Методы машинного обучения. Логические методы классификации

Воронцов Константин Вячеславович
www.MachineLearning.ru/wiki?title=User:Vokov
вопросы к лектору: k.vorontsov@iai.msu.ru

материалы курса:
github.com/MSU-ML-COURSE/ML-COURSE-25-26
орг.вопросы по курсу: ml.cmc@mail.ru

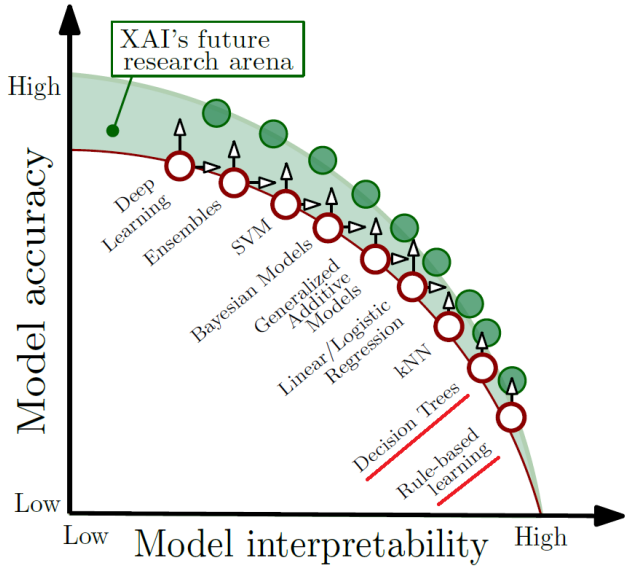
ВМК МГУ • 28 октября 2025

Содержание

1. **Решающие деревья**
   - Обучение решающих деревьев
   - Критерии ветвления
   - Усечение дерева (pruning)
2. **Индукция правил**
   - Понятие закономерности
   - Алгоритмы поиска закономерностей
   - Критерии информативности
3. **Решающие списки и таблицы**
   - Решающие списки
   - Бинаризация признаков
   - Решающие таблицы

Объяснимый ИИ (XAI, eXplainable Artificial Intelligence)

- **Interpretability**
  — пассивная интерпретируемость внутреннего строения модели или предсказания на объекте
- **Understandability**, **Transparency**
  — понятность, самоочевидность, прозрачность строения модели



- **Explainability** — активная генерация объяснений на объекте как дополнительных выходных данных модели
- **Comprehensibility** — возможность представить выученные закономерности в виде понятного людям знания

"Do you want an interpretable model, or the one that works?"

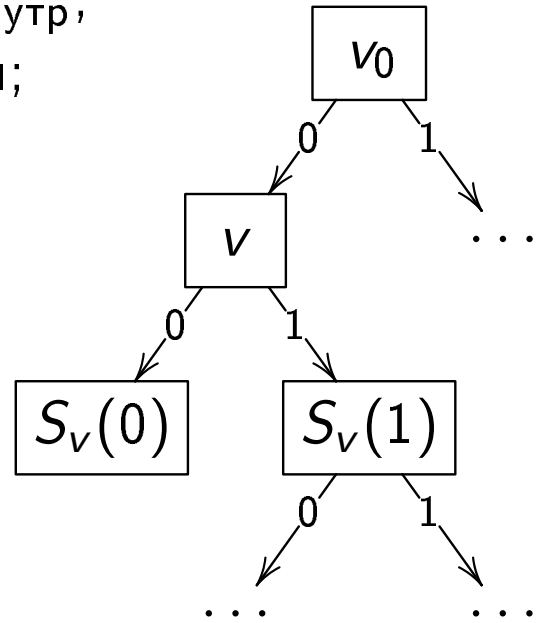
[Yann LeCun, NIPS'17]

V.Belle, I.Papantonis. Principles and practice of explainable machine learning. 2020

Определение решающего дерева (Decision Tree)

Решающее дерево — модель классификации/регрессии $a(x)$, задающаяся *деревом* (связным ациклическим графом) с корнем $v_0 \in V$ и множеством вершин $V = V_{\text{внутр}} \sqcup V_{\text{лист}}$;

$f_v : X \to D_v$ — дискретный признак, $v \in V_{\text{внутр}}$;
$S_v : D_v \to V$ — множество дочерних вершин;
$y_v \in Y$ — ответ в вершине $v \in V_{\text{лист}}$;



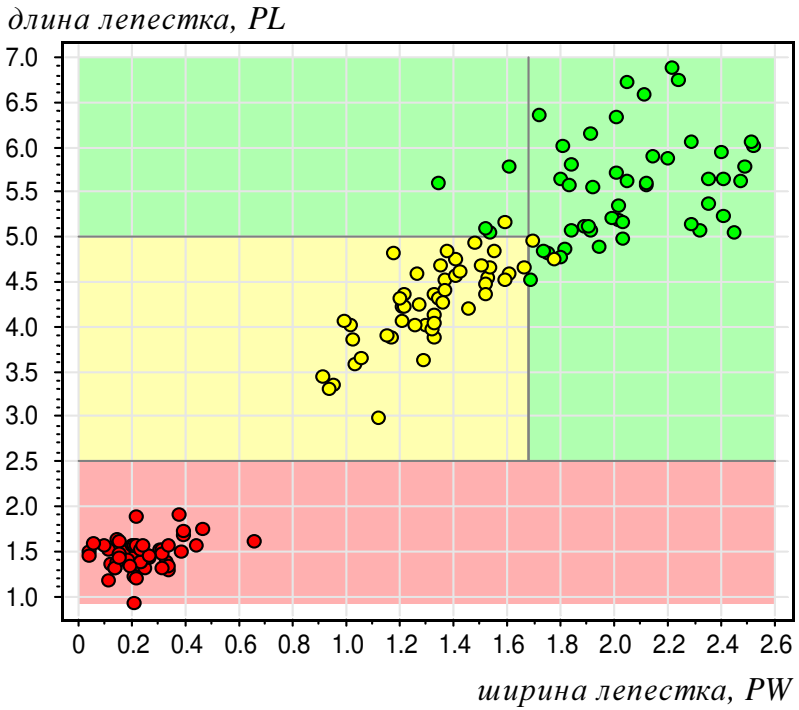
$v := v_0$;
пока $(v \in V_{\text{внутр}})$: $v := S_v(f_v(x))$;
вернуть $a(x) := y_v$;

