Задачи и методы машинного обучения

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

Кафедра «Интеллектуальные системы» ФУПМ МФТИ,
ЗАО «Фóрексис»,
Вычислительный Центр РАН,
Школа анализа данных Яндекс

Зимняя компьютерная школа МФТИ,
Долгопрудный, 8-15 января 2010

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, $f_1(x_i), \ldots, f_n(x_i)$;
  - $y_i = y(x_i)$ — класс объекта x_i (чаще всего $y_i \in \{0, 1\}$);
- **Построить:**
  - *классификатор* $a(x)$ — функцию, приближающую неизвестную $y(x)$ как можно точнее для всех $x \in X$.
- **Примеры прикладных задач:**
  - медицинская диагностика и прогнозирование;
  - кредитный скоринг (предсказание дефолта заёмщика);
  - предсказание ухода абонента от оператора сотовой связи;
  - распознавание личности по фотопортрету;
  - распознавание спама;
  - распознавание тематики текста;

Некоторые методы классификации

- Метод ближайших соседей;
- Наивный байесовский классификатор;
- Метод потенциальных функций;
- Нейронная сеть;
- Решающий список (комитет старшинства);
- Решающее дерево;
- Решающий лес;
- Машина опорных векторов;
- Логистическая регрессия;
- Голосование закономерностей (комитет большинства);
- Смесь экспертов;
- Композиция любых более простых методов;
- ... ...

Метод k ближайших соседей

Гипотеза компактности:

Схожие объекты, как правило, лежат в одном классе.

Пусть задана функция расстояния $\rho(x, x')$.

Алгоритм классификации требует хранить всю выборку:

$$a(x; X^\ell, k) = \arg\min_y \sum_{i=1}^{k} \rho\left(x, x^{(i,y)}\right),$$

где $x^{(i,y)}$ — i-й сосед объекта x среди $\{x_i : y_i = y\}$:

$$\rho(x, x^{(1,y)}) \leqslant \cdots \leqslant \rho(x, x^{(k,y)}) \leqslant \cdots$$

Маленькие хитрости: как хранить и сортировать выборку.

И есть ещё две проблемы...

Как выбрать число соседей k и метрику ρ?

1. Выбор числа k по скользящему контролю:

$$\mathrm{CK}(k) = \sum_{i=1}^{\ell} \Big[a\big(x_i; X^{\ell}\backslash\{x_i\}, k\big) \neq y_i \Big] \to \min_k .$$

Увы, в реальных задачах минимум редко бывает при $k = 1$.

2. Выбор метрики ρ (также по скользящему контролю):

$$\rho(x, x') = \left(\sum_{j \in J} \big| f_j(x) - f_j(x') \big|^{\gamma} \right)^{\frac{1}{\gamma}}.$$

При $\gamma = 2$ это обычное евклидово расстояние.

Почему J не обязательно $= \{1, \ldots, n\}$? Потому, что:
(1) боимся... **проклятия размерности**,
(2) хотим избавиться от неинформативных признаков.

Проблема отбора признаков

Суть проблемы:
обычно имеется много неинформативных признаков.

Методы отбора признаков:

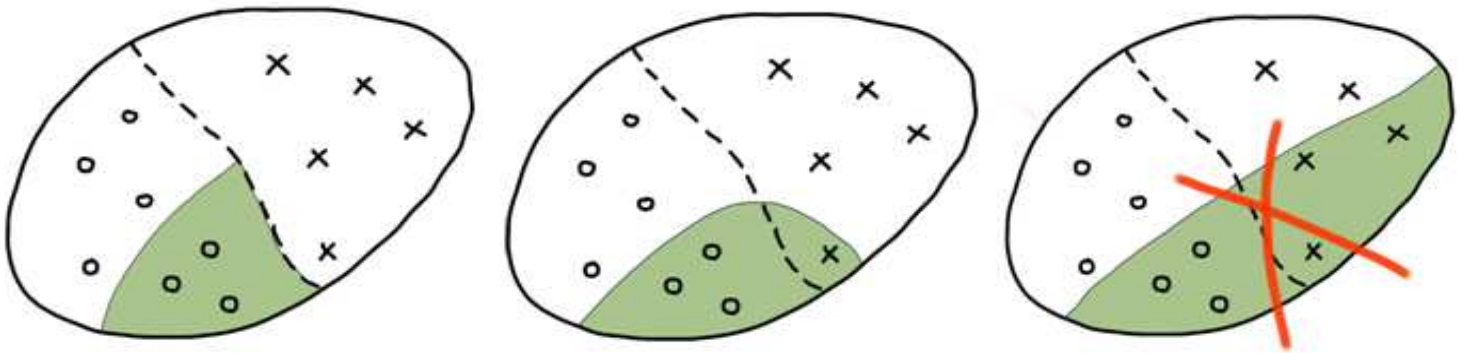
- перебор нескольких «разумных» вариантов вручную;
- полный перебор 2^n вариантов $J \subset \{1, \ldots, n\}$;
- случайный поиск;
- жадное добавление (см. далее);
- случайный поиск с адаптацией;
- поиск в глубину;
- поиск в ширину;
- генетический алгоритм;
- ... ...

