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-24-25
орг.вопросы по курсу: ml.cmc@mail.ru

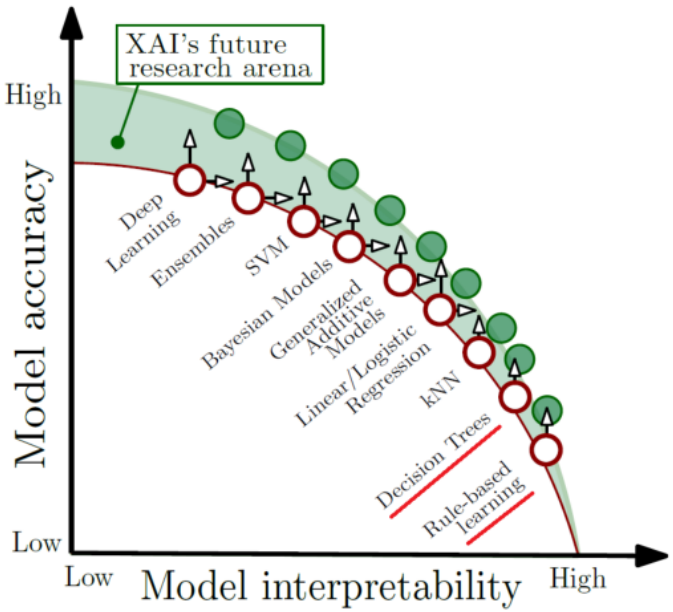
ВМК МГУ • 19 ноября 2024

Содержание

Интерпретируемость (XAI, eXplainable Artificial Intelligence)

Обучающая выборка $X^\ell = (x_i, y_i)_{i=1}^\ell$, $y_i \in Y$ — метки классов

- **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

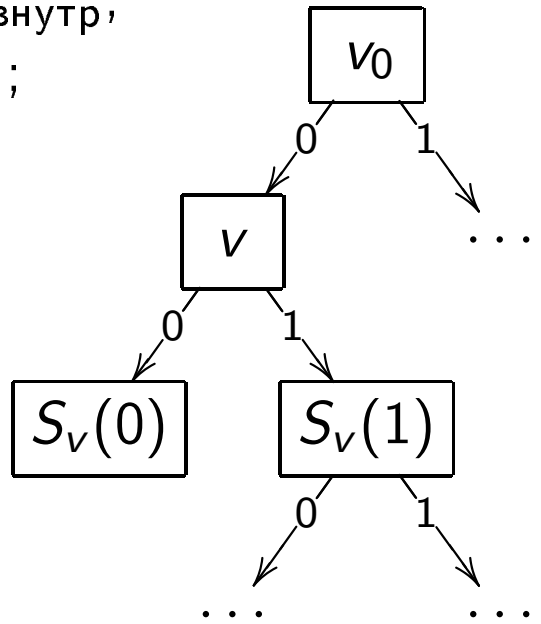
"Do you want an interpretable model, or the one that works?"
[Yann LeCun, NIPS'17]

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

Определение решающего дерева (Decision Tree)

Решающее дерево — алгоритм классификации $a(x)$, задающийся *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, $\forall v \in V_{\text{внутр}}$;
$S_v : D_v \to V$ — множество дочерних вершин;
$y_v \in Y$ — метка класса, $\forall v \in V_{\text{лист}}$;