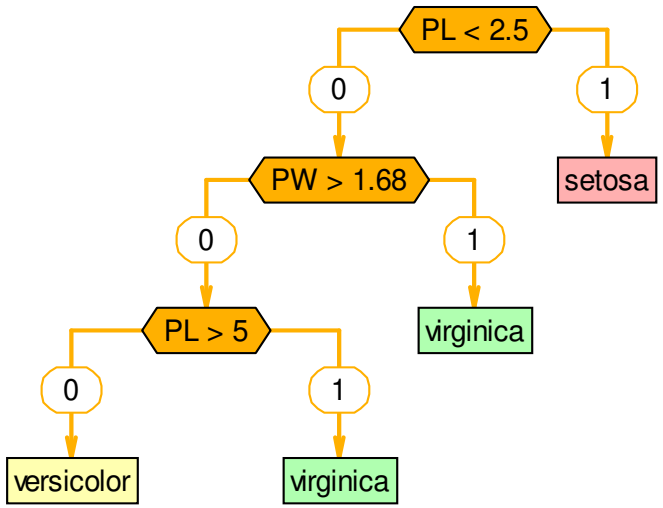
На практике часто используются бинарные признаки вида $f_v(x) = \left[f_j(x) > \theta\right]$, $j = 1, \ldots, n$

Если $D_v \equiv \{0, 1\}$, то решающее дерево называется *бинарным*

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.

Обучение решающего дерева: ID3 (Iterative Dichotomiser)

Обучающая выборка $X^\ell = (x_i, y_i)_{i=1}^\ell$, признаки $F = \{f_1, \ldots, f_n\}$
$v_0 := \text{TreeGrowing}\,(X^\ell)$ — функция рекурсивно вызывает себя

TreeGrowing (**Вход:** $U \subseteq X^\ell$) $\mapsto$ **Выход:** корень дерева v;
$f_v := \arg\max\limits_{f \in F} \text{Gain}\,(f, U)$ — критерий ветвления дерева;
если $\text{Gain}\,(f_v, U) < G_0$ **то**
  создать новый лист v; $y_v := \text{Major}\,(U)$; **вернуть** v;
создать новую внутреннюю вершину v с функцией f_v;
для всех $k \in D_v$
  $U_k := \{x \in U : f_v(x) = k\}$;
  $S_v(k) := \text{TreeGrowing}\,(U_k)$;
вернуть v;

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

John Ross Quinlan. Induction of Decision Trees // Machine Learning, 1986.

Обработка пропущенных значений (missing values)

На стадии обучения оцениваются вероятности:

- $P(k|v) := \frac{|U_k|}{|U|}$ — перехода в $S_v(k)$ из $v \in V_{\text{внутр}}$
- $P(y|x,v) := \frac{1}{|U|} \sum_{x_i \in U} [y_i = y]$ — класса y в листе $v \in V_{\text{лист}}$
- $f(x_i)$ не определено $\Rightarrow x_i$ исключается из U для $\text{Gain}(f, U)$

На стадии классификации:

- $a(x) = \arg\max_{y \in Y} P(y|x, v_0)$ — наиболее вероятный класс

если значение $f_v(x)$ не определено **то**
  по формуле полной вероятности:
  $P(y|x,v) := \sum_{k \in D_v} P\big(y \,\big|\, x, S_v(k)\big)\, P(k|v)$;
иначе
  $P(y|x,v) := P\big(y \,\big|\, x, S_v(f_v(x))\big)$;

Критерий ветвления Gain (f, U)

$\mathscr{L}(a, y)$ — потеря от ответа модели a при верном ответе y.
Суммарная потеря модели-константы a на выборке $U \subseteq X^\ell$:

$$Q(a, U) = \sum_{x_i \in U} \mathscr{L}(a, y_i)$$

Пусть U — множество объектов x_i, дошедших до вершины v.
Суммарная потеря, если вершину v сделать листом:

$$Q_0(U) = \min_{a \in Y} Q(a, U)$$

Суммарная потеря при ветвлении $U_k = \{x \in U : f(x) = k\}$:

$$Q(f, U) = \sum_{k \in D} \min_{a \in Y} Q\left(a, U_k\right)$$

$\text{Gain}\,(f, U) = Q_0(U) - Q(f, U)$ — выигрыш от ветвления по f

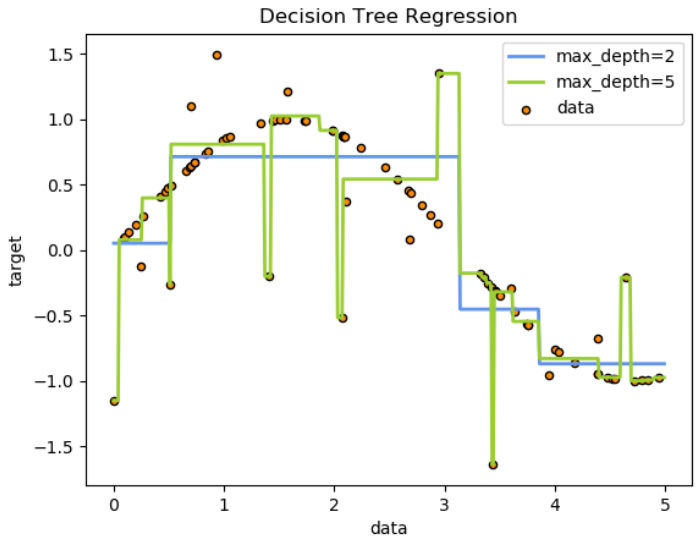
CART: деревья регрессии и классификации

При квадратичной функции потерь $\mathscr{L}(a, y) = (a - y)^2$, $Y = \mathbb{R}$, задача наименьших квадратов решается аналитически для значения y_v в листовой вершине $v \in V_{\text{лист}}$:

$$y_v = \arg\min_{a \in Y} \sum_{x_i \in U} (a - y_i)^2 = \frac{1}{|U|} \sum_{x_i \in U} y_i$$

