Решающие деревья

К. В. Воронцов, А.В. Зухба
vokov@forecsys.ru
a__l@mail.ru

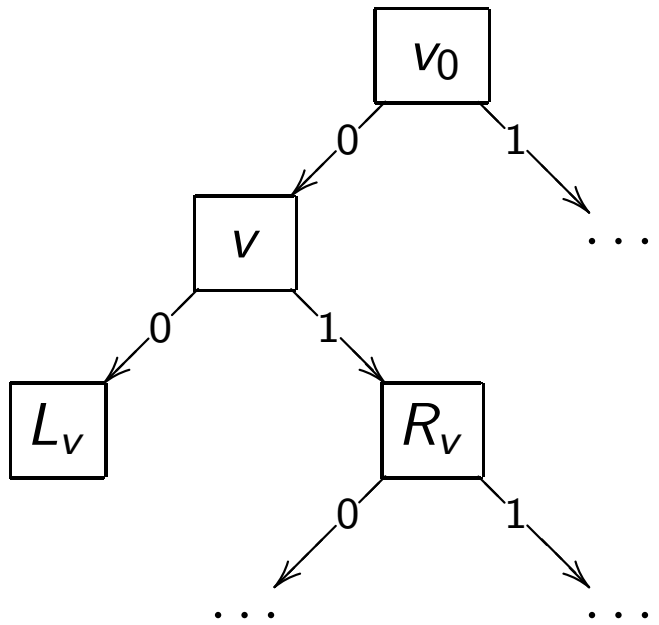
март 2015

Определение бинарного решающего дерева

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

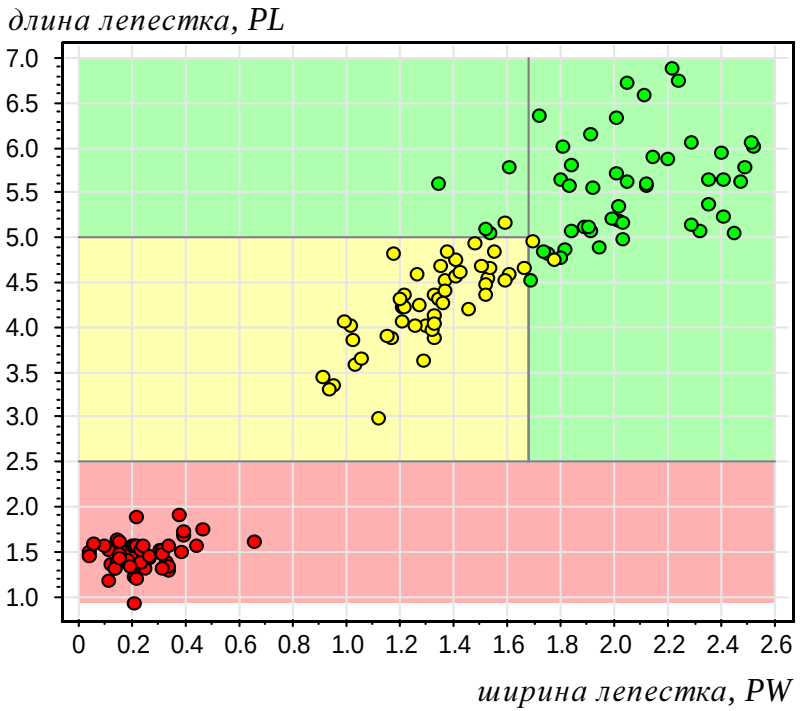
1) $\forall v \in V_{\text{внутр}} \;\rightarrow\;$ предикат $\beta_v : X \to \{0,1\},\;\; \beta_v \in \mathscr{B}$
2) $\forall v \in V_{\text{лист}} \;\rightarrow\;$ имя класса $c_v \in Y$.

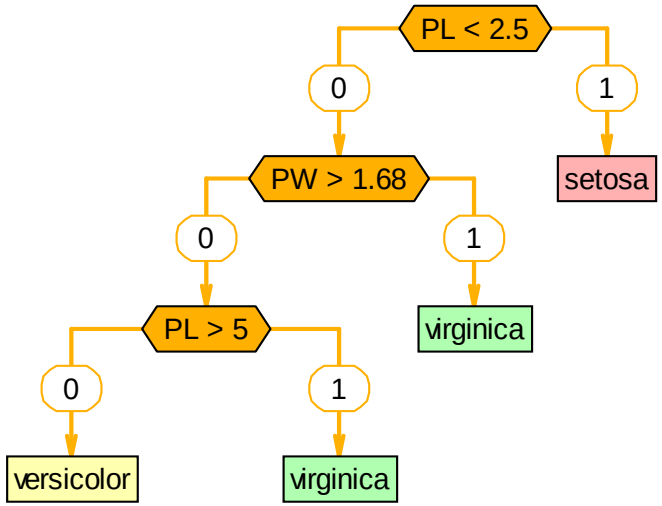
1: $v := v_0$;
2: **пока** $v \in V_{\text{внутр}}$
3: **если** $\beta_v(x) = 1$ **то**
4: переход вправо:
$v := R_v$;
5: **иначе**
6: переход влево:
$v := L_v$;
7: **вернуть** c_v.



