

Логические алгоритмы 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)»

Видеолекции: <http://shad.yandex.ru/lectures>

1 Понятия закономерности и информативности

- Понятие закономерности
- Поиск и отбор закономерностей
- Критерии информативности

2 Решающие деревья

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

3 Решающие списки, таблицы и леса

- Решающие леса
- Решающие таблицы
- Решающие списки

Логическая закономерность

$X^\ell = (x_i, y_i)_{i=1}^\ell \subset X \times Y$ — обучающая выборка, $y_i = y(x_i)$.

Логическая закономерность (правило, rule) — это предикат $R: X \rightarrow \{0, 1\}$, удовлетворяющий двум требованиям:

1) интерпретируемость:

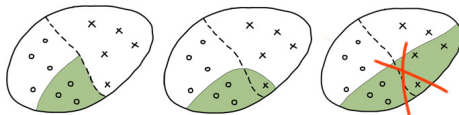
- 1) R записывается на естественном языке;
- 2) R зависит от небольшого числа признаков (1–7);

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

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

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

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%.*

Обучение логических классификаторов

Основные шаги *индукции правил* (rule induction):

- 1 Выбор семейства правил для поиска закономерностей
- 2 Порождение правил (rule generation)
- 3 Отбор правил-закономерностей (rule selection)
- 4 Построение классификатора из правил как из признаков, пример: *взвешенное голосование* (weighted voting) правил

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

Двойственная трактовка понятия «закономерности» $R(x)$:

- высокоинформативный интерпретируемый признак
- одноклассовый классификатор с отказами

Часто используемые виды закономерностей

1. Пороговое условие (решающий пень, decision stump):

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

2. Конъюнкция пороговых условий:

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

3. 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.

Часто используемые виды закономерностей

4. *Полуплоскость* — линейная пороговая функция:

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

5. *Шар* — пороговая функция близости:

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

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 критерия информативности.

Мета-эвристики для поиска информативных закономерностей

Вход: обучающая выборка X^{ℓ} ;

Выход: множество закономерностей Z ;

- 1: начальное множество правил Z ;
- 2: **повторять**
- 3: $Z' :=$ множество *локальных модификаций* правил $R \in Z$;
- 4: удалить слишком похожие правила из $Z \cup Z'$;
- 5: $Z :=$ наиболее *информативные* правила из $Z \cup Z'$;
- 6: **пока** правила продолжают улучшаться
- 7: **вернуть** Z .

Частные случаи:

- стохастический локальный поиск (stochastic local search)
- генетические (эволюционные) алгоритмы
- поиск в ширину
- поиск в глубину (метод ветвей и границ)

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:

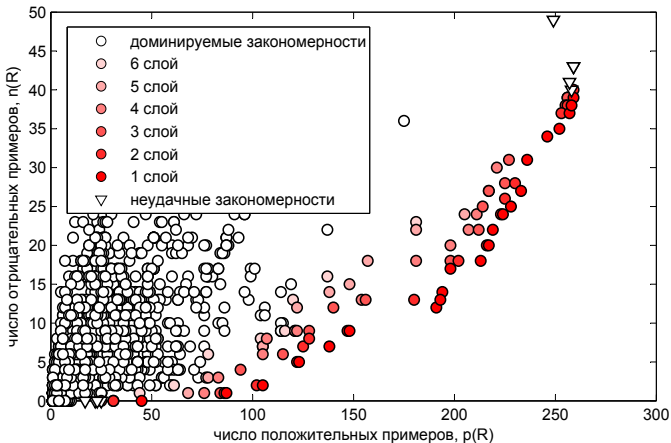
- варьирование одного из порогов a_j и b_j
- варьирование обоих порогов a_j , b_j одновременно
- добавление признака f_j в J с варьированием порогов a_j , b_j
- удаление признака f_j из J

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)

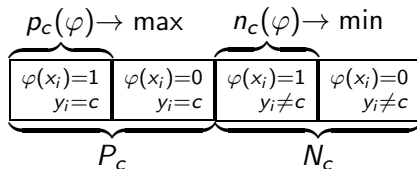
Отбор закономерностей по паре критериев $p \rightarrow \max$, $n \rightarrow \min$

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



UCI:german

Отбор закономерностей по паре пороговых критериев



Определение

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

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

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

Если $n_c(\varphi) = 0$, то φ — непротиворечивая закономерность.

