

# Логические методы 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)»

МФТИ • 8 апреля 2021

## 1 Понятия закономерности и информативности

- Понятие закономерности
- Алгоритмы перебора правил для поиска закономерностей
- Критерии информативности

## 2 Решающие деревья

- Жадный метод обучения решающего дерева
- Усечение дерева (pruning)
- CART: деревья регрессии и классификации

## 3 Решающие списки и таблицы

- Решающие списки
- Решающие таблицы
- Бинаризация признаков

## 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  $R: X \rightarrow \{0, 1\}$ , удовлетворяющий двум требованиям:

1) интерпретируемость:

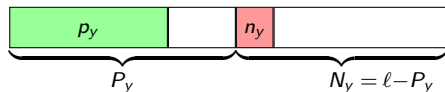
- 1)  $R$  записывается на естественном языке;
- 2)  $R$  зависит от небольшого числа признаков (1–7);

2) информативность относительно одного из классов  $y \in Y$ :

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

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

$$\frac{p_y(R)}{P_y} \gg \frac{n_y(R)}{N_y}$$



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%

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

## Обучение логических классификаторов

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

Четыре основных шага их построения:

- 1 Выбор семейства правил для поиска закономерностей
- 2 Выбор алгоритма порождения правил (rule generation)
- 3 Выбор критерия информативности (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»  $R(x)$ :

- высокоинформативный интерпретируемый признак
- одноклассовый классификатор с отказами

## Шаг 1. Часто используемые семейства правил

- Пороговое условие (решающий пень, decision stump):

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

- Конъюнкция пороговых условий:

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

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

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

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

## Шаг 1. Часто используемые семейства правил

- *Полуплоскость* — линейная пороговая функция:

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

- *Шар* — пороговая функция близости:

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

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

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

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

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

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

## Шаг 2. Мета-эвристики для поиска информативных правил

**Вход:** обучающая выборка  $X^\ell$ ;

**Выход:** множество закономерностей  $Z$ ;

инициализировать начальное множество правил  $Z$ ;

**повторять**

$Z' :=$  множество *локальных модификаций* правил из  $Z$ ;

удалить слишком похожие правила из  $Z \cup Z'$ ;

$Z :=$  наиболее *информативные* правила из  $Z \cup Z'$ ;

**пока** правила продолжают улучшаться;

**вернуть**  $Z$ ;

**Частные случаи:**

- 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:

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

*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  $J$

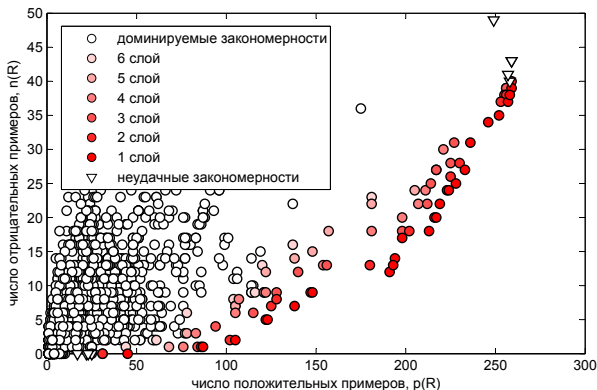
При удалении признака (pruning) информативность обычно оценивается по контрольной выборке (hold-out)

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

## Шаг 3. Двухкритериальный отбор закономерностей

Два критерия:  $p(R) \rightarrow \max$ ,  $n(R) \rightarrow \min$

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  $y \in Y$ :

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

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

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

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

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):

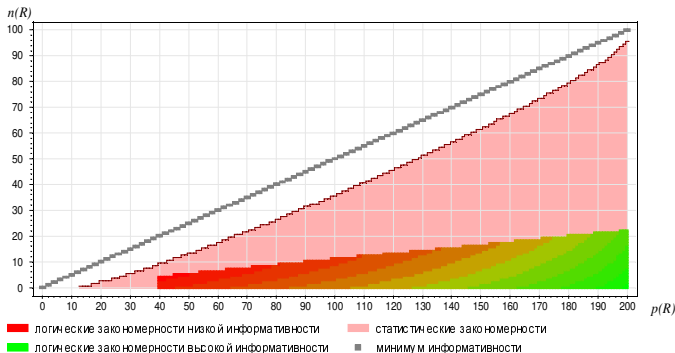
$$\text{IStat}(p, n) = -\frac{1}{\ell} \log_2 \frac{C_P^p C_N^n}{C_{P+N}^{p+n}} \rightarrow \max,$$

