака на зоны

Вход: выборка X^ℓ; класс $c \in Y$; параметры r и δ_0.
Выход: $D = \{d_1 < \dots < d_r\}$ — последовательность порогов;

$D := \varnothing$; упорядочить выборку X^ℓ по возрастанию $f(x_i)$;
для всех $i = 2, \dots, \ell$
  если $f(x_{i-1}) \neq f(x_i)$ и $[y_{i-1} = c] \neq [y_i = c]$ **то**
    добавить порог $\frac{1}{2}(f(x_{i-1}) + f(x_i))$ в конец D;
повторять
  для всех $d_i \in D$, $i = 1, \dots, |D| - 1$
    $\delta I_i := I_c(\zeta_{i-1} \vee \zeta_i \vee \zeta_{i+1}) - \max\{I_c(\zeta_{i-1}), I_c(\zeta_i), I_c(\zeta_{i+1})\}$;
  $i := \arg\max_s \delta I_s$;
  если $\delta I_i > \delta_0$ **то**
    слить зоны ζ_{i-1}, ζ_i, ζ_{i+1}, удалив d_i и d_{i+1} из D;
пока $|D| > r + 1$.

Резюме в конце лекции

- 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,
  - специальные виды деревьев ODT, ADT и др.
  - композиции (леса) деревьев — см. далее;

Yandex MatrixNet = градиентный бустинг над ODT.