Пример решающего дерева

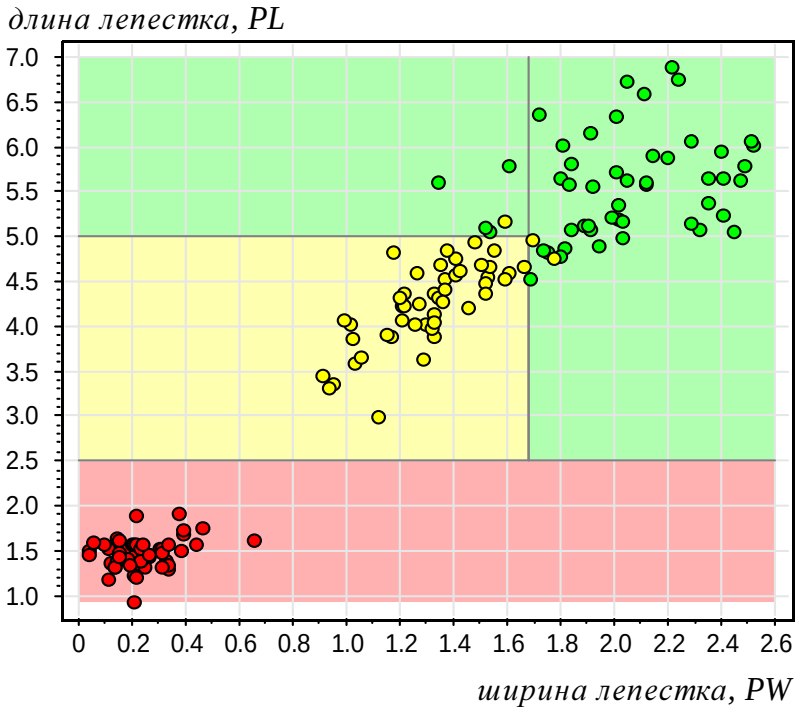
Задача Фишера о классификации цветков ириса на 3 класса, в выборке по 50 объектов каждого класса, 4 признака.

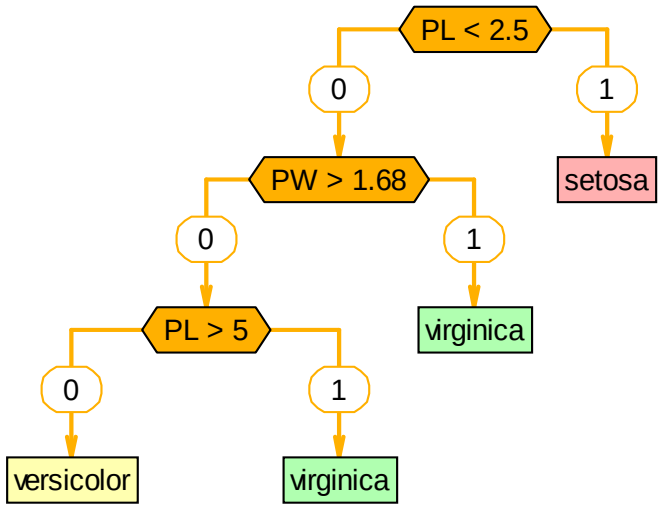




На графике: в осях двух самых информативных признаков (из 4) два класса разделились без ошибок, на третьем 3 ошибки.

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	$r_4(x) = [PL > 2.5] \wedge [PL \leqslant 5] \wedge [PW < 1.68]$

Жадный алгоритм построения дерева ID3

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

Разновидности многоклассовых 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:**

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

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:**

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

На стадии обучения:

- $\beta_v(x)$ не определено $\Rightarrow$ x_i исключается из U для $I(\beta, U)$
- $q_v = \frac{|U_0|}{|U|}$ — оценка вероятности левой ветви, $\forall v \in V_{\text{внутр}}$
- $P_v(y|x) = \frac{1}{|U|}\#\{x_i \in U\colon y_i = y\}$ для всех $v \in V_{\text{лист}}$

На стадии классификации:

- $\beta_v(x)$ определено $\Rightarrow$ либо налево, либо направо:
  $P_v(y|x) = \big(1-\beta_v(x)\big)P_{L_v}(y|x) + \beta_v(x)P_{R_v}(y|x)$.
- $\beta_v(x)$ не определено $\Rightarrow$ *пропорциональное распределение*:
  $P_v(y|x) = q_v P_{L_v}(y|x) + \big(1-q_v\big)P_{R_v}(y|x)$.
- Окончательное решение — наиболее вероятный класс:
  $$y = \arg\max_{y\in Y} P_{v_0}(y|x).$$

Решающие деревья ID3: достоинства и недостатки

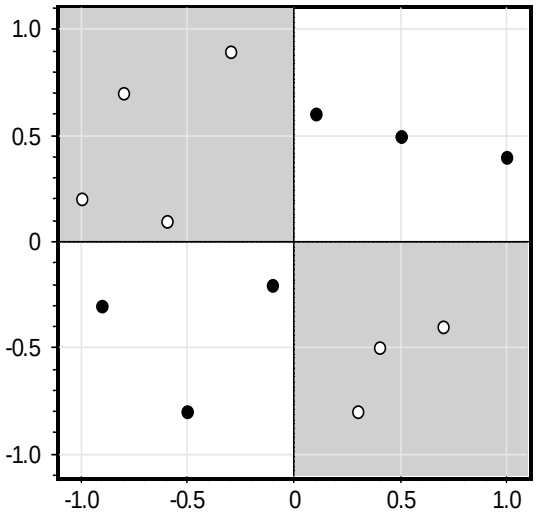
Достоинства:

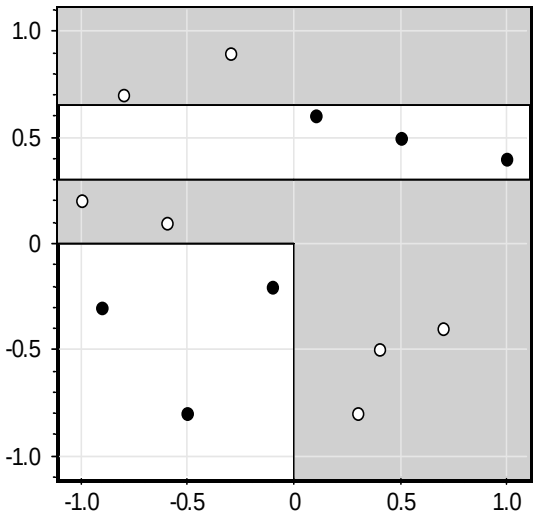
- Интерпретируемость и простота классификации.
- Гибкость: можно варьировать множество $\mathscr{B}$.
- Допустимы разнотипные данные и данные с пропусками.
- Трудоёмкость линейна по длине выборки $O(|\mathscr{B}|h\ell)$.
- Не бывает отказов от классификации.

Недостатки:

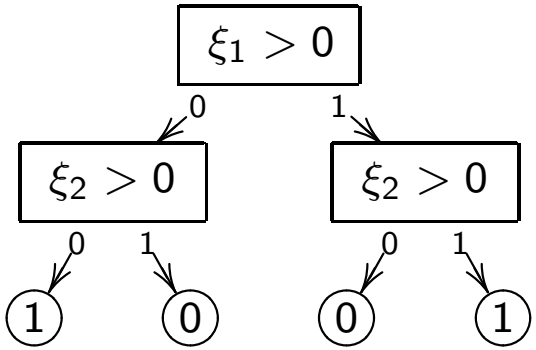
- Жадный ID3 переусложняет структуру дерева, и, как следствие, сильно переобучается.
- Фрагментация выборки: чем дальше v от корня, тем меньше статистическая надёжность выбора β_v, c_v.
- Высокая чувствительность к шуму, к составу выборки, к критерию информативности.

