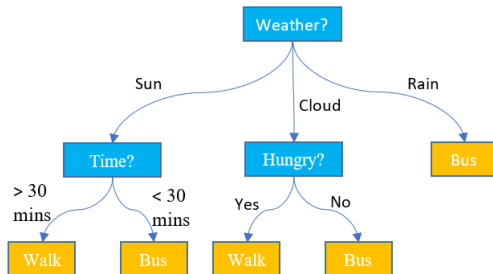


## Решающие деревья

Виктор Китов

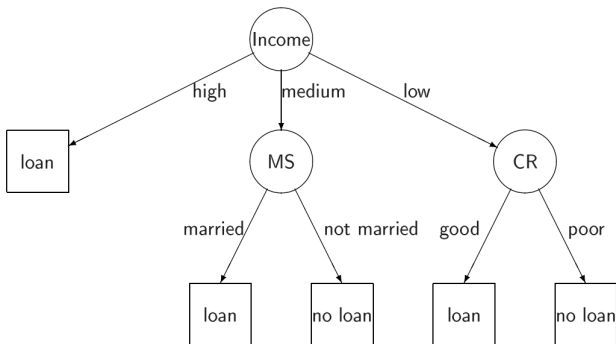
[v.v.kitov@yandex.ru](mailto:v.v.kitov@yandex.ru)



# Содержание

- 1 Понятие решающего дерева
- 2 Правила спуска
- 3 Выбор критерия ветвления
- 4 Назначение прогнозов листьям
- 5 Критерий остановки

## Пример решающего дерева



## Определение решающего дерева

- Прогнозы строятся деревом  $T$ .
- Для каждого внутреннего узла  $t$  задана функция ветвления  $Q_t(x)$ .
- Для каждого ребра  $1, \dots, K_t$  ассоциирован набор множеств  $S_t(1), \dots, S_t(K_t)$ .
  - $Q_t(x) \in S_t(i) \Rightarrow$  спуститься в узел  $i$ .
  - $\bigcup_k S_t(k) = \text{range}[Q_t(\cdot)]$
  - $S_t(i) \cap S_t(j) = \emptyset \ \forall i \neq j$

## Построение прогноза

- множество вершин разделяется на:
  - внутренние вершины  $int(T)$ , каждая имеет  $\geq 2$  потомков
  - терминальные вершины  $terminal(T)$ , которые не имеют дочерних, а ассоциированы с прогнозами.

## Построение прогноза

- множество вершин разделяется на:
  - внутренние вершины  $int(T)$ , каждая имеет  $\geq 2$  потомков
  - терминальные вершины  $terminal(T)$ , которые не имеют дочерних, а ассоциированы с прогнозами.
- Прогноз для дерева  $T$ :
  - $t = root(T)$
  - пока  $t$  - не терминальная вершина:
    - рассчитать  $Q_t(x)$
    - определить  $j$  такой, что  $Q_t(x) \in S_t(j)$
    - спуститься в  $j$ -ую дочернюю вершину  $t := t_j$
  - вернуть прогноз, ассоциированный с листом  $t$ .

## Спецификация решающего дерева

### Спецификация решающего дерева:

- функции ветвления  $Q_t(x) \forall t \in \text{IntNodes}$
- в каждом внутреннем узле:  $K_t$  и  $S_t(1), \dots, S_t(K_t)$
- прогноз в каждом листе дерева

## Спецификация решающего дерева

### Спецификация решающего дерева:

- функции ветвления  $Q_t(x) \forall t \in \text{IntNodes}$
- в каждом внутреннем узле:  $K_t$  и  $S_t(1), \dots, S_t(K_t)$
- прогноз в каждом листе дерева

### Спецификация обучения:

- критерий остановки
  - когда узел становится терминальным при построении top-down



# Содержание

- 1 Понятие решающего дерева
- 2 Правила спуска**
- 3 Выбор критерия ветвления
- 4 Назначение прогнозов листьям
- 5 Критерий остановки