Проблема: хотелось бы иметь один скалярный критерий.

Отбор закономерностей по критерию информативности

Проблема: хотелось бы иметь один скалярный критерий:

$$\begin{cases} p(R) \rightarrow \max \\ n(R) \rightarrow \min \end{cases} \xRightarrow{?} I(p, n) \rightarrow \max$$

Очевидные, но не всегда адекватные свёртки:

- $I(p, n) = \frac{p}{p+n} \rightarrow \max$ (precision);
- $I(p, n) = p - n \rightarrow \max$ (accuracy);
- $I(p, n) = p - Cn \rightarrow \max$ (linear cost accuracy);
- $I(p, n) = p/P - n/N \rightarrow \max$ (relative accuracy);
где $P = \#\{x_i: y_i=c\}$, $N = \#\{x_i: y_i \neq c\}$.

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

Нетривиальность проблемы свёртки двух критериев

Пример:

при $P = 200$, $N = 100$ и различных p и n .

Простые эвристики не всегда адекватны:

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

Часто используемые критерии информативности

Более адекватные, но менее очевидные свёртки:

- энтропийный критерий прироста информации:

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

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

- критерий Джини (Gini impurity):

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

- точный статистический тест Фишера (Fisher's Exact Test):

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

- критерий бустинга:

$$\sqrt{p} - \sqrt{n} \rightarrow \max$$

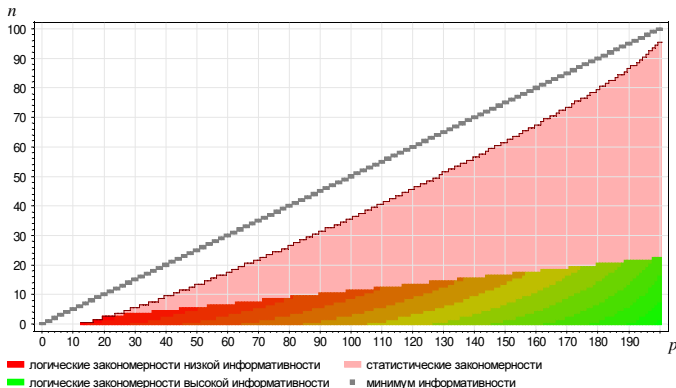
- нормированный критерий бустинга:

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

Где находятся закономерности в (p, n) -плоскости

Логические закономерности: $\frac{n}{p+n} \leq 0.1$, $\frac{p}{P+N} \geq 0.05$.

Статистические закономерности: $IStat(p, n) \geq 3$.



Вывод: неслучайность — ещё не значит закономерность.

Композиции закономерностей

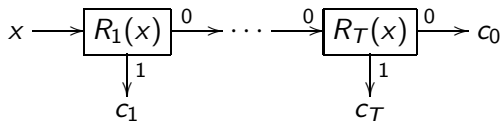
Взвешенное голосование (линейный классификатор с весами w_{yt}):

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

Простое голосование (комитет большинства)

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

Решающий список (комитет старшинства), $c_t \in Y$:



Определение решающего дерева (Decision Tree)

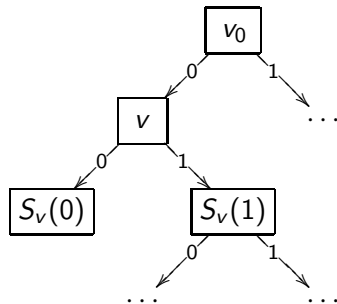
Решающее дерево — алгоритм классификации $a(x)$, задающийся *деревом* (связным ациклическим графом):

- 1) $V = V_{\text{внутр}} \sqcup V_{\text{лист}}$, $v_0 \in V$ — корень дерева;
- 2) $v \in V_{\text{внутр}}$: функции $f_v: X \rightarrow D_v$ и $S_v: D_v \rightarrow V$, $|D_v| < \infty$;
- 3) $v \in V_{\text{лист}}$: метка класса $y_v \in Y$.