Понятие закономерности

Логическая закономерность (правило, rule) класса y — это функция $r(x)$, $r\colon X \to \{0,1\}$:

- $r(x)$ определяется простой понятной формулой;
- $r(x) = 1$ преимущественно на объектах класса y:

$$p_y(r) = \#\{x_i\colon r(x_i)=1 \text{ и } y_i=y\} \to \max;$$
$$n_y(r) = \#\{x_i\colon r(x_i)=1 \text{ и } y_i\neq y\} \to \min;$$

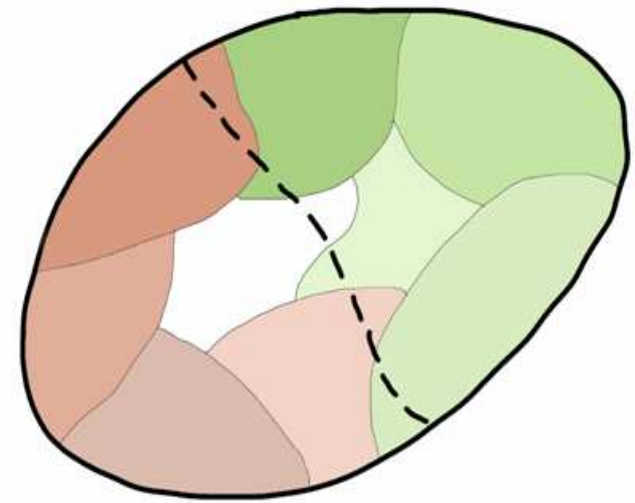


Как из закономерностей строить классификатор?

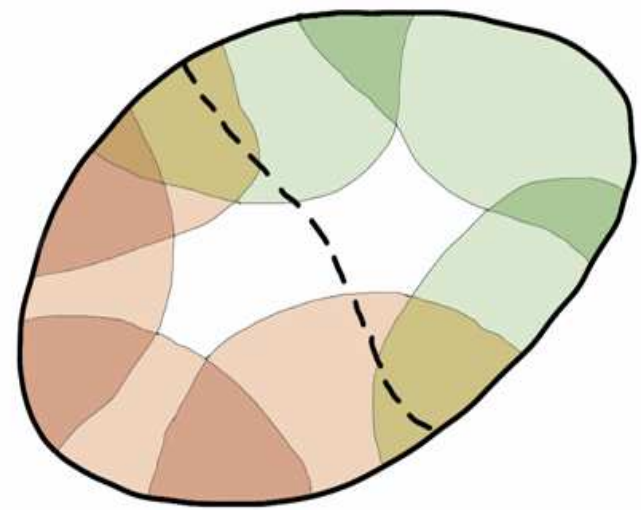
Каждое правило выделяет лишь небольшую область объектов.
Следовательно, правил нужно много.

Две основные идеи:

покрытие
(комитет старшинства)



голосование
(комитет большинства)



Что значит «простая понятная формула»?

Пример 1: распознавание спама

Если текст содержит «Иностранные работники»
и искажённый телефонный номер,
то это спам.

Пример 2: прогноз исхода операции

Если возраст пациента > 65
и увеличение скорости кровотока в артерии $> 50\%$
и перепад концентрации лимфоцитов $< 20\%$,
то высок риск повторной операции.

Итак, **требования интерпретируемости:**
1) r зависит от небольшого числа признаков;
2) формула r выражается на естественном языке.