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  $\geq 0.9$ , Recall  $\geq 0.2$

Статистические закономерности: IStat  $\geq 3$



$P = 200$

$N = 100$

- статистический критерий удобнее для поиска правил
- логический критерий — для финального отбора правил

## Шаг 3. Зоопарк критериев информативности

### Очевидные, но не адекватные критерии:

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

### Адекватные, но не очевидные критерии:

- энтропийный критерий прироста информации:

$$\text{IGain}(p, n) = h\left(\frac{P}{\ell}\right) - \frac{p+n}{\ell} h\left(\frac{p}{p+n}\right) - \frac{\ell-p-n}{\ell} h\left(\frac{P-p}{\ell-p-n}\right) \rightarrow \max,$$

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

- критерий Джини (Gini impurity):

$$\text{IGini}(p, n) = \text{IGain}(p, n) \text{ при } h(q) = 4q(1 - q)$$

- критерий бустинга и его нормированный вариант:

$$\sqrt{p} - \sqrt{n} \rightarrow \max, \quad \sqrt{p/P} - \sqrt{n/N} \rightarrow \max$$

---

*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· $\ell$	IGain· $\ell$	$\sqrt{p} - \sqrt{n}$
50	0	<b>50</b>	50	0.25	50	22.65	23.70	7.07
100	50	<b>50</b>	-150	0	1.96	2.33	1.98	2.93
50	9	41	<b>5</b>	0.16	<b>5</b>	7.87	7.94	4.07
5	0	5	<b>5</b>	0.03	<b>5</b>	2.04	3.04	2.24
100	0	<b>100</b>	100	<b>0.5</b>	100	52.18	53.32	10.0
140	20	<b>120</b>	40	<b>0.5</b>	6.67	37.09	37.03	7.36

**Замечание.** Критерии IStat и IGain асимптотически эквивалентны:  $IStat(p, n) \rightarrow IGain(p, n)$  при  $\ell \rightarrow \infty$

## Шаг 4. Построение классификатора из закономерностей

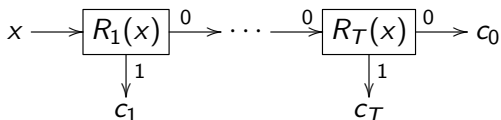
*Взвешенное голосование* (линейный классификатор с весами  $w_{yt}$  и, возможно, с регуляризацией для отбора признаков):

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

*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),  $c_0, c_1, \dots, c_T \in Y$ :



## 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  $V = V_{\text{внутр}} \sqcup V_{\text{лист}}$ ;

$f_v: X \rightarrow D_v$  — дискретный признак,  $\forall v \in V_{\text{внутр}}$ ;

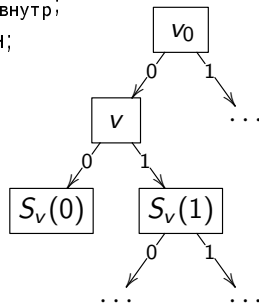
$S_v: D_v \rightarrow V$  — множество дочерних вершин;

$y_v \in Y$  — метка класса,  $\forall v \in V_{\text{лист}}$ ;

$v := v_0$ ;

**пока** ( $v \in V_{\text{внутр}}$ ):  $v := S_v(f_v(x))$ ;

**вернуть**  $a(x) = y_v$ ;



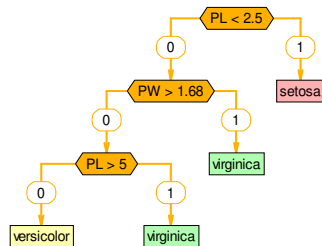
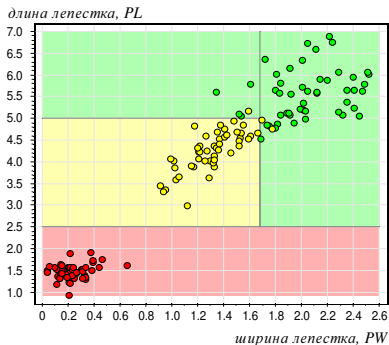
Чаще всего используются бинарные признаки вида  $f_v(x) = [f_j(x) \geq a_j]$