- 1: $v := v_0$;
- 2: **пока** $v \in V_{\text{внутр}}$
- 3: $v := S_v(f_v(x))$;
- 4: **вернуть** y_v ;

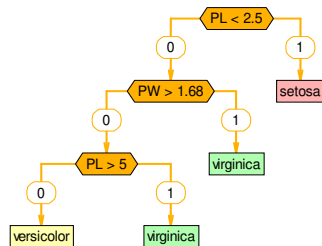
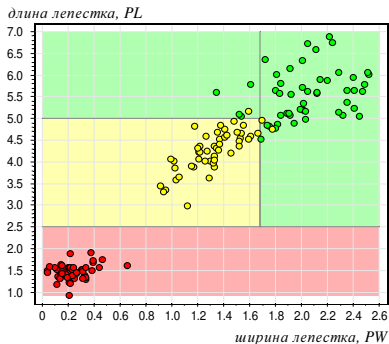
Частный случай: $D_v \equiv \{0, 1\}$
— бинарное решающее дерево

Пример: $f_v(x) = [f_j(x) \geq a_j]$



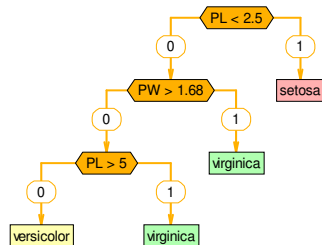
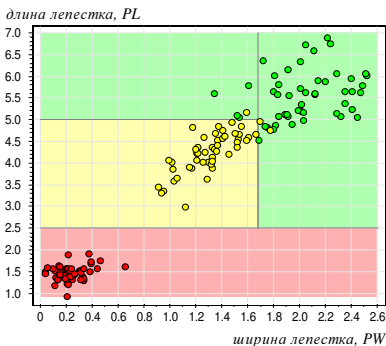
Пример решающего дерева

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

Обучение решающего дерева: стратегия «разделяй и властвуй»

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

- 1: **ФУНКЦИЯ** $\text{TreeGrowing}(U \subseteq X^\ell) \mapsto$ корень дерева v ;
- 2: **если** $\text{StopCriterion}(U)$ **то**
- 3: **вернуть** новый лист v , взяв $y_v := \text{Major}(U)$;
- 4: найти признак, наиболее выгодный для ветвления дерева:
 $f_v := \arg \max_{f \in F} \text{Gain}(f, U)$;
- 5: **если** $\text{Gain}(f_v, U) < G_0$ **то**
- 6: **вернуть** новый лист v , взяв $y_v := \text{Major}(U)$;
- 7: создать новую внутреннюю вершину v с функцией f_v ;
- 8: **для всех** $k \in D_v$
 $U_k := \{x \in U : f_v(x) = k\}, S_v(k) := \text{TreeGrowing}(U_k)$,
- 9: **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 в вершине $v \in V_{\text{внутр}}$:

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

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

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

- 1) минимальна, 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.

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

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 вершины v по признаку f и разбиения $U = \bigsqcup_{k \in D_v} U_k$:

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

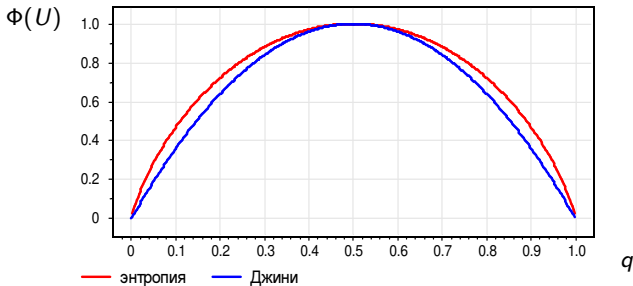
Выигрыш от ветвления вершины v :

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

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:

- $f_v(x)$ определено \Rightarrow из дочерней $s = S_v(f_v(x))$ взять $P(y|x, v) = P(y|x, s)$.
- $f_v(x)$ не определено \Rightarrow пропорциональное распределение:
$$P(y|x, v) = \sum_{k \in D_v} q_{vk} P(y|x, S_v(k)).$$
- Окончательное решение — наиболее вероятный класс:
$$a(x) = \arg \max_{y \in Y} P(y|x, v_0).$$