## Возможные правила спуска (предикаты)

- $Q_t(x) = x^{i(t)}$ , где  $S_t(j) = v_j$ ,  $v_1, \dots, v_K$  - уникальные значения  $x^{i(t)}$ .
- $S_t(1) = \{x^{i(t)} \leq h_t\}$ ,  $S_t(2) = \{x^{i(t)} > h_t\}$
- $S_t(j) = \{h_j < x^{i(t)} \leq h_{j+1}\}$  для набора порогов  $h_1, h_2, \dots, h_{K_t+1}$ .
- $S_t(1) = \{x : \langle x, w \rangle \leq h\}$ ,  $S_t(2) = \{x : \langle x, w \rangle > h\}$
- $S_t(1) = \{x : \|x\| \leq h\}$ ,  $S_t(2) = \{x : \|x\| > h\}$
- и т.д.

## Самые популярные алгоритмы решающих деревьев

- CART (classification and regression trees)
  - реализован в scikit-learn
- C4.5

## Правила спуска для CART

- рассматривается единственный признак:

$$Q_t(x) = x^{i(t)}$$

- бинарные разбиения:

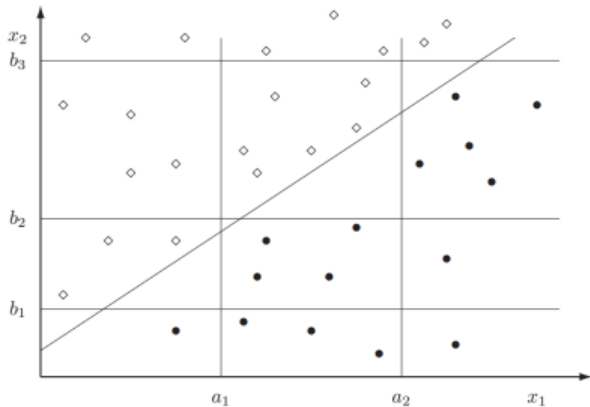
$$K_t = 2$$

- спуск основан на предикатах=сравнении с порогом  $h_t$ :

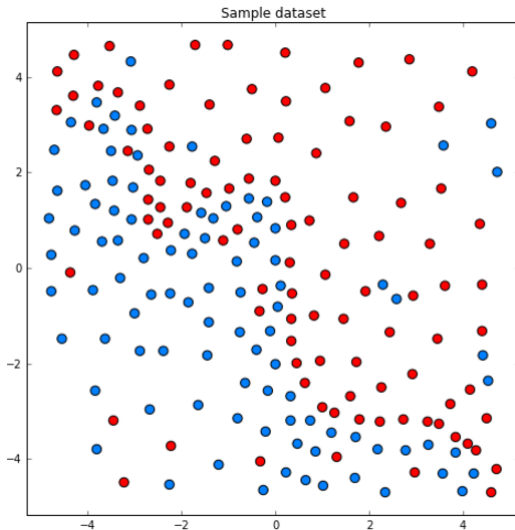
$$S_1 = \{x^{i(t)} \leq h_t\}, S_2 = \{x^{i(t)} > h_t\}$$

- достаточно выбрать порог из уникальных значений признака  $x^{i(t)}$ 
  - применимо для вещественных, порядковых и бинарных признаков
  - категориальные признаки: преобразуем в бинарные (one-hot) или вещественные (mean-value кодирование).

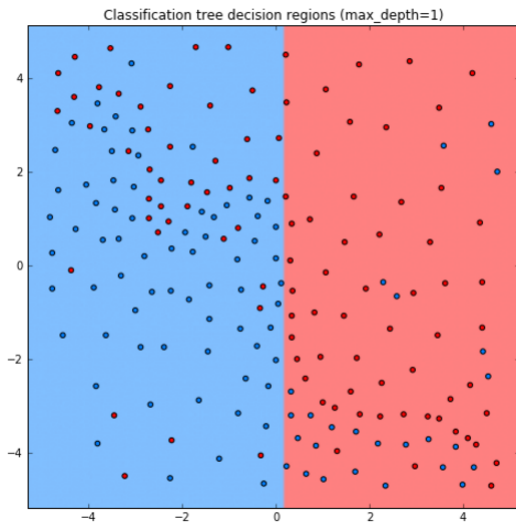
## Аппроксимация наклонных границ - много разбиений