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



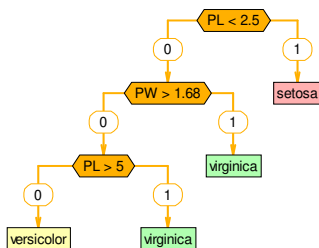
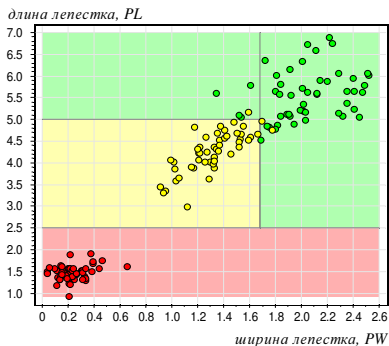
## Пример решающего дерева

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.

## Решающее дерево → покрывающий набор конъюнкций



setosa
virginica
virginica
versicolor

$$r_1(x) = [PL \leq 2.5]$$

$$r_2(x) = [PL > 2.5] \wedge [PW > 1.68]$$

$$r_3(x) = [PL > 5] \wedge [PW \leq 1.68]$$

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

## Обучение решающего дерева: ID3 (Iterative Dichotomiser)

$v_0 := \text{TreeGrowing}(X^\ell)$  — функция рекурсивно вызывает себя

$\text{TreeGrowing}(\text{Вход: } U \subseteq X^\ell) \mapsto \text{Выход:}$  корень дерева  $v$ ;

$f_v := \arg \max_{f \in F} \text{Gain}(f, U)$  — 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:  $\text{Major}(U) := \arg \max_{y \in Y} P(y|U)$ .

## Неопределённость распределения по классам в вершине

Частотная оценка вероятности класса  $y$  по выборке  $U$ :

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

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

$$\Phi\left(\begin{array}{|c|c|c|} \hline \text{■} & & \\ \hline \end{array}\right) < \Phi\left(\begin{array}{|c|c|c|} \hline \text{■} & \text{■} & \text{■} \\ \hline \end{array}\right) = \Phi\left(\begin{array}{|c|c|c|} \hline \text{■} & \text{■} & \text{■} \\ \hline \end{array}\right) < \Phi\left(\begin{array}{|c|c|c|} \hline \text{■} & \text{■} & \text{■} \\ \hline \end{array}\right)$$

- 1) минимальна и равна нулю, 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.

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

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

## Критерий ветвления

Неопределённость распределений  $P(y_i|U_k)$  после ветвления по признаку  $f_v$  и разбиения  $U$  на  $U_k = \{x \in U: f_v(x) = k\}$ :

$$\begin{aligned} \Phi(U_1, \dots, U_{|D_v|}) &= \frac{1}{|U|} \sum_{x_i \in U} \mathcal{L}(P(y_i|U_{f(x_i)})) = \\ &= \frac{1}{|U|} \sum_{k \in D_v} \sum_{x_i \in U_k} \mathcal{L}(P(y_i|U_k)) = \sum_{k \in D_v} \frac{|U_k|}{|U|} \Phi(U_k) \end{aligned}$$

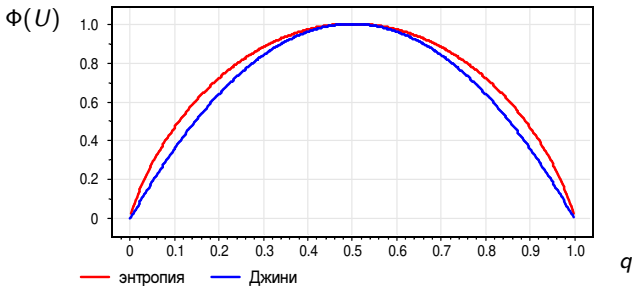
Выигрыш от ветвления вершины  $v$ :

$$\begin{aligned} \text{Gain}(f, U) &= \Phi(U) - \Phi(U_1, \dots, U_{|D_v|}) = \\ &= \Phi(U) - \sum_{k \in D_v} \frac{|U_k|}{|U|} \Phi(U_k) \rightarrow \max_{f \in F} \end{aligned}$$

## Критерий Джини и энтропийный критерий

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

- Если  $\mathcal{L}(p) = -\log_2 p$ , то  
 $\Phi(U) = -q \log_2 q - (1-q) \log_2(1-q)$  — энтропия выборки.
- Если  $\mathcal{L}(p) = 2(1-p)$ , 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,  $v \in V_{\text{внутр}}$
- $P(y|x, v) = \frac{1}{|U|} \sum_{x_i \in U} [y_i = y]$  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:

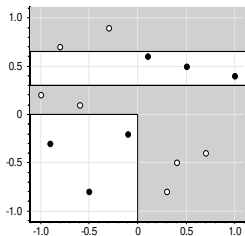
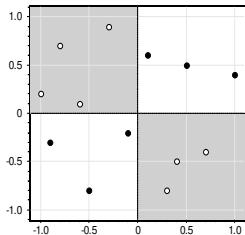
- Интерпретируемость и простота классификации.
- Правила  $[f_j(x) < \alpha]$  не требуют масштабирования признаков.
- Допустимы разнотипные данные и данные с пропусками.
- Трудоёмкость линейна по длине выборки  $O(|F|h\ell)$ .
- Не бывает отказов от классификации.

### Недостатки:

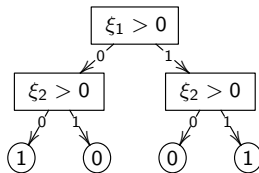
- Жадная стратегия переусложняет структуру дерева, и, как следствие, сильно переобучается.
- Фрагментация выборки: чем дальше  $v$  от корня, тем меньше статистическая надёжность выбора  $f_v, y_v$ .
- Высокая чувствительность к шуму, к составу выборки, к критерию информативности.



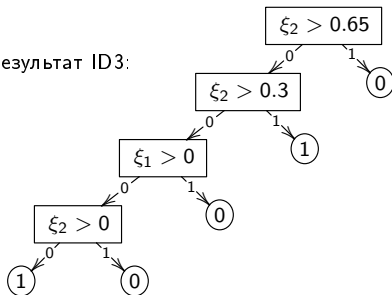
## Жадная стратегия переусложняет структуру дерева



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  $X_v^q = \emptyset$  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:  $Y = \mathbb{R}$ ,  $y_v \in \mathbb{R}$ ,

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

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

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

Значение  $y_v$  в терминальной вершине  $v$  — МНК-решение:

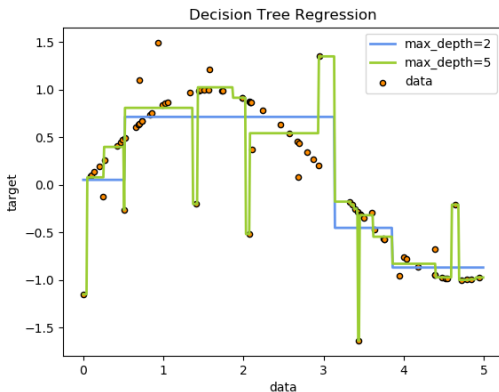
$$y_v = \frac{1}{|U|} \sum_{x_i \in U} y_i$$

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](http://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html)

## CART: критерий Minimal Cost-Complexity Pruning

Среднеквадратичная ошибка со штрафом за сложность дерева:

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

При увеличении  $\alpha$  дерево последовательно упрощается.

Причём последовательность вложенных деревьев единственна.

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

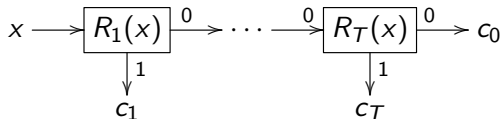
Для случая классификации используется аналогичная стратегия усечения, с критерием Джини.

---

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

## Определение решающего списка (Decision List, DL)

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



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

для всех  $t = 1, \dots, T$

└ если  $R_t(x) = 1$  то вернуть  $c_t$ ;

вернуть  $c_0$  (отказ от классификации объекта  $x$ );

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

## Жадный алгоритм построения решающего списка

**Вход:** выборка  $X^\ell$ ; параметры:  $T_{\max}$ ,  $I_{\min}$ ,  $E_{\max}$ ,  $\ell_0$ ;

**Выход:** решающий список  $\{R_t, c_t\}_{t=1}^T$ ;

$U := X^\ell$ ;

**для всех**  $t := 1, \dots, T_{\max}$

выбрать класс  $c_t$ ;

поиск правила  $R_t$  по максимуму информативности:

$R_t := \arg \max_R I(R, U)$  при ограничении  $E(R, U) \leq E_{\max}$ ;