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

Пример. Чем сложнее дерево (чем больше его глубина), тем выше влияние шумов в данных и выше риск переобучения



Leo Breiman et al. Classification and regression trees. 1984.
`scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html`

Критерий ветвления — неопределённость классификации

$P(y|U) = \frac{1}{|U|}\#\{x_i \in U\colon y_i = y\}$ — распределение U по классам

$Q(U)$ — *неопределённость* (impurity) распределения $P(y|U)$:

$$Q\left(\boxed{\phantom{\rule{1em}{1em}}}\right) < Q\left(\boxed{\phantom{\rule{1em}{1em}}}\right) = Q\left(\boxed{\phantom{\rule{1em}{1em}}}\right) < Q\left(\boxed{\phantom{\rule{1em}{1em}}}\right)$$

1) минимальна и равна нулю, когда $P(y|U) \in \{0, 1\}$,
2) максимальна для равномерного распределения $P(y|U) = \frac{1}{|Y|}$,
3) симметрична: не зависит от перенумерации классов.

$Q(U) = \mathsf{E}_y \mu(P(y|U))$ удовлетворяет этим требованиям,
где $\mu(p)$ — убывающая функция, такая, что $\mu(1) = 0$:

$$Q(U) = \frac{1}{|U|}\sum_{x_i \in U} \mu\big(P(y_i|U)\big) = \sum_{y \in Y} P(y|U)\,\mu\big(P(y|U)\big)$$

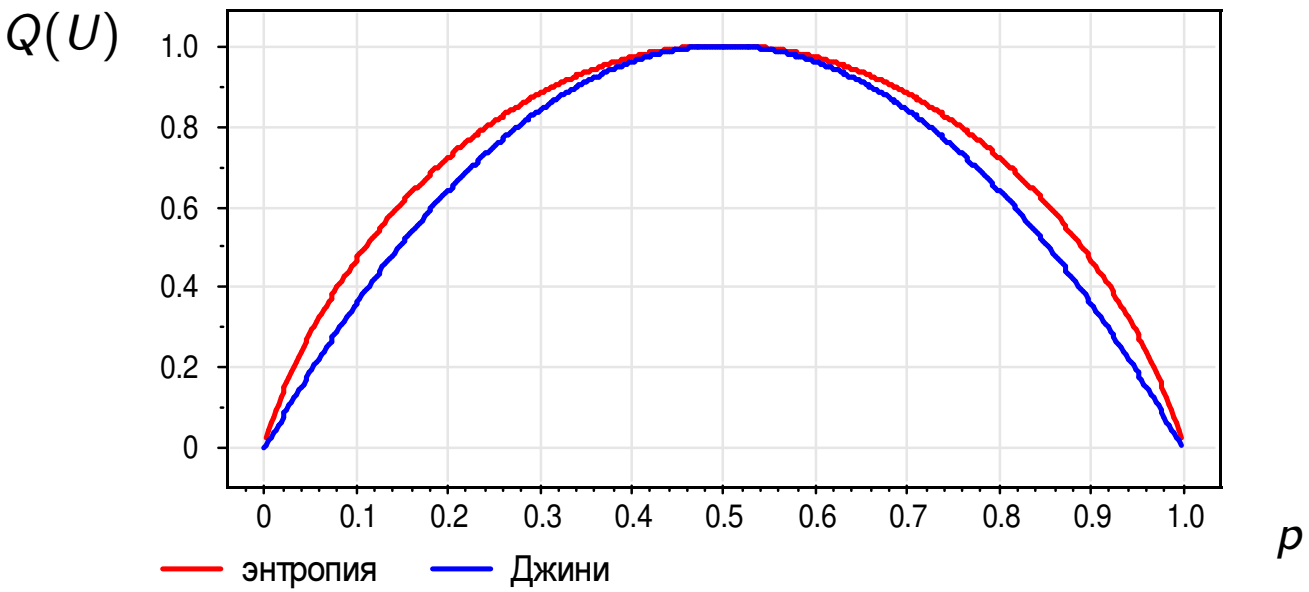
Например, подходят функции $\mu(p) = -\log p,\ \ 1-p,\ \ 1-p^2$

Преимущество: здесь нет минимизации $Q(a, U)$ по $a \in Y$

Критерий Джини и энтропийный критерий

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

- Если $\mu(p) = -\log_2 p$, то
  $Q(U) = -p\log_2 p - (1-p)\log_2(1-p)$ — *энтропия* выборки.
- Если $\mu(p) = 2(1-p)$, то
  $Q(U) = 4p(1-p)$ — *неопределённость Джини* (Gini impurity).



Жадная нисходящая стратегия: достоинства и недостатки

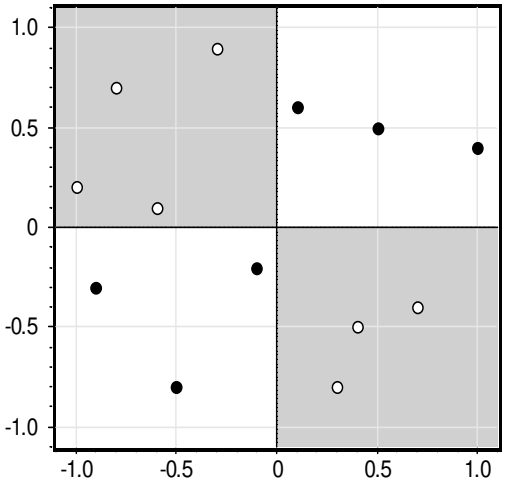
Достоинства:

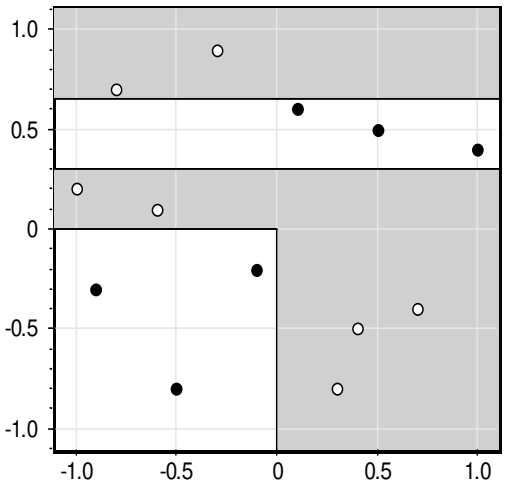
- Интерпретируемость и простота классификации
- Правила $[f_j(x) > \theta]$ не требуют масштабирования признаков
- Допустимы разнотипные данные (через бинаризацию)
- Пропуски в данных не приводят к отказу от классификации
- Трудоёмкость линейна по длине выборки $O(|F|h\ell)$