Жадная нисходящая стратегия: достоинства и недостатки

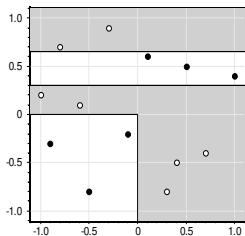
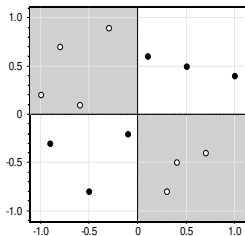
Достоинства:

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

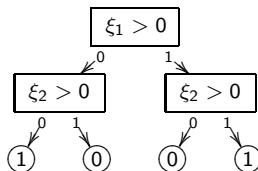
Недостатки:

- 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.

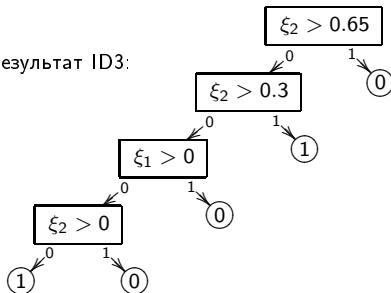
Жадная стратегия переусложняет структуру дерева



Оптимальное дерево для задачи XOR:



Результат ID3:



Усечение дерева (pruning)

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

- 1: **для всех** $v \in V_{\text{внутр}}$
- 2: $X_v^q :=$ подмножество объектов X^q , дошедших до v ;
- 3: **если** $X_v^q = \emptyset$ **то**
- 4: **вернуть** новый лист v , $y_v := \text{Major}(U)$;
- 5: число ошибок при классификации X_v^q разными способами:
 $\text{Err}(v)$ — поддеревом, растущим из вершины v ;
 $\text{Err}_k(v)$ — дочерним поддеревом $S_v(k)$, $k \in D_v$;
 $\text{Err}_c(v)$ — классом $c \in Y$.
- 6: в зависимости от того, какое из них минимально:
 сохранить поддерево v ;
 заменить поддерево v дочерним $S_v(k)$;
 заменить поддерево v листом, $y_v := \arg \min_{c \in Y} \text{Err}_c(v)$.

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

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

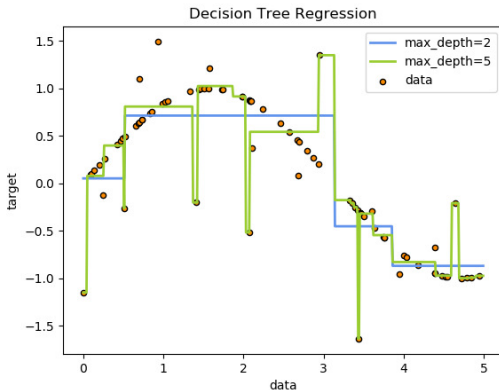
Значение y_v в терминальной вершине v — МНК-решение:

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

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

Пример. Деревья регрессии различной глубины

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



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 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.

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 возвращениями ($1 - 1/e \approx 63.2\%$ объектов)
- в каждой вершине признак выбирается из случайного подмножества \sqrt{n} признаков ($\lfloor n/3 \rfloor$ для регрессии)
- признаки и пороги выбираются по критерию Джини
- усечений (pruning) нет

Разновидности решающих лесов

- Случайный лес (Random Forest)
- Использование большого числа простых решающих деревьев в качестве признаков, в любом классификаторе.
- Oblique Random Forest, Rotation Forest
 $f_V(x)$ — линейные комбинации признаков, выбираемые по энтропийному критерию информативности.
- Решающий список из решающих деревьев:
 - при образовании статистически ненадёжного листа этот лист заменяется переходом к следующему дереву;
 - следующее дерево строится по объединению подвыборок, прошедших через ненадёжные листы предыдущего дерева.