**если**  $I(R_t, U) < I_{\min}$  **то выход**;

$U := \{x \in U : R_t(x) = 0\}$  — не покрытые правилом  $R_t$ ;

**если**  $|U| \leq \ell_0$  **то выход**;

## Замечания к алгоритму построения решающего списка

- **Стратегии выбора класса  $c_t$ :**
  - 1) все классы по очереди
  - 2) на каждом шаге определяется оптимальный класс
- Параметр  $E_{\max}$  управляет сложностью списка:  
 $E_{\max} \downarrow \Rightarrow p(R_t) \downarrow, T \uparrow$
- **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)



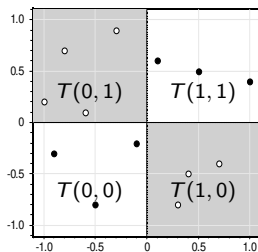
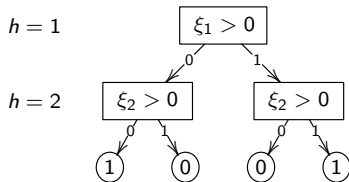
## Небрежные решающие деревья (Oblivious Decision Tree, ODT)

**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*  $T: \{0, 1\}^H \rightarrow Y$ :

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

**Пример:** задача XOR,  $H = 2$ .



## Алгоритм обучения ODT

**Вход:** выборка  $X^\ell$ ; множество признаков  $F$ ; глубина дерева  $H$ ;

**Выход:** признаки  $f_h$ ,  $h = 1, \dots, H$ ; таблица  $T: \{0, 1\}^H \rightarrow Y$ ;

для всех  $h = 1, \dots, H$

    предикат с максимальным выигрышем определённости:

$$f_h := \arg \max_{f \in F} \text{Gain}(f_1, \dots, f_{h-1}, f);$$

классификация по мажоритарному правилу:

$$T(\beta) := \text{Major}(U_H\beta);$$

Выигрыш от ветвления на уровне  $h$  по всей выборке  $X^\ell$ :

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

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

## Вспомогательная задача бинаризации вещественного признака

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

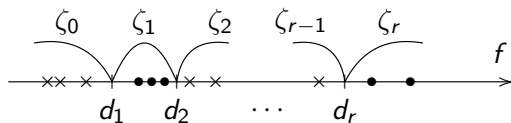
**Дано:** выборка значений вещественного признака  $f(x_i)$ ,  $x_i \in X^\ell$ .

**Найти:** наилучшее (в каком-то смысле) разбиение области значений признака на относительно небольшое число зон:

$$\zeta_0(x) = [f(x) < d_1];$$

$$\zeta_s(x) = [d_s \leq f(x) < d_{s+1}], \quad s = 1, \dots, r-1;$$

$$\zeta_r(x) = [d_r \leq f(x)].$$



## Способы разбиения области значений признака на зоны

- 1 Жадная максимизация информативности путём слияний
- 2 Разбиение на равномошные подвыборки
- 3 Разбиение по равномерной сетке «удобных» значений
- 4 Объединение нескольких разбиений

### Повышение «удобства» пороговых значений

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

Если таких  $x^*$  несколько, выбрать

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

## Алгоритм разбиения области значений признака на зоны

**Вход:** выборка  $X^\ell$ ; класс  $c \in Y$ ; параметры  $r$  и  $\delta_0$ ;

**Выход:**  $D = \{d_1 < \dots < d_r\}$  — последовательность порогов;

$D := \emptyset$ ; упорядочить выборку  $X^\ell$  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  $\zeta_{i-1}, \zeta_i, \zeta_{i+1}$ , удалив  $d_i$  и  $d_{i+1}$  из  $D$ ;

**пока**  $|D| > r + 1$ ;

- Основные требования к логическим закономерностям:
  - интерпретируемость, информативность, различность.
- Преимущества решающих деревьев:
  - интерпретируемость,
  - допускаются разнотипные данные,
  - возможность обхода пропусков.
- Недостатки решающих деревьев:
  - переобучение,
  - чувствительность к шумам, составу выборки, критерию.
- Способы устранения этих недостатков:
  - редукция,
  - композиции (леса) деревьев (в следующей лекции).
- Упрощённые варианты решающих деревьев:
  - решающие списки — для вывода знаний из данных;
  - ODT — для построения композиций (Yandex MatrixNet).