Виды закономерностей

Параметрическое семейство *конъюнкций пороговых термов*:

$$r(x) = \prod_{j \in J} [f_j(x) \leqslant \alpha_j].$$

Параметрическое семейство *полуплоскостей*:

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

Параметрическое семейство *шаров*:

$$r(x) = [\rho(x, x_i) \leqslant R], \quad \rho(x, x_i) = \sum_{j \in J} \alpha_j \left| f_j(x) - f_j(x_0) \right|.$$

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

Критерии информативности

Два критерия $p(r) \to \max$, $n(r) \to \min$ — это не удобно!

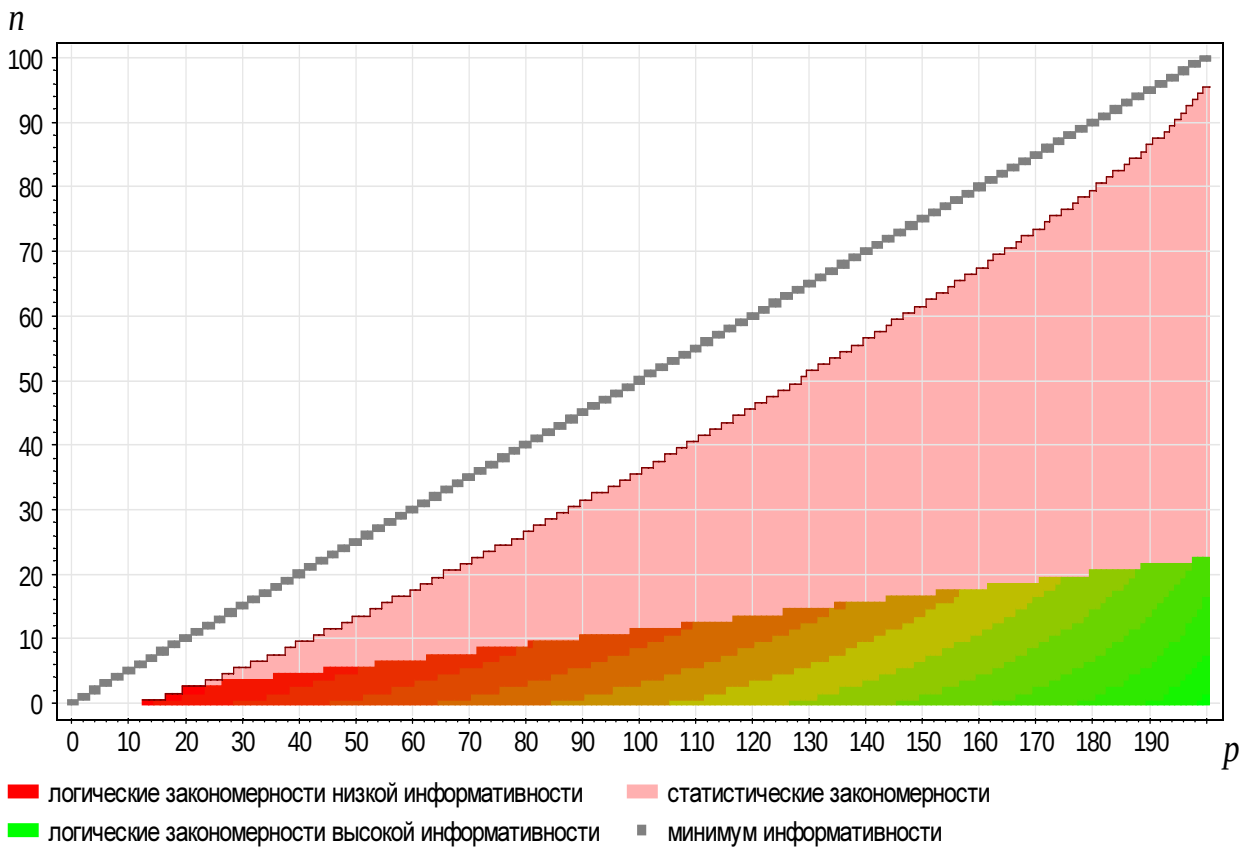
Варианты критериев:

- $E(r) = \dfrac{n}{p+n} \leqslant \varepsilon$; // доля ошибок;
- $I(r) = \sqrt{p} - \sqrt{n} \to \max$; // хороший критерий;
- $I(r) = \sqrt{\dfrac{p}{P}} - \sqrt{\dfrac{n}{N}} \to \max$, где $P{=}p(1)$, $N{=}n(1)$;
- $I(r) = p - \lambda n \to \max$; // не ясно, как выбрать λ;
- $I(r) = \dfrac{C_{P+N}^{p+n}}{C_P^p C_N^n} \to \max$; // степень неслучайности;

Вообще, критериев информативности придумано более 20...