Недостатки:

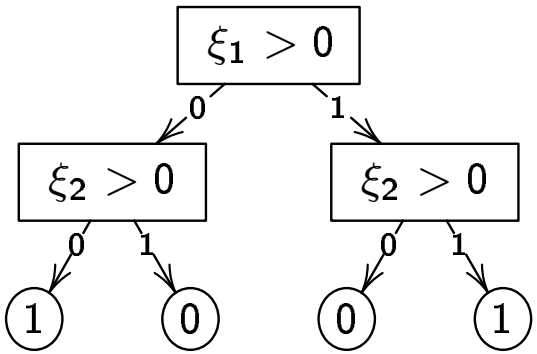
- Жадная стратегия переусложняет структуру дерева, и, как следствие, сильно переобучается
- Фрагментация выборки: чем дальше v от корня, тем меньше статистическая надёжность выбора f_v, y_v
- Высокая чувствительность к шуму, к составу выборки, к критерию ветвления

Жадная стратегия может переусложнять структуру дерева

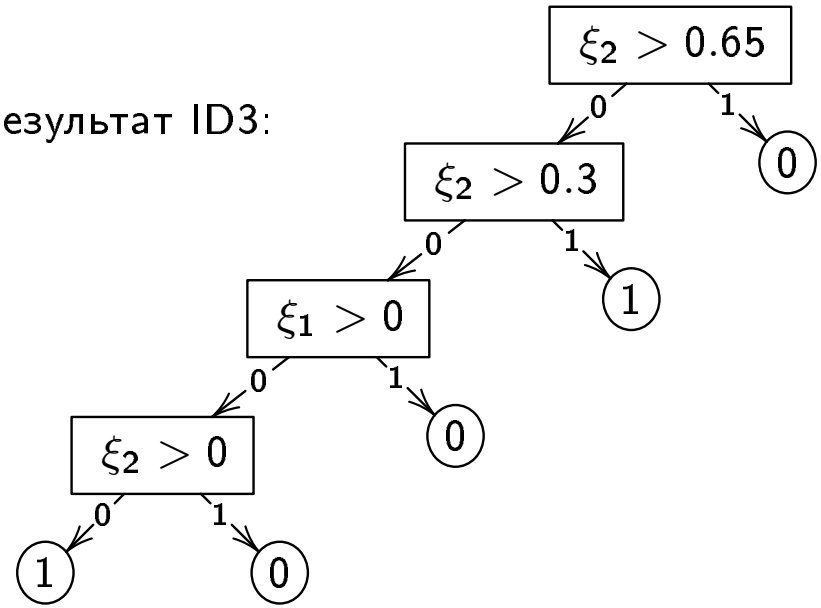




Оптимальное дерево для задачи XOR:



Результат ID3:



Усечение дерева: стратегии post-pruning

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

для всех $v \in V_{\text{внутр}}$:
  $X_v^q :=$ подмножество объектов X^q, дошедших до v;
  если $X_v^q = \varnothing$ **то**
    создать новый лист v; $y_v := \text{Major}(U)$; **вернуть** v;
  по минимуму числа ошибок классификации $Q(X_v^q)$:
    либо сохранить целиком поддерево вершины v;
    либо заменить поддерево v дочерним $S_v(k)$, $k \in D_v$;
    либо заменить поддерево v листом, выбрав класс y_v;

Стратегии перебора вершин:

- снизу вверх: Minimum Cost Complexity Pruning (MCCP), Reduced Error Pruning (REP), Minimum Error Pruning (MEP)
- сверху вниз: Pessimistic Error Pruning (PEP)

CART: критерий Minimal Cost-Complexity Pruning

Среднеквадратичная ошибка со штрафом за сложность дерева:

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

Дерево — линейный классификатор над бинарными признаками:

$$a(x) = \sum_{v \in V_{\text{лист}}} w_v B_v(x),$$

$B_v(x) = \big[\text{объект } x \text{ прошёл путь от корня } v_0 \text{ до листа } v\big]$;
w_y — вес признака B_v, среднее y_i по всем x_i: $B_v(x) = 1$.

При увеличении α дерево последовательно упрощается,
причём последовательность вложенных деревьев единственна.
Из неё выбирается дерево с min ошибкой на тесте (Hold-Out).

Leo Breiman et al. Classification and regression trees. 1984.

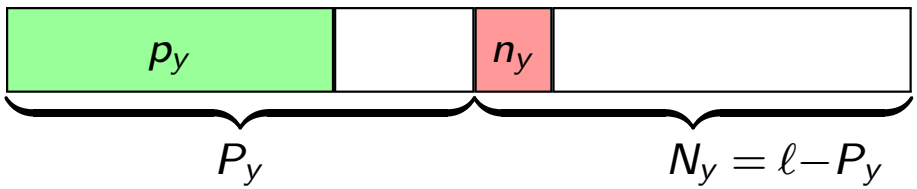
Логические закономерности в задачах классификации

$X^\ell = (x_i, y_i)_{i=1}^\ell \subset X \times Y$ — 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 (не более 7)
2. *информативность* относительно одного из классов $y \in Y$:
   $p_y(R, X^\ell) = \#\{x_i \in X^\ell\colon R(x_i)=1 \text{ и } y_i=y\} \to \max;$
   $n_y(R, X^\ell) = \#\{x_i \in X^\ell\colon R(x_i)=1 \text{ и } y_i\neq y\} \to \min;$

$$\frac{p_y}{P_y} \gg \frac{n_y}{N_y}$$



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

Требование интерпретируемости

1) $R_y(x)$ записывается на естественном языке
2) $R_y(x)$ зависит от небольшого числа признаков (не более 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%

Замечание. *Риск* — частотная оценка вероятности класса, вычисляемая, как правило, по отложенной контрольной выборке

Обучение логических классификаторов

Алгоритмов *индукции правил* (rule induction) очень много!

