Методы машинного обучения, основанные на индукции правил (логические методы классификации)

К. В. Воронцов
(vokov@forecsys.ru, http://www.ccas.ru/voron)

Вычислительный Центр им. А. А. Дородницына РАН

Knowledge and Ontology *ELSEWHERE*
(*ELSEWHERE*2009)
Москва, 27 июля 2009

Задача классификации (определения и обозначения)

- **Дано:** $X^\ell = (x_i, y_i)_{i=1}^\ell$ — *обучающая выборка*,
  - x_i — объекты из X, описываемые n признаками: $x_i \mapsto \big(f_1(x_i), \ldots, f_n(x_i)\big)$,
  - $y_i = y(x_i)$ — класс объекта x_i, $y_i \in Y = \{1, \ldots, m\}$.
- **Построить:**
  - классификатор $a\colon X \to Y$, аппроксимирующий неизвестную «истинную» функцию $y(x)$, $\forall x \in X$.
- **Примеры прикладных задач:**
  - медицинская диагностика и прогнозирование;
  - кредитный скоринг (предсказание дефолта заёмщика);
  - предсказание ухода абонента от оператора сотовой связи;
  - фильтрация спама;
  - категоризация текстовых документов;
  - распознавание терминов в тексте.

Методы классификации

- Статистические (Generative models):
  - (Fisher's Discriminant, Logistic Regression, Naive Bayes, etc.)
- Разделяющие (Discriminative models):
  - (SVM, RVM, Kernel methods, etc.)
- Метрические (Similarity-based models):
  - (kNN, RBF, Parzen window, etc.)
- Нейросетевые (Neural nets):
  - (Back-Propagation, OBD, etc.)
- Логические (Rule induction):
  - взвешенное голосование (SLIPPER, etc.);
  - решающие списки, деревья, леса (DL, DT, DF);
  - взвешенные деревья (ODT, ALT–TreeNet);
- Композиционные (Compositions):
  - (Boosting, Bagging, RSM, Mixture of experts, etc.)

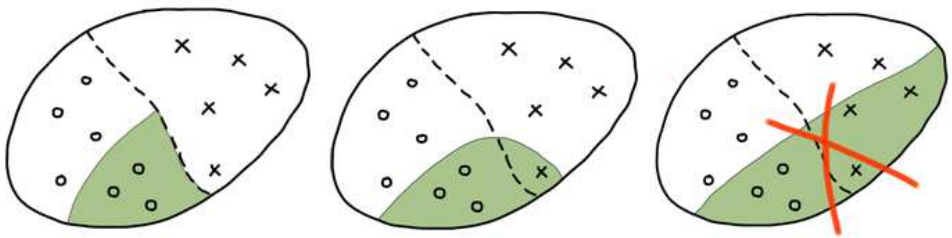
Логические методы классификации

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

1) *интерпретируемость*:
   φ записывается на естественном языке;
   φ зависит от небольшого числа признаков (1–7);

2) *информативность* относительно одного из классов $c \in Y$:
   $p_c(\varphi) = \#\{x_i\colon \varphi(x_i) = 1 \text{ и } y_i = c\} \to \max$;
   $n_c(\varphi) = \#\{x_i\colon \varphi(x_i) = 1 \text{ и } y_i \neq c\} \to \min$;

Если $\varphi(x) = 1$, то говорят «φ покрывает x» (φ covers x).



Содержание

Требование интерпретируемости

Пример (spam detection)

Если *текст содержит «Иностранные работники»*
и *искажённый телефонный номер,*
то *это спам с вероятностью 99%.*

Пример (credit scoring)

Если *в анкете указан домашний телефон*
и *зарплата* $>$ *\$2000* ***и*** *сумма кредита* $<$ *\$5000*
то *кредит можно выдать, риск дефолта 5%.*

Требования интерпретируемости:

1) φ зависит от малого числа признаков;
2) формула φ выражается на естественном языке.

Виды закономерностей

Параметрическое семейство *конъюнкций пороговых термов*:

$$\varphi(x) = \bigwedge_{j \in J} \left[\alpha_j \leqslant f_j(x) \leqslant \beta_j\right].$$

Параметрическое семейство *полуплоскостей*:

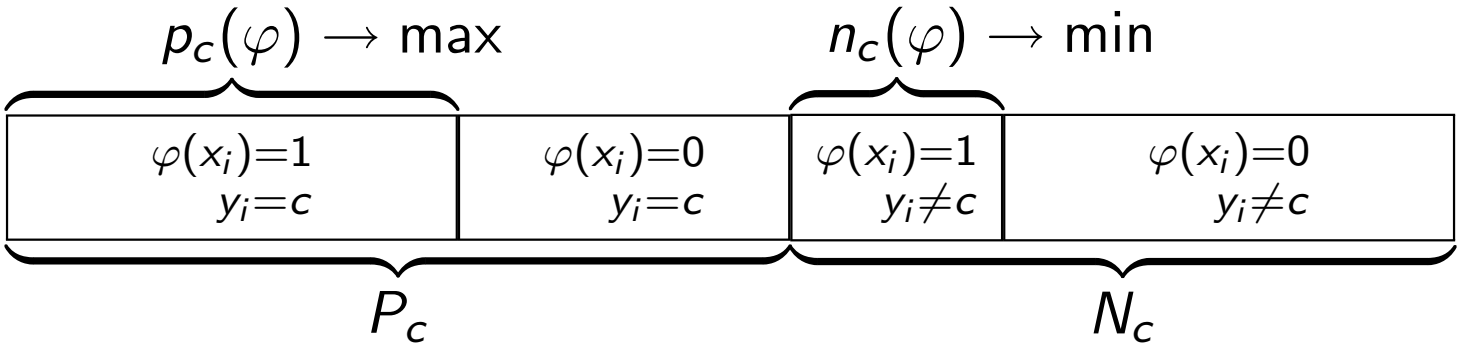
$$\varphi(x) = \left[\sum_{j \in J} \alpha_j f_j(x) \geqslant \alpha_0\right].$$