Где живут закономерности в плоскости p, n?

$E(r) \leqslant \varepsilon$ — доля ошибок не выше заданной ε;
$I(r) \to \max$ — информативность как можно выше;
Пример при $\varepsilon = 0.1$, $P = 200$, $N = 100$.



Локальный поиск закономерностей

Пусть закономерности — это конъюнкции:

$$r(x) = \prod_{j \in J} [f_j(x) \leqslant \alpha_j].$$

Опр. *Окрестность* $V(r)$ — это множество всех конъюнкций, получаемых из $V(r)$ путём добавления, удаления или модификации одного из термов.

Основная идея: на t-й итерации найти лучшую конъюнкцию в окрестности (или в её случайном подмножестве, если полный перебор слишком долгий):

$$r_t := \arg\max_{r \in V(r_{t-1})} I_y(r).$$

Алгоритм локального поиска

Вход: выборка X^ℓ; класс $y \in Y$;
начальное приближение r_0; параметры T, d, ε;
Выход: конъюнкция r;

1: $I^* := I_y(r_0)$; $r^* := r_0$;
2: **для всех** $t = 1, \ldots, T$
3: поиск лучшей конъюнкции в окрестности r_{t-1}:
$r_t^* := \arg\max I_y(r)$ по всем $r \in V(r_{t-1})$: $\underline{E_y(r) < \varepsilon}$;
4: **если** $I_y(r_t^*) > I^*$, **то**
$t^* := t$; $r^* := r_t^*$; $I^* := I_y(r^*)$
5: **если** $t - t^* > d$, **то**
6: **выход**;
7: наиболее перспективная конъюнкция:
$r_t := \arg\max I_y(r)$ по всем $r \in V(r_{t-1})$;
8: **вернуть** r^*;

Частные случаи

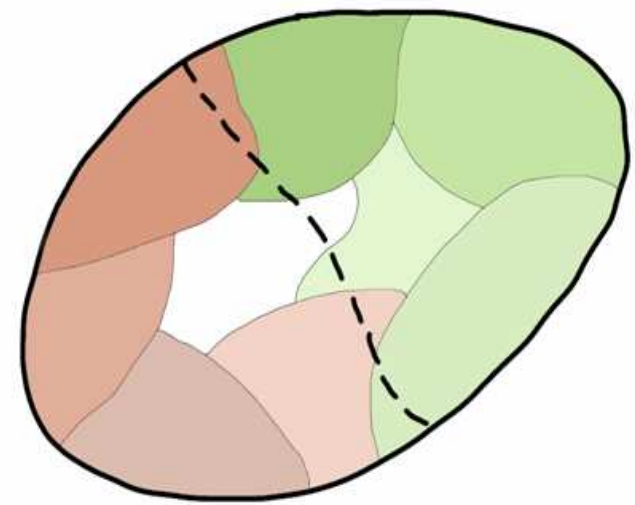
- **жадный алгоритм:**
  $V(r)$ — только добавления термов; $r_0 = \varnothing$;
- **стохастический локальный поиск (SLS):**
  $V(r)$ — случайное подмножество всевозможных добавлений, удалений, модификаций термов; $r_0 = \varnothing$;
- **стабилизация:**
  $V(r)$ — удаления термов или изменение параметров в термах; $r_0 \neq \varnothing$;
  (рекомендуется для финальной настройки порогов α_j)

Как из закономерностей строить классификатор?

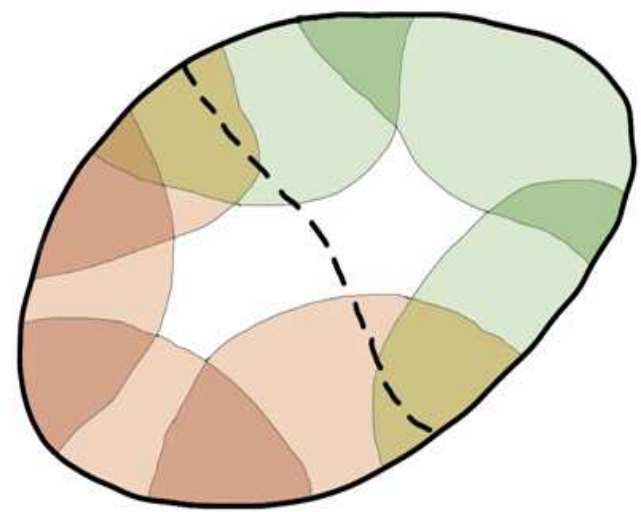
Каждое правило выделяет лишь небольшую область объектов.
Следовательно, правил нужно много.