Основные шаги их построения — надо задать:

1. семейство правил, в котором будем искать закономерности
2. способ порождения правил (rule generation)
3. критерий отбора информативных правил (rule selection)
4. модель классификации с правилами в роли признаков, например, линейный классификатор (weighted voting):

$$a(x) = \arg\max_{y \in Y} \sum_{j=1}^{n_y} w_{yj} R_{yj}(x)$$

Две трактовки понятия «*логическая закономерность*» $R_y(x)$:

- обучаемый информативный интерпретируемый признак
- классификатор одного класса y с отказами

Шаг 1. Часто используемые семейства правил

- Пороговое условие (решающий пень, decision stump):

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

- *Конъюнкция* пороговых условий:

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

- *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 заданного *критерия информативности*.

Шаг 1. Часто используемые семейства правил

- *Полуплоскость* — линейная пороговая функция:

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

- *Шар* — пороговая функция близости:

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

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

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

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

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

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

Шаг 2. Мета-эвристики для поиска информативных правил

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

инициализировать начальное множество правил Z;
повторять
  $Z' :=$ множество *локальных модификаций* правил из Z;
  $Z :=$ наиболее *информативные* правила из $Z \cup Z'$;
пока правила продолжают улучшаться;
вернуть Z;

Частные случаи (см. лекцию про методы отбора признаков):
— стохастический локальный поиск (stochastic local search)
— генетические (эволюционные) алгоритмы
— усечённый поиск в ширину (beam search)
— поиск в глубину (метод ветвей и границ)

Шаг 2. Локальные модификации правил

Пример. Семейство конъюнкций пороговых условий:

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

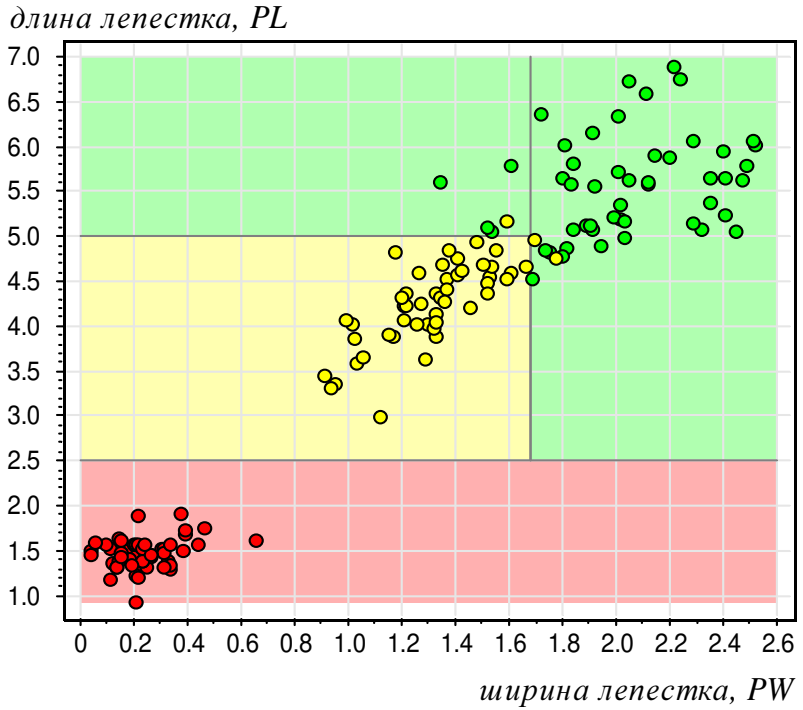
Локальные модификации конъюнктивного правила:

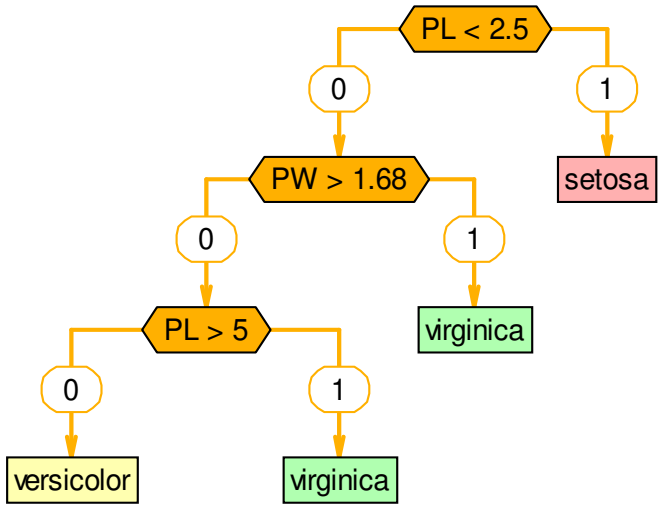
- варьирование одного из порогов a_j и b_j
- варьирование обоих порогов a_j, b_j одновременно
- добавление признака f_j в J с варьированием порогов a_j, b_j
- удаление признака f_j из J

При удалении признака (pruning) информативность обычно оценивается по контрольной выборке (hold-out)

Вообще, для оптимизации множества J подходят те же методы, что и для отбора признаков (feature selection)