Параметрическое семейство *шаров* (Алгоритмы вычисления оценок [Ю.И.Журавлёв], DLM, SCM [Marchand]):

$$\varphi(x) = \left[R(x, x_0) \leqslant R_0\right], \quad R(x, x_0) = \sum_{j \in J} \alpha_j \left|f_j(x) - f_j(x_0)\right|^{\gamma}.$$

Основная проблема — отбор признаков $J \subseteq \{1, \ldots, n\}$.

Логический (эвристический) критерий закономерности



Определение

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

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

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

Проблема: хотелось бы иметь один скалярный критерий.

Нетривиальность проблемы свёртки двух критериев

Претенденты на звание «Критерий информативности» при $P = 200$, $N = 100$ и различных p и n.

p	n	$p - n$	$p - 5n$	$\frac{p}{P} - \frac{n}{N}$	$\frac{p}{n+1}$	I_c	IGain_c	$\sqrt{p} - \sqrt{n}$
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

Вывод:

придумать хороший критерий информативности не так просто!

Статистический критерий информативности

Пусть X — в.п., выборка X^ℓ — i.i.d.
Гипотеза H_0: $y(x)$ и $\varphi(x)$ — независимые случайные величины.
Тогда вероятность реализации пары (p, n) описывается функцией гипергеометрического распределения:

$$h(p,n) = \frac{C_P^p C_N^n}{C_{P+N}^{p+n}}, \quad 0 \leqslant p \leqslant P, \quad 0 \leqslant n \leqslant N,$$

где $C_N^n = \frac{N!}{n!(N-n)!}$ — биномиальные коэффициенты.

Определение

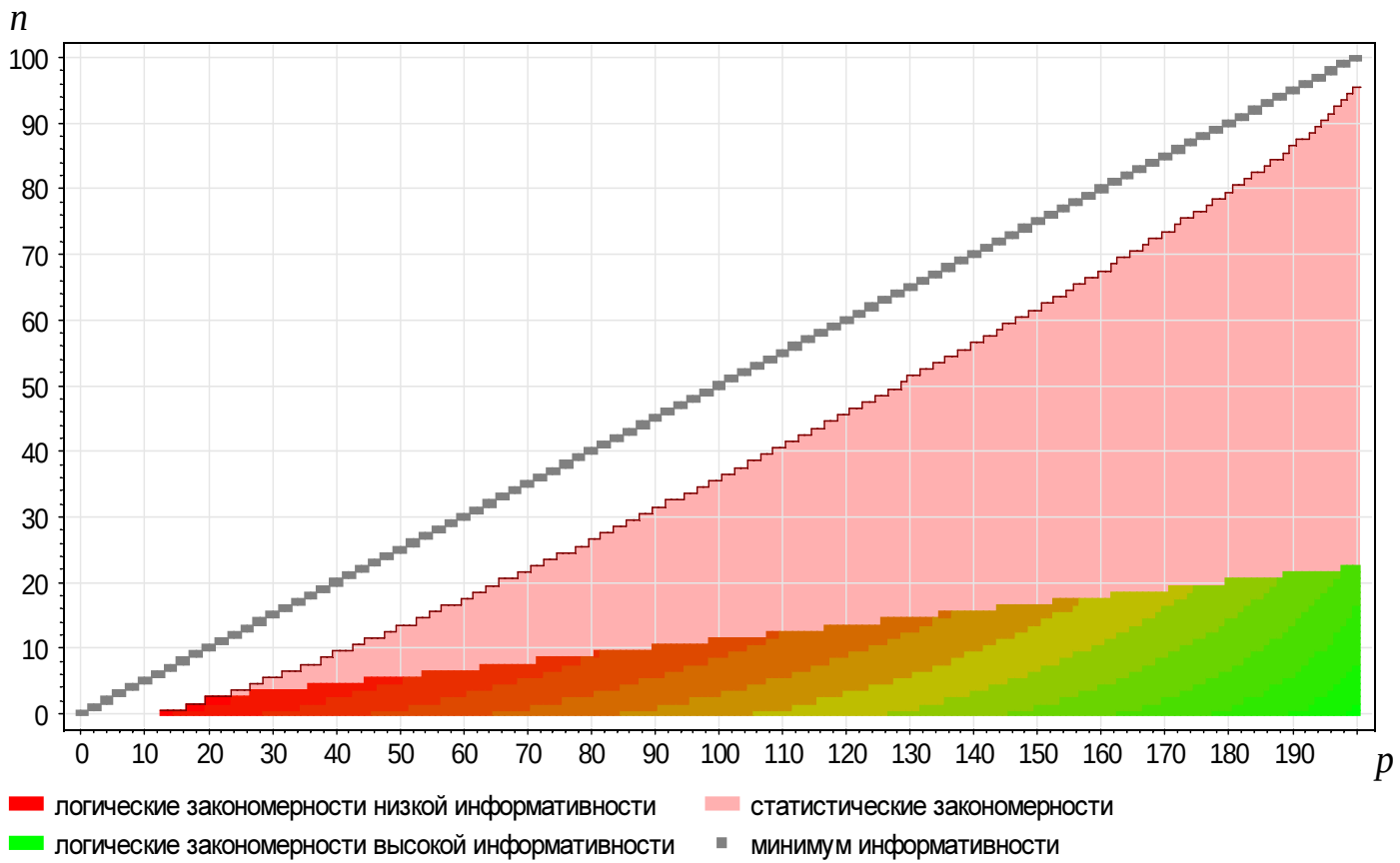
Информативность предиката $\varphi(x)$ относительно класса $c \in Y$:

$$I_c(\varphi, X^\ell) = -\ln h(p_c(\varphi), n_c(\varphi)).$$

$I_c(\varphi, X^\ell) \geqslant I_0$ *— статистическая закономерность класса $c \in Y$.*

Соотношение логического и статистического критериев

Области логических ε, δ-закономерностей (при $\varepsilon = 0.1$)
и статистических закономерностей (при $l_0 = 5$)
в координатах (p, n) при $P = 200$, $N = 100$.



Энтропийный критерий информативности

Пусть ω_0, ω_1 — два исхода с вероятностями q и $1-q$.
Количество информации: $I_0 = -\log_2 q$, $I_1 = -\log_2(1-q)$.
Энтропия — матожидание количества информации:

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

Энтропия выборки X^ℓ, если исходов два — «$y{=}c$» и «$y{\neq}c$»:

$$H(y) = h\left(\frac{P}{\ell}\right).$$

Энтропия выборки X^ℓ после получения информации φ:

$$H(y|\varphi) = \frac{p+n}{\ell} h\left(\frac{p}{p+n}\right) + \frac{\ell-p-n}{\ell} h\left(\frac{P-p}{\ell-p-n}\right).$$

Информационный выигрыш (Information gain, IGain):

$$\mathrm{IGain}_c(\varphi, X^\ell) = H(y) - H(y|\varphi).$$

Соотношение статистического и энтропийного критериев

Определение

Предикат φ — закономерность по энтропийному критерию, если $\mathrm{IGain}_c(\varphi, X^\ell) > G_0$ *при некотором* G_0.

Теорема

Энтропийный критерий IGain_c *асимптотически эквивалентен статистическому* I_c:

$$\mathrm{IGain}_c(\varphi, X^\ell) \to \frac{1}{\ell \ln 2} I_c(\varphi, X^\ell) \quad \text{при } \ell \to \infty.$$

Доказательство:
применить формулу Стирлинга к статистическому критерию.

Выводы и практические рекомендации

- Статистический критерий лучше использовать на этапе синтеза правил, для оценивания их перспективности.
- Логический ε, δ-критерий лучше использовать на этапе финального отбора правил.
- Энтропийный критерий можно использовать только при достаточно больших ℓ, p, n.

- Философский вывод:
«неслучайность — это ещё не значит закономерность».

Задача перебора конъюнкций

Пусть $\mathscr{B}$ — конечное множество *элементарных предикатов*.
Множество конъюнкций с небольшим числом термов из $\mathscr{B}$:

$$\varphi(x) = \beta_1(x) \wedge \cdots \wedge \beta_k(x), \quad \beta_1, \ldots, \beta_k \in \mathscr{B}, \quad k \leqslant K.$$

Число допустимых конъюнкций огромно: $O(|\mathscr{B}|^K)$.

Семейство методов локального поиска

Окрестность $V(\varphi)$ — все конъюнкции, получаемые из $V(\varphi)$ добавлением, изъятием или модификацией одного из термов.