Две основные идеи:

покрытие
(комитет старшинства)



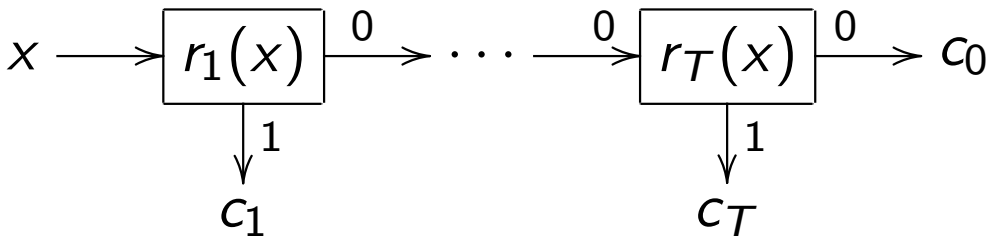
голосование
(комитет большинства)



Решающий список закономерностей

Решающий список (комитет старшинства)
— алгоритм классификации $a\colon X \to Y$, который задаётся закономерностями $r_1(x), \ldots, r_T(x)$ классов $c_1, \ldots, c_T \in Y$:

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



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

Построение решающего списка

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

1: $U := X^\ell$;
2: **для всех** $t := 1, \ldots, T$
3: $y := c_t$ — выбрать, за какой класс строим правило;
4: поиск лучшего правила по выборке U:
$r_t := \arg\max I_y(r, U)$ по всем r: $E_y(r, U) \leqslant \varepsilon$;
5: **если** найти такое r_t не удалось или $I_y(r_t, U) < I_{\min}$ **то**
выход;
6: исключить из выборки объекты, выделенные правилом r_t:
$U := \{x \in U : r_t(x) = 0\}$;
7: **если** $|U| \leqslant \ell_0$ **то выход**;

Замечания к алгоритму построения решающего списка

- Параметр ε позволяет управлять сложностью списка:
  $\varepsilon\downarrow \quad \Rightarrow \quad p(r_t)\downarrow \quad \Rightarrow \quad T\uparrow$.
- Возможные стратегии выбора класса c_t на шаге 3:
  1) все классы по очереди;
  2) на каждом шаге определяется оптимальный класс:
  $$(r_t, c_t) := \arg\max_{r,y} I_y(r, U);$$
- Простой обход проблемы пропусков в данных.
- Простое обобщения на правила любого вида.
- Другие названия:
  — комитет с логикой старшинства;
  — голосование по старшинству;
  — машина покрывающих множеств (SCM);

Голосование закономерностей

Пусть $r_{yt}(x)$, $t = 1, \ldots, T_y$ — закономерности класса y.

Простое голосование:

$$a(x) = \arg\max_{y \in Y} \underbrace{\frac{1}{T_y} \sum_{t=1}^{T_y} r_{yt}(x)}_{\Gamma_y(x)}.$$

Взвешенное голосование:

$$a(x) = \arg\max_{y \in Y} \underbrace{\sum_{t=1}^{T_y} \alpha_{yt} r_{yt}(x)}_{\Gamma_y(x)}.$$

$\Gamma_y(x)$ — сумма «голосов» правил за класс y.

Перевес голосов (отступ, margin) объекта x_i:

$$M(x_i) = \Gamma_{y_i}(x_i) - \max_{y \neq y_i} \Gamma_y(x_i).$$

Бустинг закономерностей: основные идеи

Задача: добавить в композицию ещё одно правило $\alpha_{yt} r_{yt}(x)$.

Вводим веса объектов $w_i = \frac{1}{Z} e^{-M(x_i)}$, где $Z = \frac{1}{\ell} \sum_{i=1}^{\ell} e^{-M(x_i)}$.