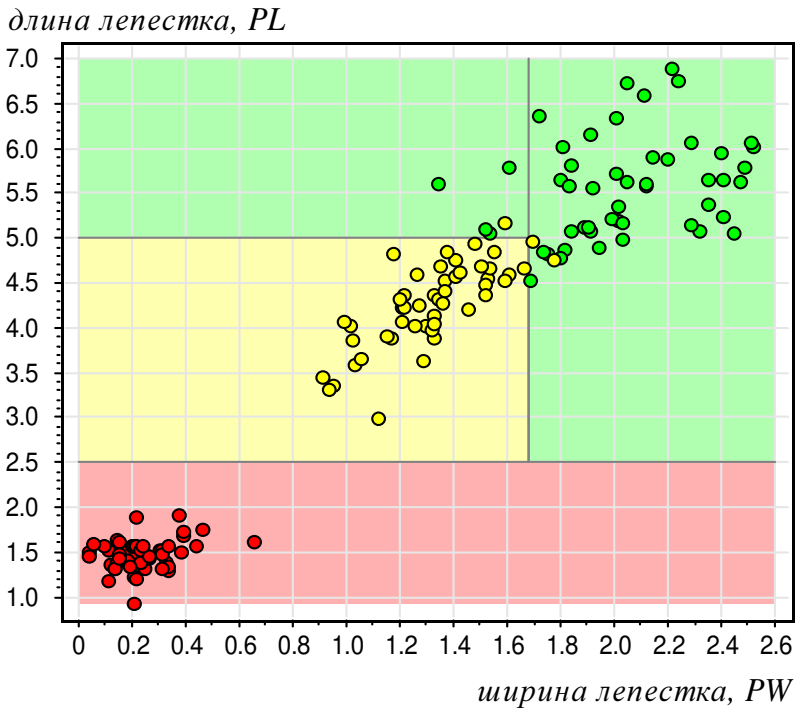
$v := v_0$;
пока $(v \in V_{\text{внутр}})$: $v := S_v(f_v(x))$;
вернуть $a(x) := y_v$;

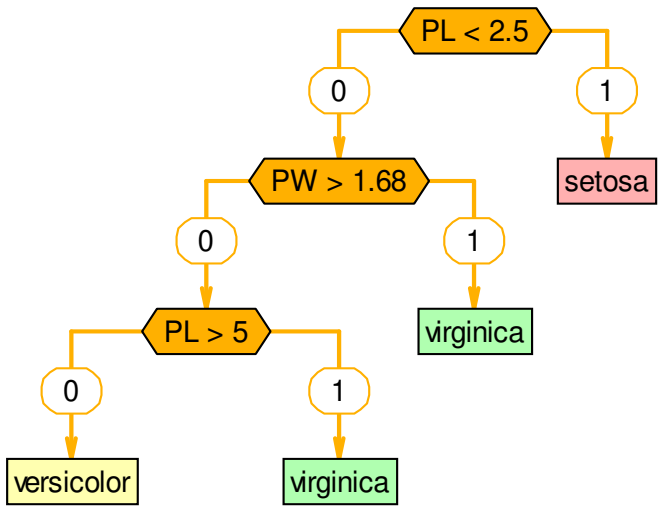
Чаще всего используются бинарные признаки вида $f_v(x) = [f_j(x) \geqslant a]$

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)

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

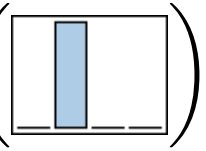
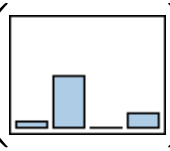
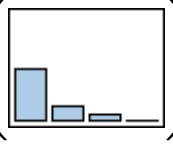
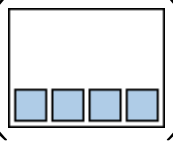
Мажоритарное правило: $\text{Major}\,(U) := \arg\max\limits_{y \in Y} P(y|U)$.

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

Неопределённость распределения по классам в вершине

$p_y = \frac{1}{|U|} \sum_{x_i \in U} [y_i = y]$ — частотная оценка вероятности $P(y|U)$

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

$$\Phi\left(\text{  }\right) < \Phi\left(\text{  }\right) = \Phi\left(\text{  }\right) < \Phi\left(\text{  }\right)$$

1) минимальна и равна нулю, когда $p_y \in \{0, 1\}$,
2) максимальна, когда $p_y = \frac{1}{|Y|}$ — равномерное распределение,
3) симметрична: не зависит от перенумерации классов.

Этим требованиям удовлетворяет функция вида

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

где функция потерь $\mathscr{L}(p_y)$ убывающая, $\mathscr{L}(1) = 0$.