Основная идея: на t-й итерации взять лучшую конъюнкцию из окрестности:

$$\varphi_t := \operatorname*{arg\,max}_{\varphi \in V(\varphi_{t-1})} I_c(\varphi, X^\ell).$$

Алгоритм локального поиска

Вход: выборка X^ℓ; класс $c \in Y$;
начальное приближение φ_0; параметры $t_{\max}$, d, ε;
Выход: конъюнкция φ;

1: $I^* := I_c(\varphi_0, X^\ell)$; $\varphi^* := \varphi_0$;
2: **для всех** $t = 1, \ldots, t_{\max}$
3: поиск лучшей конъюнкции в окрестности φ_{t-1}:
$\varphi_t^* := \arg\max I_c(\varphi, X^\ell)$ по всем $\varphi \in V(\varphi_{t-1})$: $E_c(\varphi) < \varepsilon$;
4: **если** $I_c(\varphi_t^*) > I^*$ **то**
$t^* := t$; $\varphi^* := \varphi_t^*$; $I^* := I_c(\varphi^*)$
5: **если** $t - t^* > d$ **то**
6: **выход**;
7: поиск наиболее перспективной конъюнкции:
$\varphi_t := \arg\max I_c(\varphi, X^\ell)$ по всем $\varphi \in V(\varphi_{t-1})$;
8: **вернуть** φ^*;

Частные случаи

- **жадный алгоритм:**
  $V(\varphi)$ — только добавления термов; $\varphi_0 = \varnothing$;
- **стохастический локальный поиск (SLS):**
  $V(\varphi)$ — случайное подмножество всевозможных добавлений, удалений, модификаций термов; $\varphi_0 = \varnothing$;
- **стабилизация:**
  $V(\varphi)$ — удаления термов или изменение параметров в термах; $\varphi_0 \neq \varnothing$;
  (рекомендуется для финальной настройки порогов в термах)
- **редукция:**
  $V(\varphi)$ — только удаления термов; $\varphi_0 \neq \varnothing$;
  $I_c(\varphi, X^k)$ оценивается по контрольной выборке X^k;
  (для повышения обобщающей способности правила)

Обобщение алгоритма локального поиска

- Генетические алгоритмы. Основные отличия от SLS:
  — строится не одна конъюнкция φ_t, а много (поколение);
  — наряду со случайными модификациями (мутациями) конъюнкции обмениваются термами (скрещивания).
- Другие эвристические стратегии поиска:
  — поиск в глубину, алгоритм «Кора» [Бонгард, 1961, Вайнцвайг, 1973];
  — поиск в ширину, алгоритм «ТЭМП» [Лбов, Загоруйко, 1976];
  — случайный поиск с адаптацией, СПА [Лбов, 1985].

Диверсификация закономерностей

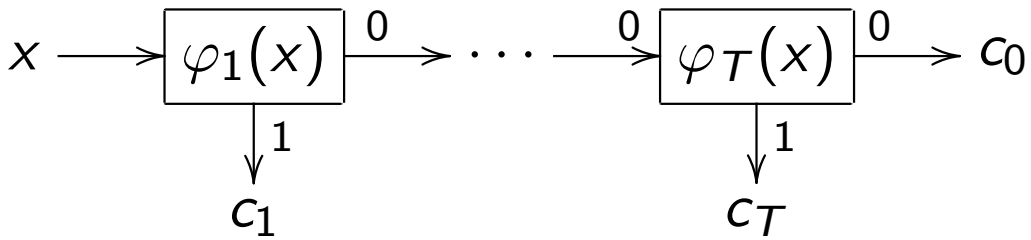
Как построить много хороших, но разных, закономерностей?

- кооперативная коэволюция (cooperative coevilution):
  — правило оценивается не по его индивидуальной информативности, а по качеству классификатора, в котором оно используется.
- поиск правил на больших данных, алгоритм DMEL
  — [*Wai-Ho Au, et al.* A novel evolutionary data mining algorithm with applications to churn prediction // IEEE Trans. Evolutionary Computation, 2003., Vol. 7, no. 6., Pp. 532–545].
- бустинг закономерностей (см. далее)
  — [*Cohen W. W., Singer Y.* A simple, fast and effective rule learner // 16-th National Conference on Artificial Intelligence, 1999., Pp. 335–342].

Определение решающего списка

Решающий список (decision list, DL)
— алгоритм классификации $a\colon X \to Y$, который задаётся закономерностями $\varphi_1(x), \ldots, \varphi_T(x)$ классов $c_1, \ldots, c_T \in Y$:

1: **для всех** $t = 1, \ldots, T$
2: **если** $\varphi_t(x) = 1$ **то**
3: **вернуть** c_t;
4: **вернуть** c_0.



«Особый ответ» c_0 — отказ от классификации объекта x.

Построение решающего списка (жадный алгоритм покрытия выборки)

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

1: $U := X^\ell$;
2: **для всех** $t := 1, \ldots, T_{\max}$
3: $c := c_t$ — выбрать класс из Y;
4: поиск лучшего правила по выборке U:
$\varphi_t := \arg\max I_c(\varphi, U)$ по всем $\varphi \in \Phi$: $E_c(\varphi, U) \leqslant E_{\max}$;
5: **если** $I_c(\varphi_t, U) < I_{\min}$ **то выход**;
6: исключить из выборки объекты, выделенные правилом φ_t:
$U := \{x \in U : \varphi_t(x) = 0\}$;
7: **если** $|U| \leqslant \ell_0$ **то выход**;

Замечания к алгоритму построения решающего списка

- Параметр $E_{\max}$ позволяет управлять сложностью списка:
  $E_{\max} \downarrow \quad \Rightarrow \quad p(\varphi_t) \downarrow,\ T \uparrow.$
- Стратегии выбора класса c_t:
  1) все классы по очереди;
  2) на каждом шаге определяется оптимальный класс:
  $$(\varphi_t, c_t) := \underset{\varphi \in \Phi', c \in Y}{\arg\max} I_c(\varphi, U);$$
- Простой обход проблемы пропусков в данных.
- Другие названия:
  — комитет с логикой старшинства;
  — голосование по старшинству;
  — машина покрывающих множеств (SCM);

Решающие списки: достоинства и недостатки

Достоинства:

- Интерпретируемость и простота классификации.
- Гибкость: можно варьировать множество Ф.
- Допустимы разнотипные данные и данные с пропусками.