Ищем правило $r_{yt}(x)$ по максимуму *w-информативности:*

$$I_y(r) = \sqrt{p_y(r)} - \sqrt{n_y(r)} \to \max_r;$$

$$p_y(r) = \sum_{i=1}^{\ell} w_i [y_i = y][r(x_i) = 1];$$

$$n_y(r) = \sum_{i=1}^{\ell} w_i [y_i \neq y][r(x_i) = 1].$$

Тогда (это теорема!) оптимальное $\alpha_{yt} = \frac{1}{2} \ln \frac{p_y(r_{yt})}{n_y(r_{yt})}$.

Бустинг закономерностей: алгоритм AdaBoost

Вход: выборка X^ℓ; параметр T;
Выход: закономерности и их веса $r_{yt}(x), \alpha_{yt}$, $t = 1..T_y$, $y \in Y$;

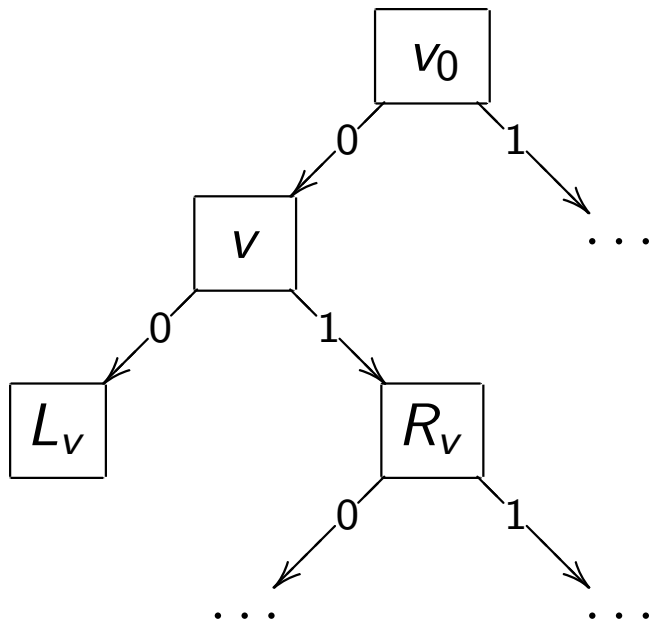
1: инициализация: $w_i := 1$, $i = 1, \ldots, \ell$;
2: **для всех** $t = 1, \ldots, T$
3: выбрать класс закономерности y;
4: $r_{yt} := \arg\max_r \sqrt{p_y(r)} - \sqrt{n_y(r)}$;
5: $\alpha_{yt} := \frac{1}{2} \ln \frac{p_y(r)}{n_y(r)+1}$;
6: **для всех** $i = 1, \ldots, \ell$
7: **если** $r_y(x_i) = 1$ **то** $w_i := \begin{cases} w_i \exp(-\alpha_{yt}), & y_i = y; \\ w_i \exp(\alpha_{yt}), & y_i \neq y; \end{cases}$
8: нормировка: $Z := \frac{1}{\ell} \sum_{i=1}^{\ell} w_i$; $w_i := w_i / Z$, $i = 1, \ldots, \ell$;

Определение бинарного решающего дерева

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

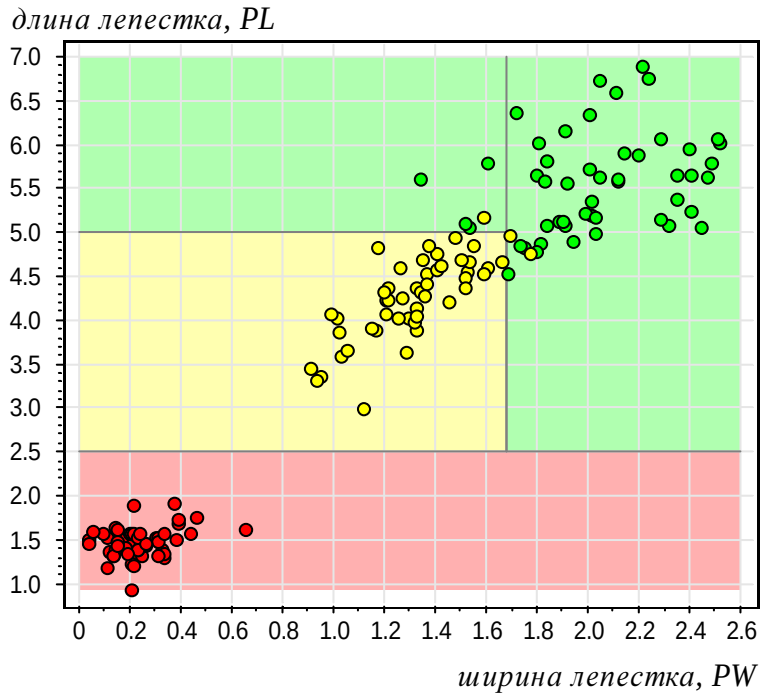
1) $\forall v \in V_{\text{внутр}} \;\rightarrow$ условие ветвления $\beta_v \colon X \to \{0,1\}$;
2) $\forall v \in V_{\text{лист}} \;\rightarrow$ имя класса $c_v \in Y$.