Небрежные решающие деревья (Oblivious Decision Tree, ODT)

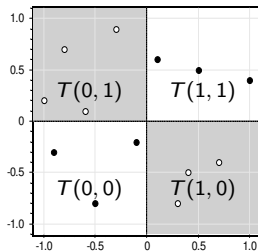
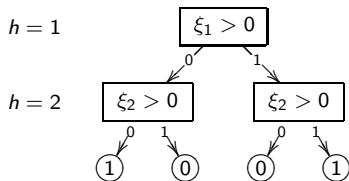
Решение проблемы фрагментации в деревьях:

строится сбалансированное 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^ℓ ; множество признаков F ; глубина дерева H ;

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

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

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

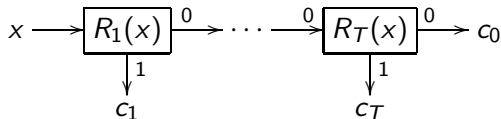
$$\text{Gain}(f_1, \dots, f_h) = \Phi(X^\ell) - \sum_{\beta \in \{0, 1\}^h} \frac{|U_{h\beta}|}{\ell} \Phi(U_{h\beta}),$$

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

Определение решающего списка

Решающий список (Decision List, DL)

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



- 1: **для всех** $t = 1, \dots, T$
- 2: **если** $R_t(x) = 1$ **то**
- 3: **вернуть** c_t ;
- 4: **вернуть** c_0 — отказ от классификации объекта x .

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

Жадный алгоритм построения решающего списка

Вход: выборка X^ℓ ; семейство правил \mathcal{R} ;

параметры: T_{\max} , I_{\min} , E_{\max} , ℓ_0 ;

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

- 1: $U := X^\ell$;
- 2: **для всех** $t := 1, \dots, T_{\max}$
- 3: выбрать класс c_t ;
- 4: максимизация информативности $I(R, U)$ при ограничении на число ошибок $E(R, U)$:
$$R_t := \arg \max_{R \in \mathcal{R}: E(R, U) \leq E_{\max}} I(R, U);$$
- 5: **если** $I(R_t, U) < I_{\min}$ **то выход**;
- 6: оставить объекты, не покрытые правилом R_t :
$$U := \{x \in U : R_t(x) = 0\};$$
- 7: **если** $|U| \leq \ell_0$ **то выход**;

Замечания к алгоритму построения решающего списка

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

Вспомогательная задача бинаризации вещественного признака

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

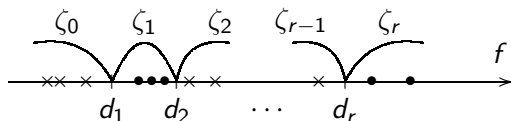
Дано: выборка значений вещественного признака $f(x_i)$, $x_i \in X^\ell$.

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^ℓ ; класс $c \in Y$; параметры r и δ_0 .

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

-
- 1: $D := \emptyset$; упорядочить выборку X^ℓ по возрастанию $f(x_i)$;
 - 2: **для всех** $i = 2, \dots, \ell$
 - 3: **если** $f(x_{i-1}) \neq f(x_i)$ и $[y_{i-1} = c] \neq [y_i = c]$ **то**
 - 4: добавить порог $\frac{1}{2}(f(x_{i-1}) + f(x_i))$ в конец D ;
 - 5: **повторять**
 - 6: **для всех** $d_j \in D, j = 1, \dots, |D| - 1$
 - 7: $\delta I_j := I(\zeta_{i-1} \vee \zeta_i \vee \zeta_{i+1}) - \max\{I(\zeta_{i-1}), I(\zeta_i), I(\zeta_{i+1})\}$;
 - 8: $i := \arg \max_s \delta I_s$;
 - 9: **если** $\delta I_i > \delta_0$ **то**
 - 10: слить зоны $\zeta_{i-1}, \zeta_i, \zeta_{i+1}$, удалив d_j и d_{j+1} из D ;
 - 11: **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,
 - фрагментация,
 - неустойчивость к шуму, составу выборки, критерию;
- Способы устранения этих недостатков:
 - редукция,
 - композиции (леса) деревьев.

Yandex MatrixNet = голосование (градиентный бустинг) над ODT.