Недостатки:

- При неудачном выборе Ф список может не построиться, будет много отказов от классификации.
- Список плохо интерпретируется, если он длинный и/или правила различных классов следуют вперемежку.
- Каждый объект классифицируется только одним правилом, правила не могут компенсировать ошибки друг друга.

Простое и взвешенное голосование

Пусть $\varphi_c^t(x),\ \ t = 1, \ldots, T_c$ — закономерности класса c.

Простое голосование (simple voting):

$$a(x) = \arg\max_{c \in Y} \underbrace{\frac{1}{T_c} \sum_{t=1}^{T_c} \varphi_c^t(x)}_{\Gamma_c(x)}.$$

Взвешенное голосование (weighted voting):

$$a(x) = \arg\max_{c \in Y} \underbrace{\sum_{t=1}^{T_c} \alpha_c^t \varphi_c^t(x)}_{\Gamma_c(x)}.$$

$\Gamma_c(x)$ — сумма «голосов» правил за класс c.

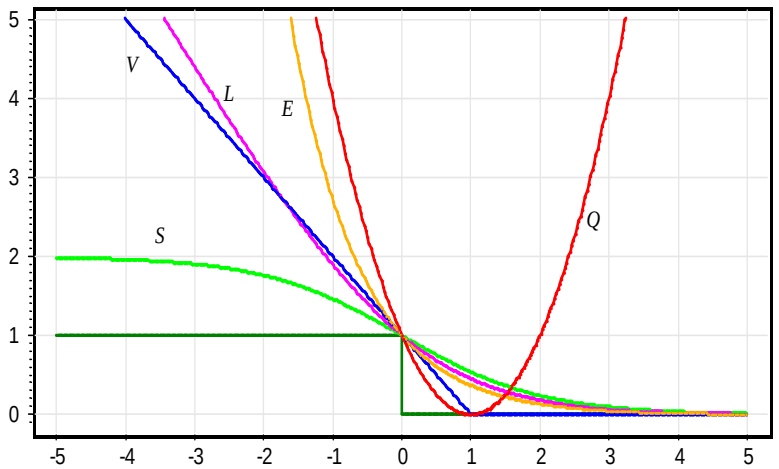
Отступ объекта x_i: $M(x_i) = \Gamma_{y_i}(x_i) - \max\limits_{y \in Y \setminus y_i} \Gamma_y(x_i)$.

Бустинг закономерностей

Пусть (для простоты) только два класса, $Y = \{-1, +1\}$.
Функционал числа ошибок композиции на обучающей выборке:

$$Q(\alpha_c^t, \varphi_c^t) = \sum_{i=1}^{\ell} \big[M(x_i) < 0\big] \leqslant \sum_{i=1}^{\ell} \exp\big(-M(x_i)\big) \to \min_{\alpha_c^t, \varphi_c^t}.$$

Экспоненциальная аппроксимация пороговой функции потерь:



Следствия экспоненциальной аппроксимации

Будем строить правила последовательно, оптимизируя $\alpha_c^t \varphi_c^t$ при фиксированных $\alpha^1\varphi^1, \ldots, \alpha^{t-1}\varphi^{t-1}$.
Тогда

Теорема

Минимум функционала $\tilde{Q}(\alpha_c, \varphi_c)$ *достигается при*

$$\varphi_c^* = \arg\max_{\varphi} \sqrt{p_c^w(\varphi)} - \sqrt{n_c^w(\varphi)};$$

$$\alpha_c^* = \frac{1}{2} \ln \frac{p_c^w(\varphi_c^*)}{n_c^w(\varphi_c^*)};$$

$$p_c^w(\varphi) = \sum_{i=1}^{\ell} w_i [y_i = c][\varphi(x_i) = 1];$$

$$n_c^w(\varphi) = \sum_{i=1}^{\ell} w_i [y_i \neq c][\varphi(x_i) = 1];$$

Алгоритм AdaBoost для взвешенного голосования правил

Вход: выборка X^ℓ; семейство правил Φ; параметры T, λ;
Выход: закономерности и их веса $\varphi_c^t(x), \alpha_c^t$, $t = 1..T_c$, $c \in Y$;