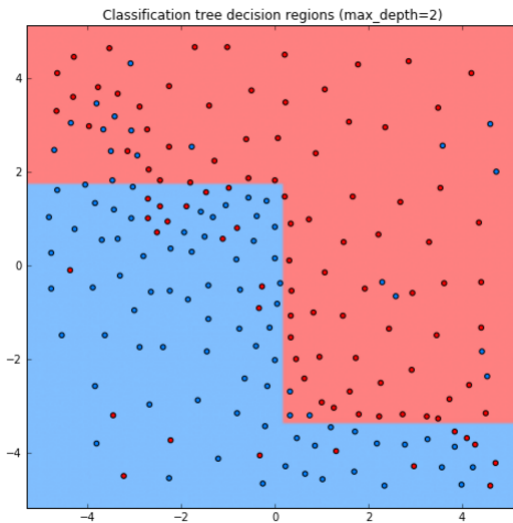
# Пример обучающей выборки



# Разбиение на классы (глубина=1)

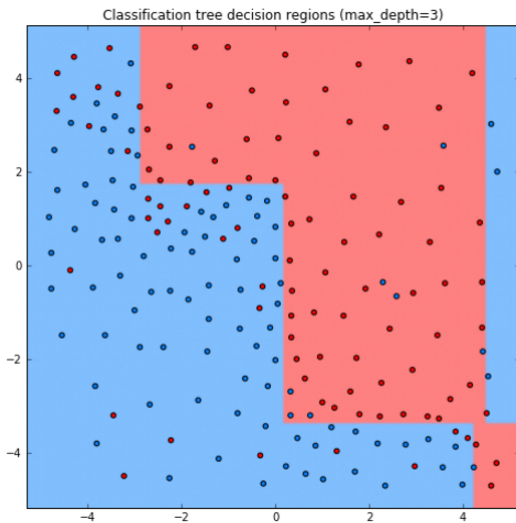


# Разбиение на классы (глубина=2)

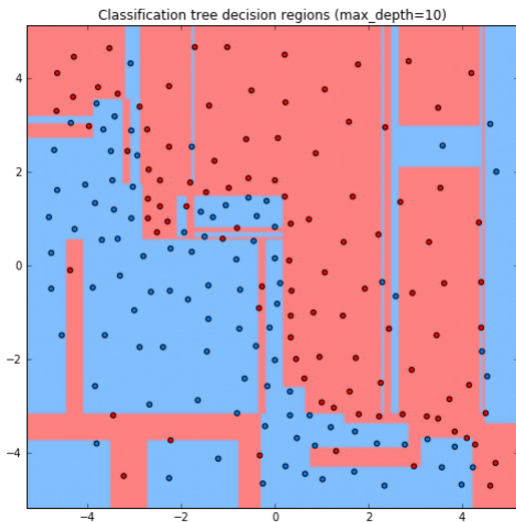




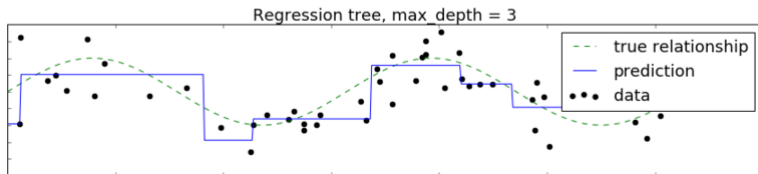
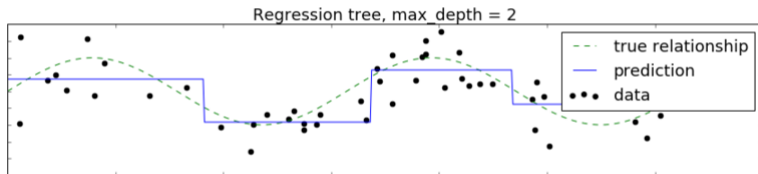
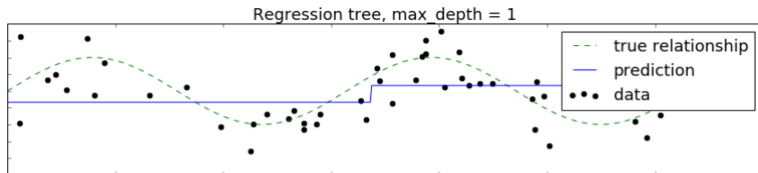
# Разбиение на классы (глубина=3)



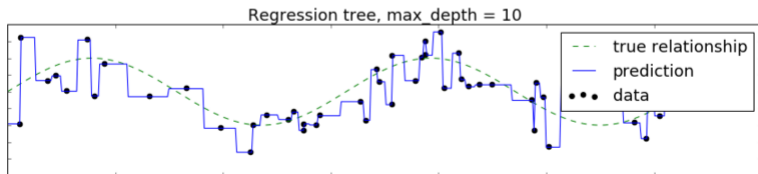
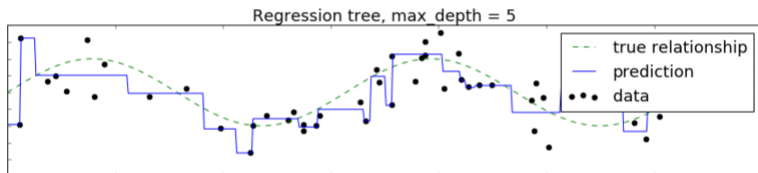
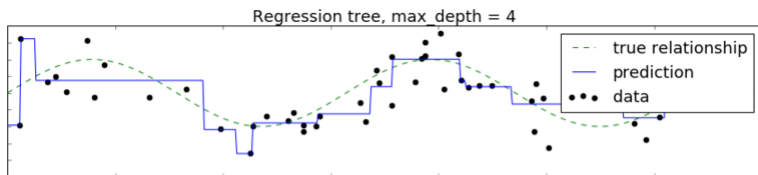