Например, подходят функции $\mathscr{L}(p) = -\log p, \;\; 1-p, \;\; 1-p^2$

Критерий ветвления

Признак f разбивает U на подвыборки $U_k = \{x \in U\colon f(x) = k\}$

Знание признака f меняет неопределённость, так как вместо распределений $P(y|U)$ теперь известны $P(y|U_k)$:

$$\Phi(U|f) = \frac{1}{|U|}\sum_{x_i \in U} \mathscr{L}\big(P(y_i|U_{f(x_i)})\big) =$$

$$= \frac{1}{|U|}\sum_k \sum_{x_i \in U_k} \mathscr{L}\big(P(y_i|U_k)\big) = \sum_k \frac{|U_k|}{|U|}\Phi(U_k)$$

Выигрыш от ветвления в вершине v по признаку f:

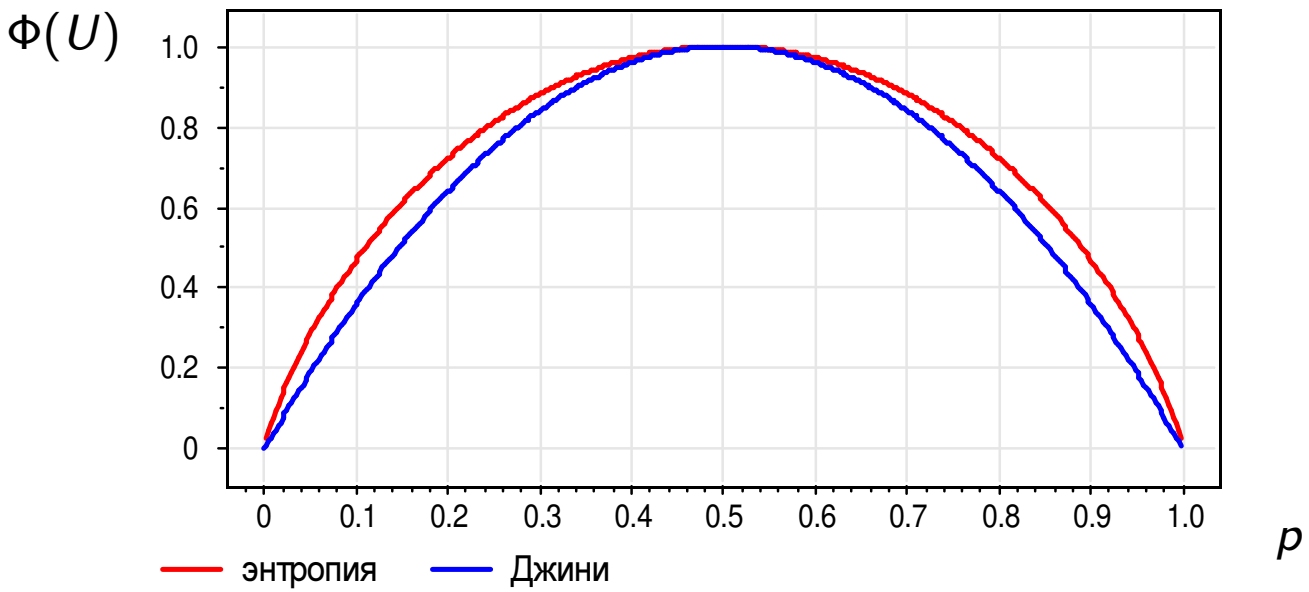
$$\text{Gain}\,(f, U) = \Phi(U) - \Phi(U|f) =$$

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

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



Обработка пропущенных значений

На стадии обучения:

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

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**

средневзвешенное распределение по всем дочерним:
$P(y|x, v) = \sum_{k \in D_v} q_{vk} P(y|x, S_v(k))$;

иначе

$P(y|x, v) = P(y|x, s)$ из дочерней вершины $s = S_v(f_v(x))$;

Жадная нисходящая стратегия: достоинства и недостатки

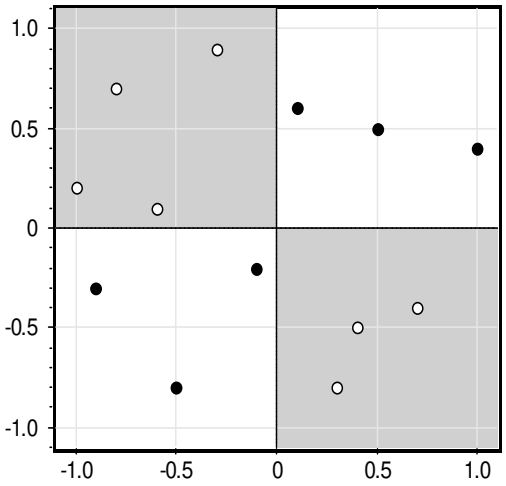
Достоинства:

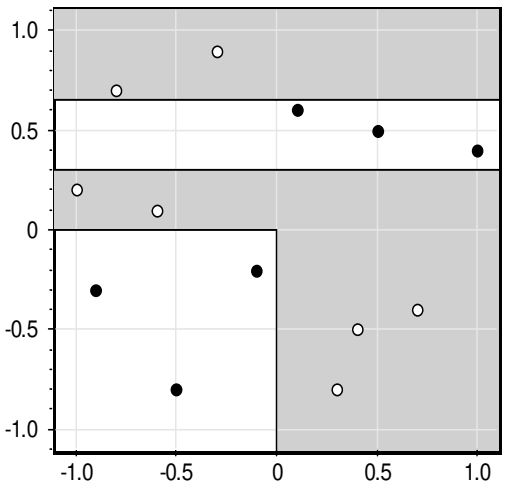
- Интерпретируемость и простота классификации
- Правила $[f_j(x) < \alpha]$ не требуют масштабирования признаков
- Допустимы разнотипные данные и данные с пропусками
- Трудоёмкость линейна по длине выборки $O(|F|h\ell)$
- Не бывает отказов от классификации

Недостатки:

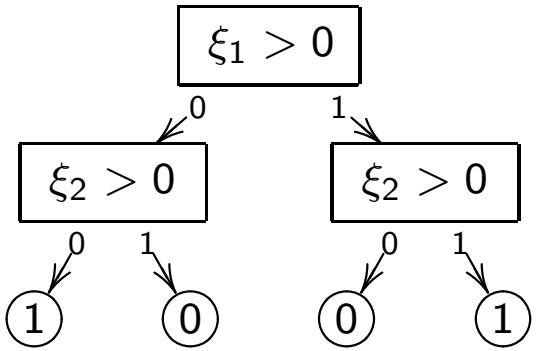
- Жадная стратегия переусложняет структуру дерева, и, как следствие, сильно переобучается
- Фрагментация выборки: чем дальше v от корня, тем меньше статистическая надёжность выбора f_v, y_v
- Высокая чувствительность к шуму, к составу выборки, к критерию информативности

Жадная стратегия может переусложнять структуру дерева

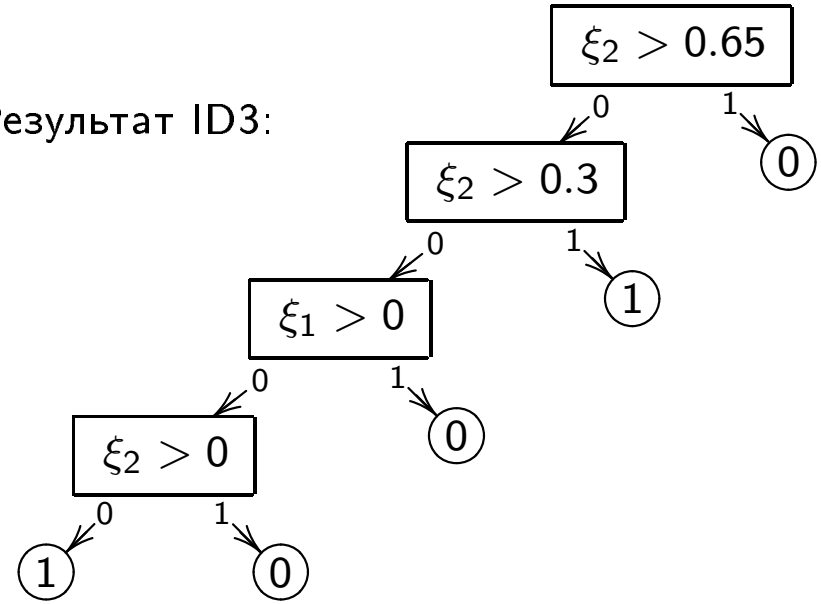




Оптимальное дерево для задачи XOR:



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: деревья регрессии и классификации

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

$$C(a) = \sum_{i=1}^{\ell} \big(a(x_i) - y_i\big)^2 \to \min_a$$

Пусть U — множество объектов x_i, дошедших до вершины v
Мера неопределённости — среднеквадратичная ошибка

$$\Phi(U) = \min_{y \in Y} \frac{1}{|U|} \sum_{x_i \in U} \big(y - y_i\big)^2$$

Значение y_v в терминальной вершине v — МНК-решение:

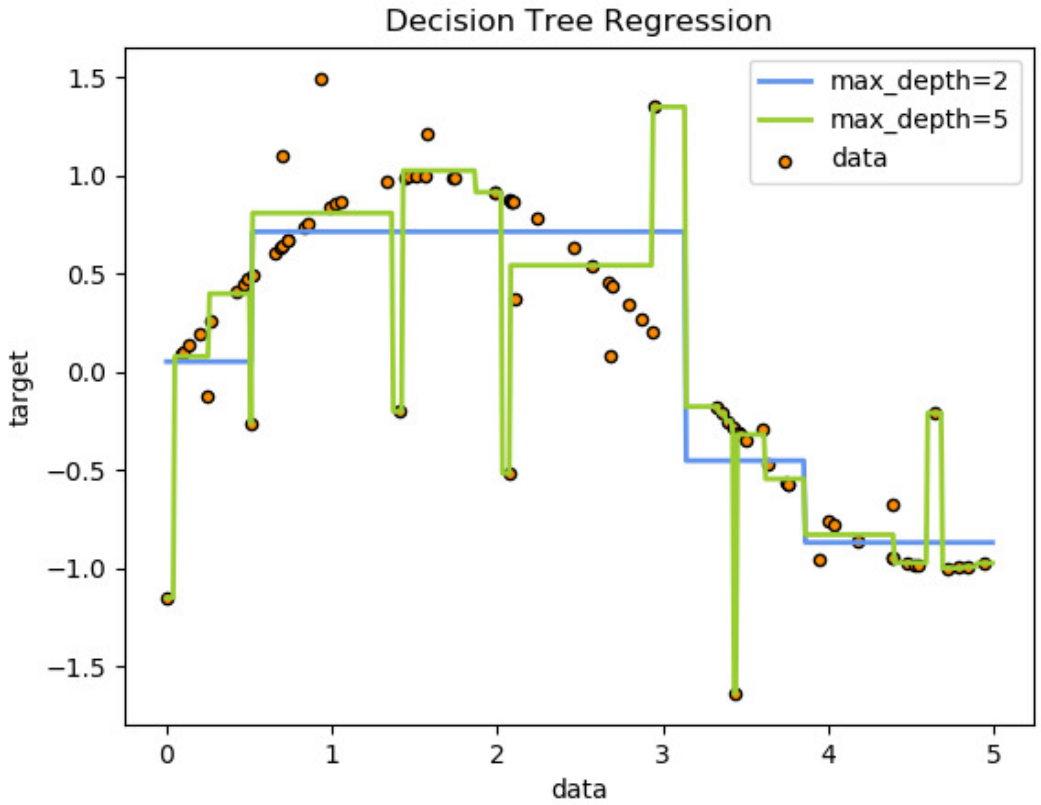
$$y_v = \frac{1}{|U|} \sum_{x_i \in U} y_i$$

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

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

Пример. Деревья регрессии различной глубины

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



scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html

CART: критерий Minimal Cost-Complexity Pruning

Среднеквадратичная ошибка со штрафом за сложность дерева:

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

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.

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

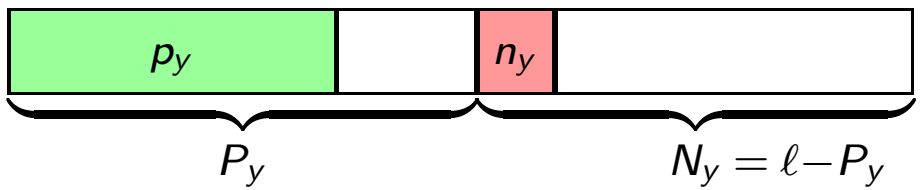
Логические закономерности в задачах классификации

$X^\ell = (x_i, y_i)_{i=1}^\ell \subset X \times Y$ — 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) $R_y(x)$ относит покрываемые объекты к заданному классу y
2) $R_y(x)$ записывается на естественном языке
3) $R_y(x)$ 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!