1: инициализация: $w_i := 1$, $i = 1, \ldots, \ell$;
2: **для всех** $t = 1, \ldots, T$
3: $\quad c := c_t$ — выбрать класс закономерности;
4: $\quad \varphi_c^t := \arg\max\limits_{\varphi \in \Phi} \sqrt{p_c^w(\varphi)} - \sqrt{n_c^w(\varphi)}$;
5: $\quad \alpha_c^t := \frac{1}{2} \ln \frac{p_c^w(\varphi)}{n_c^w(\varphi) + \lambda}$;
6: $\quad$ **для всех** $i = 1, \ldots, \ell$
7: $\quad\quad$ **если** $\varphi_c(x_i) = 1$ **то** $w_i := \begin{cases} w_i \exp(-\alpha_c^t), & y_i = c; \\ w_i \exp(\alpha_c^t), & y_i \neq c; \end{cases}$
8: $\quad$ нормировка: $Z := \frac{1}{\ell} \sum\limits_{i=1}^{\ell} w_i$; $w_i := w_i / Z$, $i = 1, \ldots, \ell$;

Алгоритм AdaBoost для закономерностей

Достоинства:

- Переобучение, как правило, не увеличивается с ростом T за счёт максимизации и выравнивания отступов.
- Возможность выделения и фильтрации шумовых объектов.
- Универсальность: можно использовать любое семейство Ф.
- Низкая чувствительность к пропускам в данных.

Недостатки:

- В некоторых задачах переобучение всё же наблюдается, хотя, при очень больших $T \sim 10^3$.
- Утрата интерпретируемости при больших T.

Решающие деревья и леса

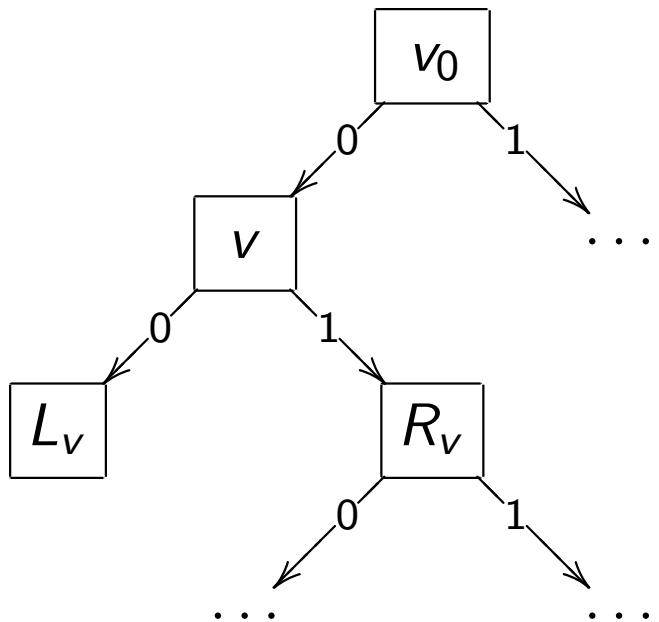
- Базовый вариант
  — алгоритм ID3 [Quinlan,1986].
- Редукция улучшает обобщающую способность
  — алгоритмы C4.5, C5.0 [Quinlan], CART [Breiman, 1984], ...
- Альтернативные ветвления с весами
  — Alternating Decision Tree, ADTree [Freund, Mason, 1999].
- Линейные комбинации признаков в узлах
  — Oblique Decision Tree.
- Линейные комбинации деревьев, решающие леса:
  — бустинг или бэггинг;
  — решающий список над решающими деревьями;
  — TreeNet [Friedman, 1999].

Определение бинарного решающего дерева

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

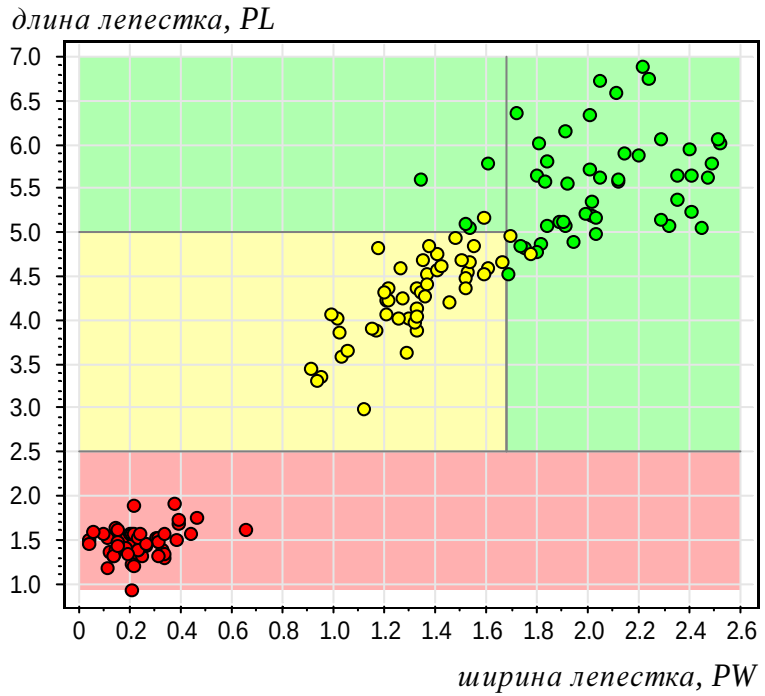
1) $\forall v \in V_{\text{внутр}} \;\rightarrow\;$ предикат $\beta_v : X \to \{0,1\}$, $\beta \in \mathscr{B}$
2) $\forall v \in V_{\text{лист}} \;\rightarrow\;$ имя класса $c_v \in Y$.