Шаг 2. Решающее дерево → покрывающий набор конъюнкций





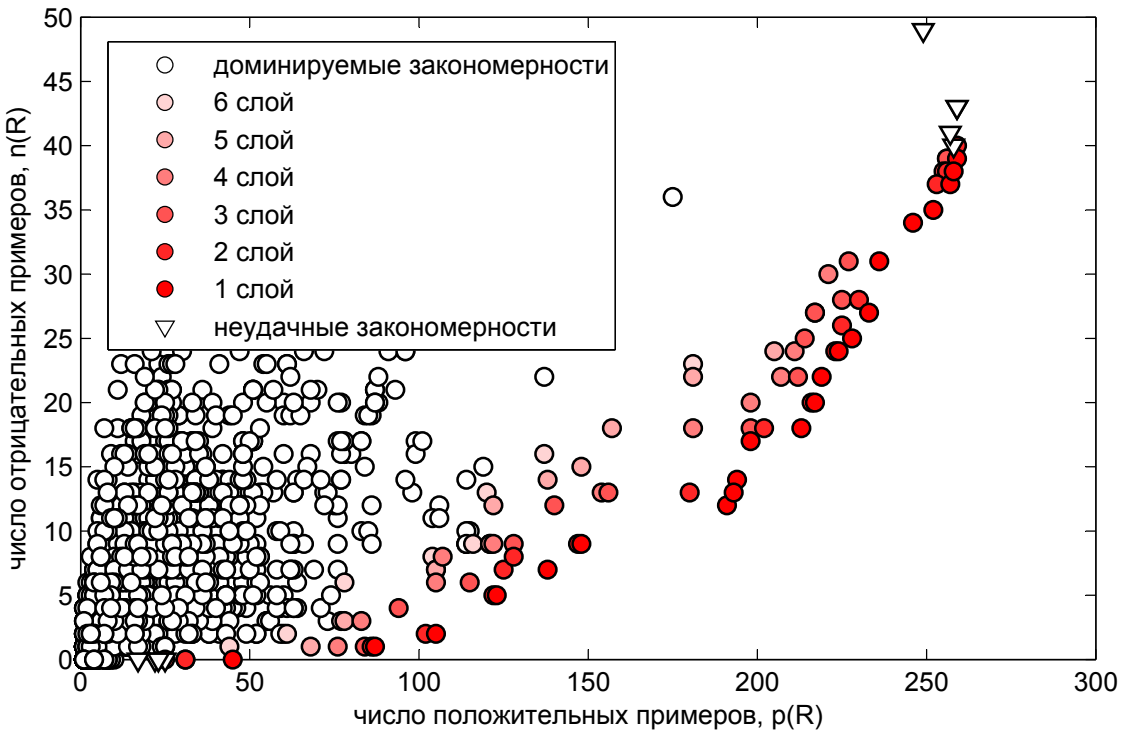
setosa	$r_1(x) = [PL \leqslant 2.5]$
virginica	$r_2(x) = [PL > 2.5] \wedge [PW > 1.68]$
virginica	$r_3(x) = [PL > 5] \wedge [PW \leqslant 1.68]$
versicolor	$r_4(x) = [PL > 2.5] \wedge [PL \leqslant 5] \wedge [PW < 1.68]$

Шаг 3. Двухкритериальный отбор правил в плоскости (p, n)

позитивные: $p_y(R) = \#\{x_i : R(x_i)=1 \text{ и } y_i=y\} \to \max;$

негативные: $n_y(R) = \#\{x_i : R(x_i)=1 \text{ и } y_i\neq y\} \to \min;$

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



UCI:german

Шаг 3. Логические и статистические закономерности

Предикат $R(x)$ — *логическая закономерность* класса $y \in Y$:

$$\text{Precision} = \frac{p_y(R)}{p_y(R) + n_y(R)} \geqslant \pi_0 \qquad \text{Recall} = \frac{p_y(R)}{P_y} \geqslant \rho_0$$

Если $n_y(R) = 0$, то R — *непротиворечивая закономерность*

Предикат $R(x)$ — *статистическая закономерность* класса $y \in Y$:

$$\text{IStat}\big(p_y(R), n_y(R)\big) \geqslant \sigma_0$$

IStat — минус-log вероятности реализации (p, n) при условии нулевой гипотезы, что $y(x)$ и $R(x)$ — независимые случайные величины (точный тест Фишера, Fisher's Exact Test):

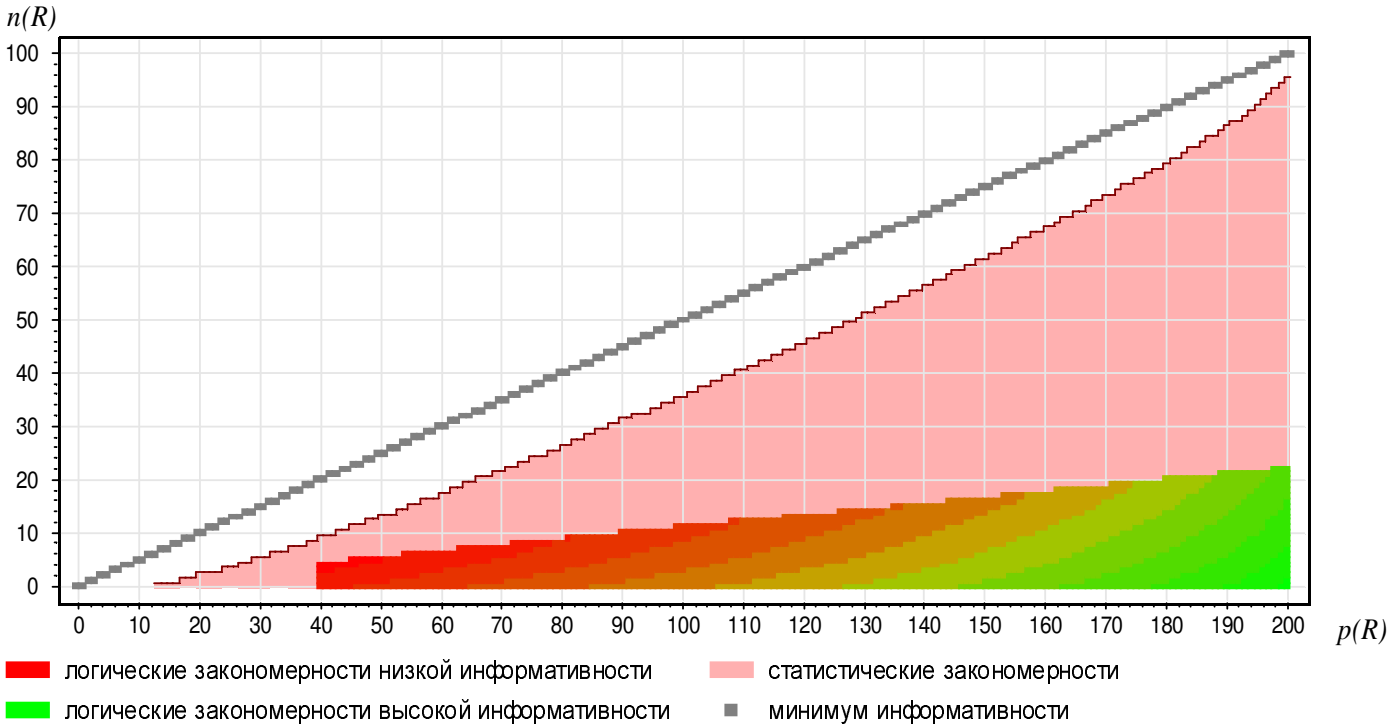
$$\text{IStat}(p, n) = -\frac{1}{\ell} \log_2 \frac{C_P^p C_N^n}{C_{P+N}^{p+n}} \;\to\; \max,$$