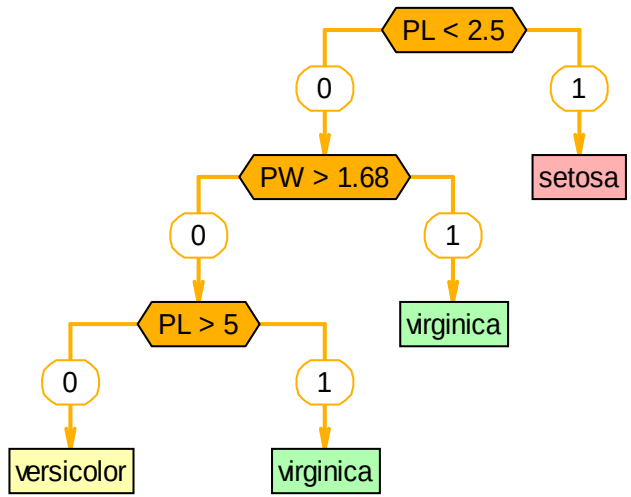
1: $v := v_0$;
2: **пока** $v \in V_{\text{внутр}}$
3: **если** $\beta_v(x) = 1$ **то**
4: переход вправо:
$v := R_v$;
5: **иначе**
6: переход влево:
$v := L_v$;
7: **вернуть** c_v.



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.

Жадный алгоритм построения дерева ID3

ПРОЦЕДУРА LearnID3 $(U \subseteq X^\ell)$;
если все объекты из U лежат в одном классе $y \in Y$ **то**
    вернуть новый лист v, $c_v := y$;
найти условие ветвления с максимальной информативностью:
$\beta := \arg\max_\beta I(\beta, U)$;
разбить выборку на две части $U = U_0 \cup U_1$ по условию β:
$U_0 := \{x \in U : \beta(x) = 0\}$;
$U_1 := \{x \in U : \beta(x) = 1\}$;
если $I(\beta, U) \leqslant I_0$ **то**
    вернуть новый лист v, $c_v :=$ Большинство(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;

Снова голосование

Допустим (для простоты), что классов два: $Y = \{-1, +1\}$.
Теперь голосуют не закономерности, а классификаторы:

$$a(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t b_t(x)\right).$$

где $b_t(x)$, $b_t : X \to [-1, +1]$ — «слабые» классификаторы.

Основной теоретический вопрос:
возможно ли из набора «слабых» классификаторов построить один «сильный» с помощью голосования?

Ответ — ДА! Если только b_t различны и хотя бы немного лучше случайного гадания.

Слабый классификатор. Пример №1

Элементарное пороговое условие:

$$b(x) = \big[f_j(x) \leqslant \alpha\big]$$

Как настроить порог α по выборке $(x_i, y_i)_{i=1}^{\ell}$?

Полным перебором: $\alpha \in \big\{f_j(x_i)\colon i = 1, \ldots, \ell\big\}$,
так, чтобы средневзвешенное число ошибок

$$Q(\alpha, X^\ell) = \sum_{i=1}^{\ell} w_i \big[b(x_i) \neq y_i\big] \to \min,$$

где веса объектов w_i задаются бустингом, см. далее...