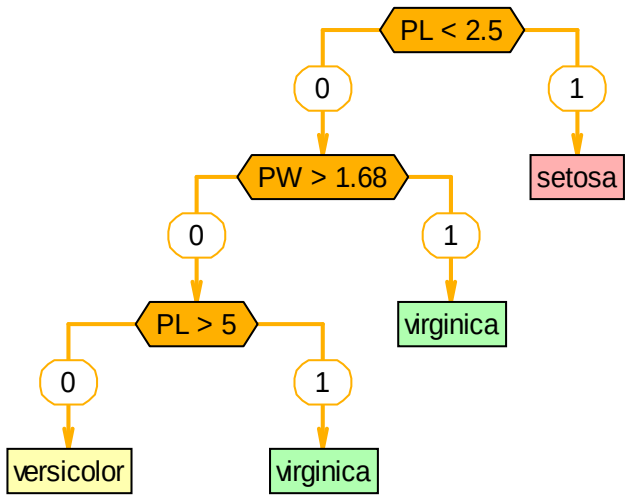
1: $v := v_0$;
2: **пока** $v \in V_{\text{внутр}}$
3: **если** $\beta_v(x) = 1$ **то**
4: переход вправо:
$v := R_v$;
5: **иначе**
6: переход влево:
$v := L_v$;
7: **вернуть** c_v.



Пример решающего дерева

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





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

Жадный алгоритм построения дерева ID3

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

Разновидности критериев ветвления

1. **Отделение одного класса:**

$$I(\beta, X^\ell) = \max_{c \in Y} I_c(\beta, U).$$

2. **Многоклассовый энтропийный критерий:**

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

где $P_c = \#\{x_i : y_i = c\}$, $\quad h(z) \equiv -z \log_2 z$.

3. ***D*-критерий:**

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

Обработка пропусков

На стадии обучения:

- Если $\beta(x_i)$ не определено, то при вычислении $I(\beta, U)$ объект x_i исключается из выборки U.
- Для $\forall v \in V_{\text{внутр}}$ оценивается:
  $\hat{p}_L = |U_0|/|U|$ — вероятность левой ветви;
  $\hat{p}_R = |U_1|/|U|$ — вероятность правой ветви.

На стадии классификации:

- $\beta_v(x)$ не определено $\Rightarrow$ *пропорциональное распределение*:

$$\hat{P}_v(y|x) = \begin{cases} [y = c_v], & v \in V_{\text{лист}}; \\ \hat{p}_L \hat{P}_{L_v}(y|x) + \hat{p}_R \hat{P}_{R_v}(y|x), & v \in V_{\text{внутр}}. \end{cases}$$

- Окончательное решение — байесовское правило:
  $y = \arg\max\limits_{y \in Y} \hat{P}_v(y|x)$.

Решающие деревья: достоинства и недостатки

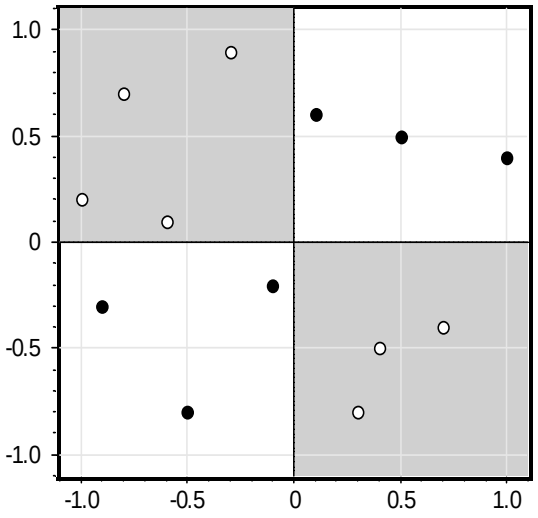
Достоинства:

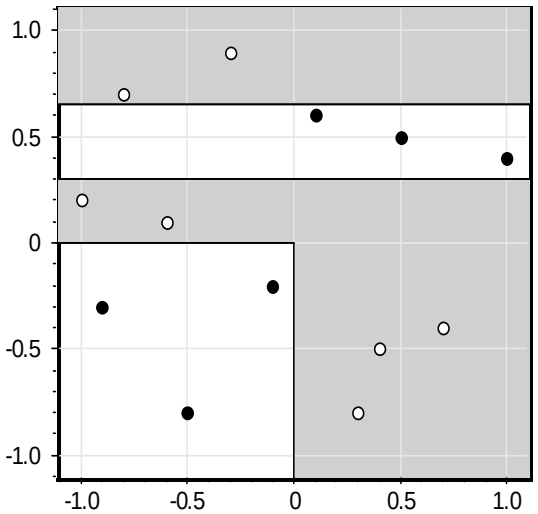
- Интерпретируемость и простота классификации.
- Гибкость: можно варьировать множество $\mathscr{B}$.
- Допустимы разнотипные данные и данные с пропусками.
- Трудоёмкость линейна по длине выборки $O(|\mathscr{B}|h\ell)$.
- Не бывает отказов от классификации.