где $P = \#\{x_i : y_i = y\}$, $N = \#\{x_i : y_i \neq y\}$, $C_N^n = \frac{N!}{n!(N-n)!}$

Шаг 3. Критерии поиска закономерностей в плоскости (p, n)

Логические закономерности: Precision $\geqslant 0.9$, Recall $\geqslant 0.2$

Статистические закономерности: IStat $\geqslant 3$



$P = 200$

$N = 100$

Логический критерий удобнее для финального отбора правил;
статистический критерий — в процессе модификации правил

Шаг 3. Зоопарк критериев информативности

Очевидные, но не вполне адекватные критерии:

- $I(p,n) = \frac{p}{p+n} \to \max$ (precision)
- $I(p,n) = p/P \to \max$ (recall)
- $I(p,n) = p/P - n/N \to \max$ (relative accuracy)

Адекватные, но не очевидные критерии:

- энтропийный критерий прироста информации:
  $$\text{IGain}(p,n) = h\Big(\frac{P}{\ell}\Big) - \frac{p+n}{\ell} h\Big(\frac{p}{p+n}\Big) - \frac{\ell-p-n}{\ell} h\Big(\frac{P-p}{\ell-p-n}\Big) \to \max$$
  где $h(q) = -q \log_2 q - (1-q)\log_2(1-q)$
- критерий Джини (Gini impurity):
  $\text{IGain}(p,n)$ при $h(q) = 4q(1-q)$
- критерий бустинга и его нормированный вариант:
  $\sqrt{p} - \sqrt{n} \to \max$, $\sqrt{p/P} - \sqrt{n/N} \to \max$

J.Fürnkranz, P.Flach. ROC'n'rule learning – towards a better understanding of covering algorithms // Machine Learning, 2005.

Шаг 3. Нетривиальность проблемы свёртки двух критериев

Пример: в каждой паре правил первое гораздо лучше второго, однако простые эвристики не различают их по качеству (при $P = 200$, $N = 100$).

p	n	$p-n$	$p-5n$	$\frac{p}{P}-\frac{n}{N}$	$\frac{p}{n+1}$	IStat$\cdot\ell$	IGain$\cdot\ell$	$\sqrt{p}-\sqrt{n}$
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

Замечание. Критерии IStat и IGain асимптотически эквивалентны: $\text{IStat}(p, n) \to \text{IGain}(p, n)$ при $\ell \to \infty$

Шаг 4. Построение классификатора из закономерностей

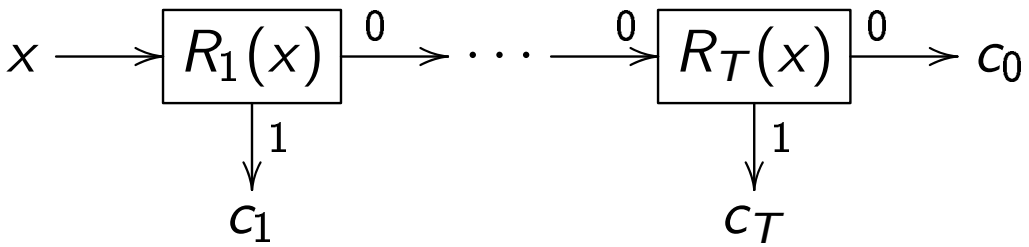
Взвешенное голосование (линейный классификатор с весами w_{yt} и, возможно, с регуляризацией для отбора признаков):

$$a(x) = \arg\max_{y \in Y} \sum_{t=1}^{T_y} w_{yt} R_{yt}(x)$$

Простое голосование (комитет большинства):

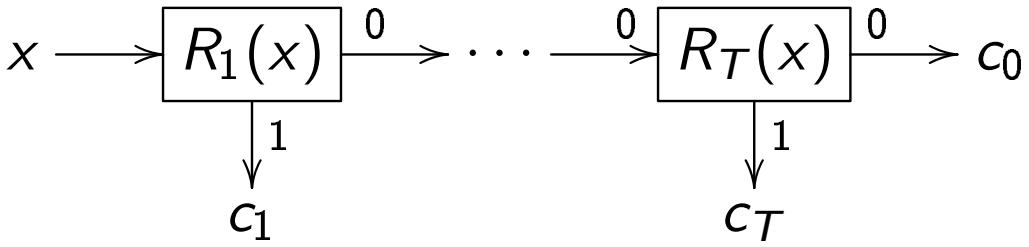
$$a(x) = \arg\max_{y \in Y} \frac{1}{T_y} \sum_{t=1}^{T_y} R_{yt}(x)$$

Решающий список (комитет старшинства), $c_0, c_1, \ldots, c_T \in Y$:



Определение решающего списка (Decision List, DL)

DL — это алгоритм классификации $a\colon X \to Y$, задаваемый закономерностями $R_1(x), \ldots, R_T(x)$ классов $c_1, \ldots, c_T \in Y$:



Это способ представления знаний в виде *системы продукций* — последовательности правил «**если**-условие **то**-решение»

для всех $t = 1, \ldots, T$
  если $R_t(x) = 1$ **то вернуть** c_t;
вернуть c_0 (отказ от классификации объекта x);

$$E(R_t, X^\ell) = \frac{n_{c_t}(R_t)}{n_{c_t}(R_t) + p_{c_t}(R_t)} \to \min \quad \text{— доля ошибок } R_t \text{ на } X^\ell$$

Жадный алгоритм обучения решающего списка

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

$U := X^\ell$;
для всех $t := 1, \ldots, T_{\max}$
- выбрать класс c_t;
- поиск правила R_t по максимуму информативности:
  $R_t := \arg\max_R I(R, U)$ при ограничении $E(R, U) \leqslant E_{\max}$;
- **если** $I(R_t, U) < I_{\min}$ **то выход**;
- $U := \{x \in U : R_t(x) = 0\}$ — не покрытые правилом R_t;
- **если** $|U| \leqslant \ell_0$ **то выход**;

Ronald Rivest. Learning decision lists // Machine Learning, 1987.