Три основных шага их построения:

1. Выбор семейства правил для поиска закономерностей
2. Порождение правил (rule generation)
3. Отбор информативных правил (rule selection)
4. Построение классификатора из правил как из признаков, например, линейного классификатора (weighted voting):

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

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

- высокоинформативный интерпретируемый признак
- классификатор одного класса y с отказами

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*.

Часто используемые семейства правил

- *Полуплоскость* — линейная пороговая функция:

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

- *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 заданного *критерия информативности*.

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)

Локальные модификации правил

Пример. Семейство конъюнкций пороговых условий:

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

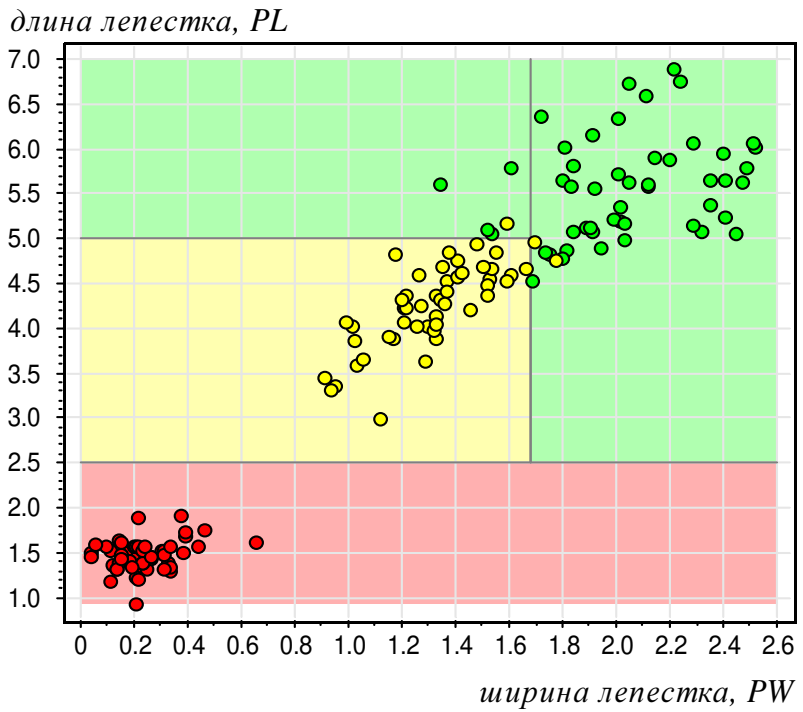
Локальные модификации конъюнктивного правила:

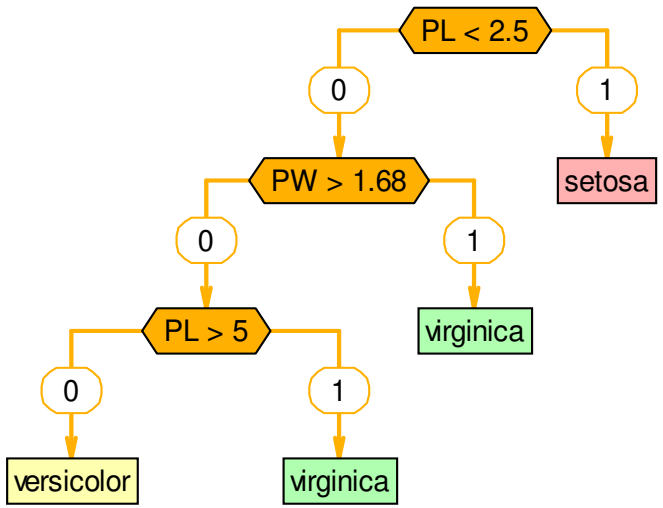
- добавление признака f_j в J с варьированием порогов a_j, b_j
- удаление признака f_j из J
- варьирование одного из порогов a_j и b_j
- варьирование обоих порогов a_j, b_j одновременно