Снова бустинг: знаменитый AdaBoost

Вход: выборка X^ℓ; параметр T;

Выход: классификаторы b_t и их веса α_t, $t = 1..T$;

1: инициализация: $w_i := 1$, $i = 1, \ldots, \ell$;

2: **для всех** $t = 1, \ldots, T$

3: $b_t := \arg\min_b \underbrace{\frac{1}{\ell} \sum_{i=1}^{\ell} w_i \big[b(x_i) \neq y_i\big]}_{Q(b)}$;

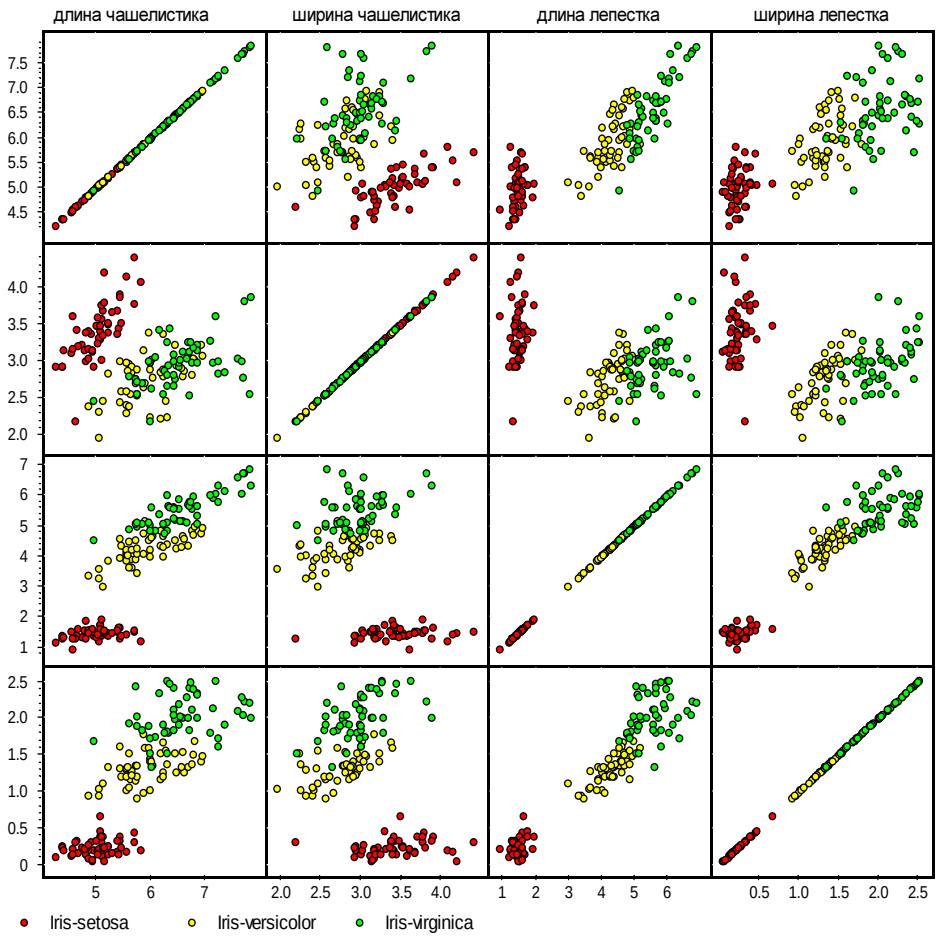
4: $\alpha_t := \frac{1}{2} \ln \frac{1 - Q(b_t)}{Q(b_t)}$;

5: $w_i := w_i \exp\big(-\alpha_t y_i b_t(x_i)\big)$, $i = 1, \ldots, \ell$;

6: нормировка: $Z := \sum_{i=1}^{\ell} w_i$; $w_i := \frac{1}{Z} w_i$, $i = 1, \ldots, \ell$;

Визуальный разведочный анализ данных

Пример: Снова ирисы Фишера: $\ell = 150,\ \ n = 4,\ \ |Y| = 3.$



Методология анализа данных

- Визуализация исходных данных (1-мерные, 2-мерные графики)
- Визуализация промежуточных данных (про распределение отступов)
- Предварительная обработка данных
  - проблема разнотипности данных
  - проблема переобучения (скользящий контроль)
- Как программировать алгоритмы (контест)
  - загрузка входных данных
  - функция обучения
  - функция классификации
  - запись выходных данных

Что ещё есть на свете?

- Системы с алгоритмами машинного обучения:
  WEKA — `www.cs.waikato.ac.nz/ml/weka`
  RapidMiner — `rapidminer.com`
- Репозиторий реальных задач UCI:
  `archive.ics.uci.edu/ml`
- Полигон алгоритмов классификации:
  `Poligon.MachineLearning.ru`
- Вики-ресурс на русском языке:
  `www.MachineLearning.ru`
  я там: «Участник:Vokov»