Недостатки:

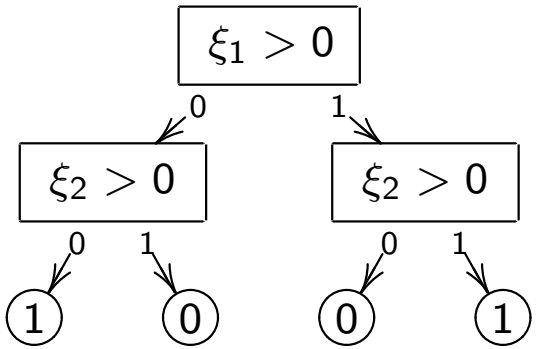
- Жадный ID3 переусложняет структуру дерева.
- Чем дальше v от корня, тем меньше выборка $|U|$, тем меньше статистическая надёжность выбора β_v, c_v.
- Высокая чувствительность к составу выборки.
- Как правило, ID3 сильно переобучается.

Жадный ID3 переусложняет структуру дерева

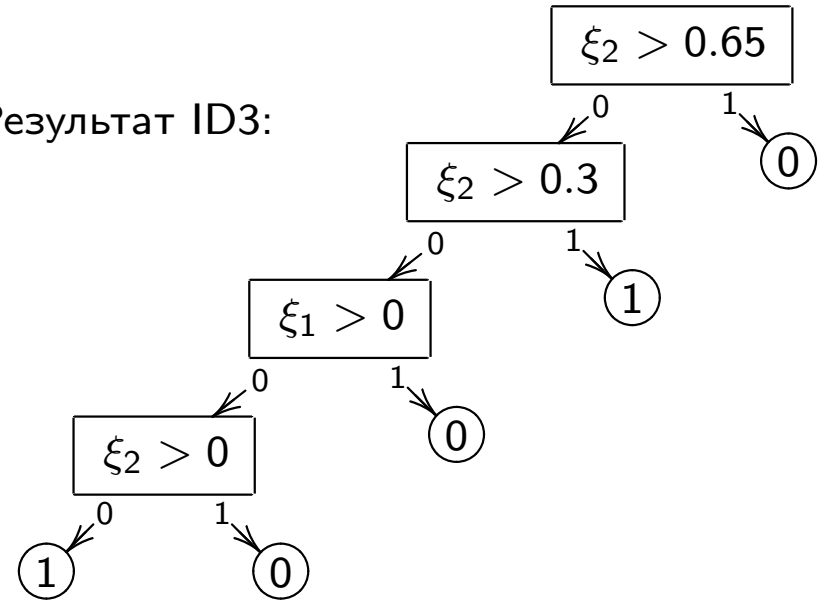




Оптимальное дерево для задачи XOR:



Результат ID3:



Стратегия пред-просмотра (look ahead)

Шаг 6:
если $U_0 = \varnothing$ или $U_1 = \varnothing$ **то**
    вернуть новый лист v, $c_v :=$ Мажоритарный класс(U);

Шаг 6 заменяется на более ресурсоёмкую процедуру:

для всех деревьев T глубины h
    $r(U) =$ число ошибок дерева T на выборке U;
вернуть новый лист v, $\beta_v :=$ корень лучшего поддерева;

Достоинства:

- Задача XOR уже решается почти идеально.

Недостатки:

- При $h > 2$ ооооооочень долго.
- На реальных данных улучшение незначительно.

Стратегия пред-редукции (pre-pruning)

Шаг 6:
если $U_0 = \varnothing$ или $U_1 = \varnothing$ **то**
    вернуть новый лист v;

Шаг 6 заменяется на более мягкое условие:

если $I(\beta, U) \leqslant I_0$ **то**
    вернуть новый лист v;

Достоинства:

- Сразу строится более простое дерево.

Недостатки:

- Качество дерева может и не улучшиться.

Стратегия пост-редукции (post-pruning)

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

Преобразование решающего дерева в решающий список

- Для любого бинарного решающего дерева

$$a(x) = \arg\max_{y \in Y} \sum_{v \in V_{\text{лист}}} [c_v = y] K_v(x),$$

где $K_v(x)$ — конъюнкция по всем рёбрам пути из v_0 в v:

$$K_v(x) = \bigwedge_{(u,R_u)} \beta_u(x) \bigwedge_{(u,L_u)} \neg\beta_u(x).$$

- Редукция конъюнкций $K_v(x)$, $\forall v \in V_{\text{лист}}$ по контрольной выборке X^k.

Достоинства:

- Переобучение, как правило, уменьшается.

Недостатки:

- Преобразование в список необратимо: это уже не дерево.

Решающие леса

Случайные леса:

- T деревьев обучаются по случайным подвыборкам (bagging).
- Построение внутренних вершин, $\beta := \arg\max_{\beta \in \mathscr{B}} I(\beta, U)$:

  $\mathscr{B}$ — случайное множество гиперплоскостей;
  $I(\beta, U)$ — энтропийный критерий информативности.
- Простое голосование по T деревьям.

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.