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)

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 классификатора на закономерностях

Взвешенное голосование (Weighted Voting):
много-классовый линейный классификатор с весами w_{yt}
(возможна L_1-регуляризация для отбора закономерностей)

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

Простое голосование (Simple Voting, комитет большинства):

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

Решающий список (Majority Voting, комитет старшинства):
обучаемая *система продукций* — правил «**если** $R_{yt}(x)$ **то** y»

для всех $t = 1, \ldots, T$: **если** $R_{yt}(x) = 1$ **то вернуть** y;

Жадный алгоритм обучения решающего списка (Decision List)

(DL, он же Majority Committee и Set Covering Machine, SCM)
$I\big(p_y(R,U), n_y(R,U)\big)$ — информативность $R(x)$ на выборке U

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

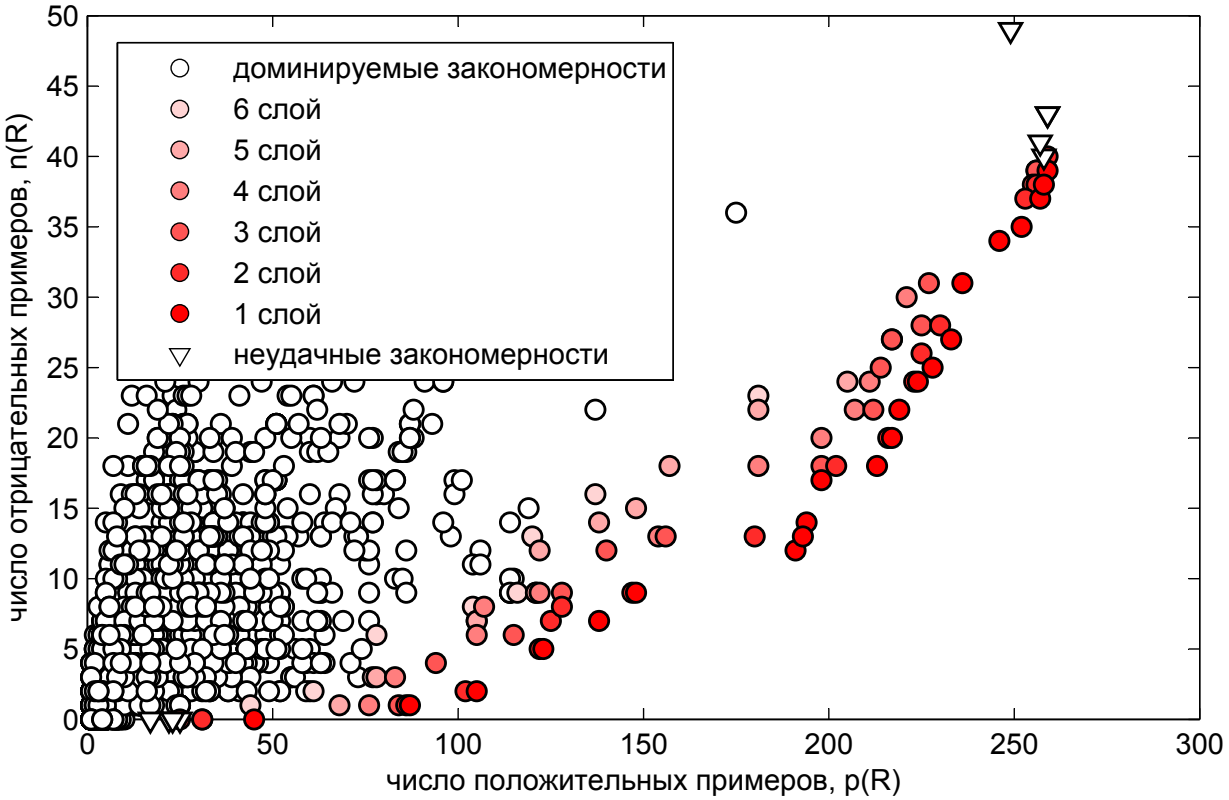
$U := X^\ell$ — инициализация;
для всех $t := 1, \ldots, T_{\max}$
  $R_{yt} := \arg\max\limits_{R,\ y \in Y} I\big(p_y(R,U), n_y(R,U)\big)$;
  если нет хороших правил $(I < I_{\min})$ **то выход**;
  $U := \{x \in U : R_{yt}(x) = 0\}$ — ещё не покрытые объекты;
  если $|U| \leqslant \ell_0$ **то выход**;

Ronald Rivest. Learning decision lists // Machine Learning, 1987.
Mario Marchand, John Shawe-Taylor. The set covering machine // JMLR, 2002.