Жадный ID3 переусложняет структуру дерева

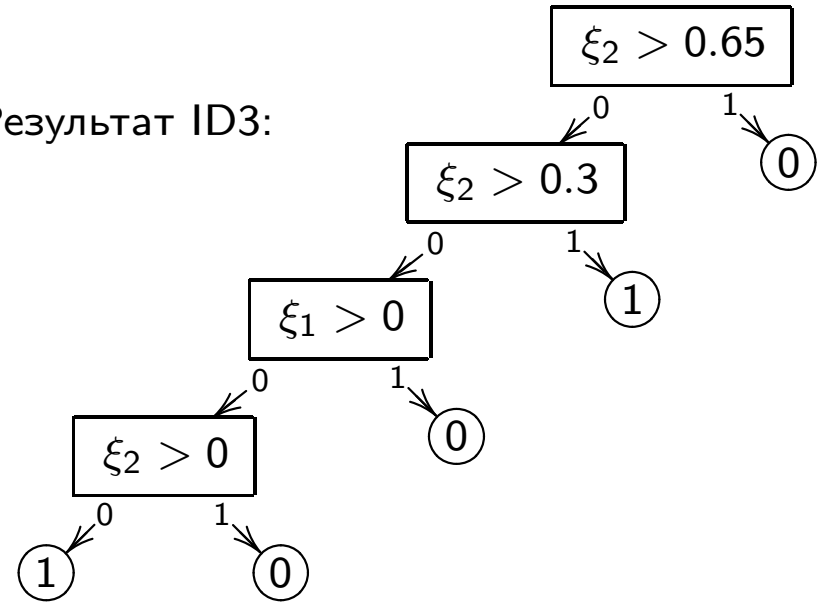




Оптимальное дерево для задачи XOR:



Результат ID3:



Усечение дерева (pruning). Алгоритм C4.5

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, $c_v := \arg\min\limits_{c \in Y} r_c(v)$.

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

Пусть U_v — множество объектов x_i, дошедших до вершины v

Значения в терминальных вершинах — МНК-решение:

$$c_v := \hat{y}(U_v) = \frac{1}{|U_v|} \sum_{x_i \in U_v} y_i$$

Критерий информативности — среднеквадратичная ошибка

$$I(\beta, U_v) = \sum_{x_i \in U_v} \big(\hat{y}_i(\beta) - y_i\big)^2,$$

где $\hat{y}_i(\beta) = \beta(x_i)\hat{y}(U_{v1}) + \big(1 - \beta(x_i)\big)\hat{y}(U_{v0})$
— прогноз после ветвления β и разбиения $U_v = U_{v0} \sqcup U_{v1}$

CART: критерий Minimal Cost-Complexity Pruning

Среднеквадратичная ошибка со штрафом за сложность дерева

$$C_\alpha = \sum_{x_i=1}^{\ell} \left(\hat{y}_i - y_i\right)^2 + \alpha |V_{\text{лист}}| \to \min$$

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.

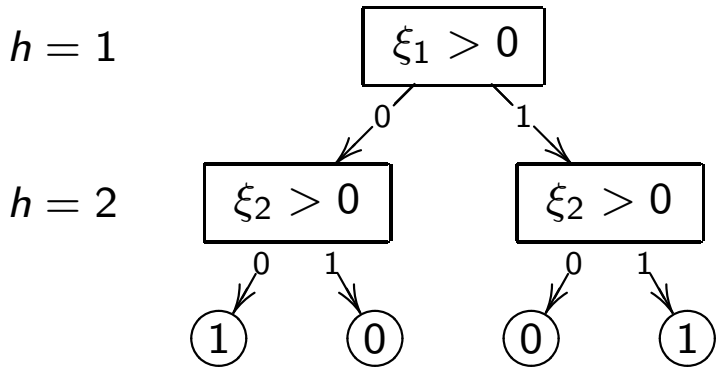
Небрежные решающие деревья — ODT (Oblivious Decision Tree) [1991]

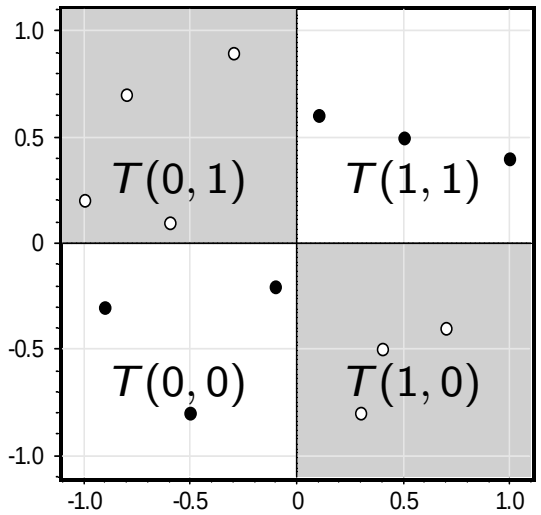
Строится сбалансированное дерево высоты H;
для всех узлов уровня h условие ветвления $\beta_h(x)$ *одинаково*;
на уровне h ровно 2^{h-1} вершин; X делится на 2^H ячеек.

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:

$$T(b_1, \ldots, b_H) := \arg\max_{c \in Y} \sum_{i=1}^{\ell} [y_i = c] \prod_{h=1}^{H} \left[\beta_h(x_i) = b_h\right];$$

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

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

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 повторениями
- в каждой вершине признак выбирается из случайного подмножества $\sqrt{n}$ признаков
- признаки и пороги выбираются по критерию Джини
- усечений (pruning) нет