Замечания к алгоритму построения решающего списка

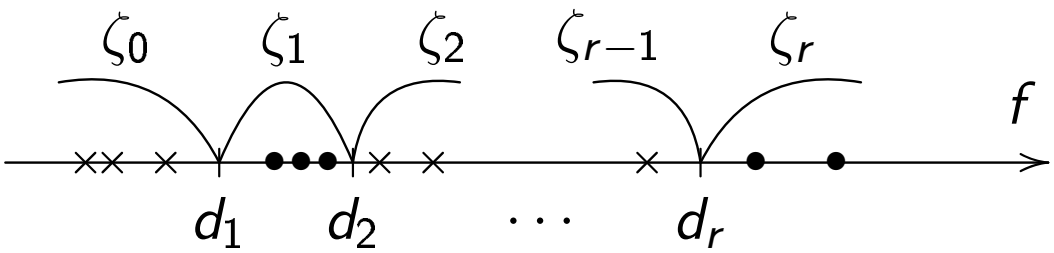
- **Стратегии выбора класса c_t:**
  1) все классы по очереди (лучше для интерпретируемости)
  2) на каждом шаге определяется оптимальный класс
- Параметр $E_{\max}$ управляет сложностью списка:
  $E_{\max} \downarrow \quad \Rightarrow \quad p(R_t) \downarrow,\ T \uparrow$
- **Преимущества:**
  — интерпретируемость модели и классификаций
  — простой обход проблемы пропусков в данных
- **Недостаток:** низкое качество классификации
- **Другие названия:**
  комитет с логикой старшинства (Majority Committee)
  голосование по старшинству (Majority Voting)
  машина покрывающих множеств (Set Covering Machine, SCM)

Вспомогательная задача бинаризации вещественного признака

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

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 «удобных» значений
4. Объединение нескольких разбиений

Выбор «удобных» пороговых значений

Задача: на отрезке $[a, b]$ найти значение x^* с минимальным числом значащих цифр.

Если таких x^* несколько, выбрать

$$x^* = \arg\min_x \left|\tfrac{1}{2}(a+b) - x\right|.$$

a =	2,16667
	2,19
x^* =	2,2
	2,21
$(a+b)/2$ =	2,23889
	2,29
	2,3
	2,31
b =	2,31111

Жадный алгоритм слияния зон по критерию информативности

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 $d_i \in D$, $i = 1, \dots, |D| - 1$
    $\delta I_i := I(\zeta_{i-1} \vee \zeta_i \vee \zeta_{i+1}) - \max\{I(\zeta_{i-1}), I(\zeta_i), I(\zeta_{i+1})\}$;
  $i := \arg\max_s \delta I_s$;
  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 $|D| > r + 1$;

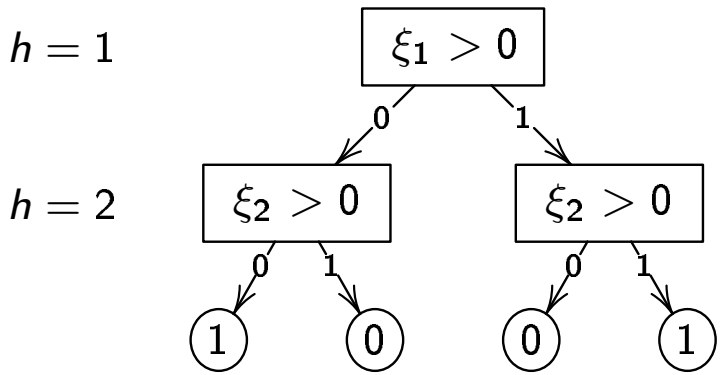
Небрежные решающие деревья (Oblivious Decision Tree, ODT)

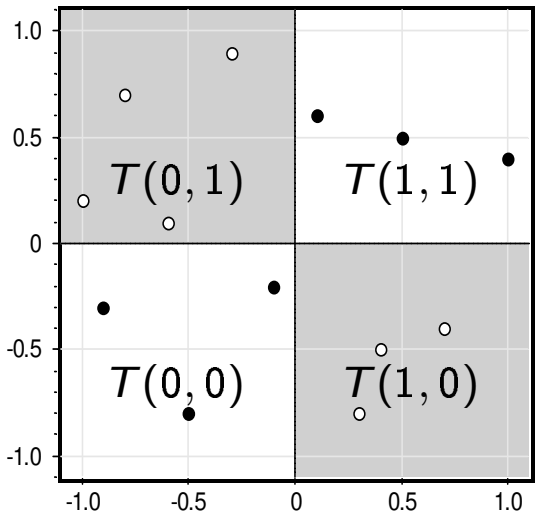
Решающая таблица: дерево глубины H, $D_v = \{0,1\}$;
для всех узлов уровня h условие ветвления $f_h(x)$ *одинаково*;
на уровне h ровно 2^{h-1} вершин; X делится на 2^H ячеек.

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, $H = 2$.





R.Kohavi, C.-H.Li. Oblivious decision trees, graphs, and top-down pruning. 1995.

Жадный алгоритм обучения ODT

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

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

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

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

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

Резюме по логическим методам

- Эмпирическая индукция — вывод знаний из данных:
  - индукция правил (Rule Induction)
  - решающие деревья, списки, таблицы
- Преимущества логических методов:
  - интерпретируемость
  - возможность обработки разнотипных данных
  - возможность обработки данных с пропусками
- Недостатки логических методов:
  - ограниченное качество классификации
  - решающие деревья неустойчивы, склонны к переобучению
- Способы устранения этих недостатков:
  - редукция по тестовым данным
  - композиции правил, леса деревьев (в следующих лекциях)