Двухкритериальный отбор закономерностей

Два критерия: $p_y(R) \to \max, \quad n_y(R) \to \min$

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



UCI:german

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 $y \in Y$:

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

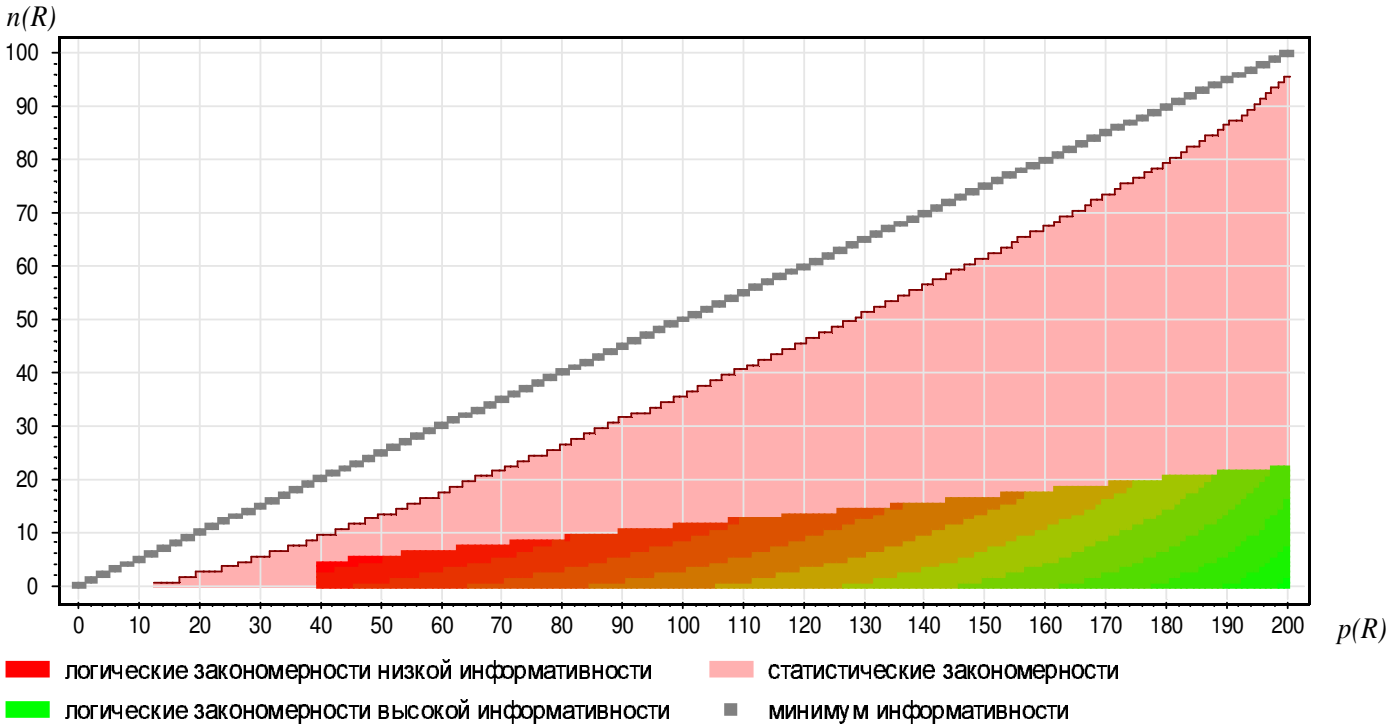
IStat — минус-log вероятности реализации (p, n) при условии нулевой гипотезы, что $y(x)$ и $R(x)$ — независимые случайные величины (точный тест Фишера, Fisher's Exact Test):

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

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

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

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.

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 $\ell \to \infty$

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)