# Разбиение на классы (глубина=10)



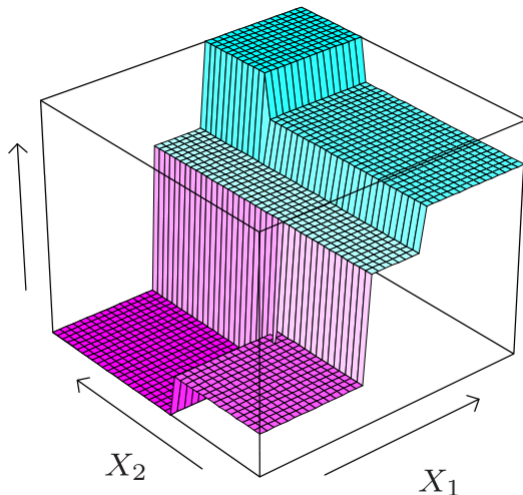
# CART для задачи регрессии (малая глубина)



# CART для задачи регрессии (большая глубина)



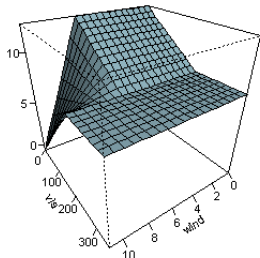
# Кусочно-постоянные прогнозы CART



# CART vs. MARS

MARS превращается в CART при

- решении задачи регрессии с критерием MSE
- $\left\{ (x_j - t)_+, (t - x_j)_- \right\} \rightarrow \{ \mathbb{I}[x_j \geq t], \mathbb{I}[x_j < t] \}$
- если добавляем слагаемое с произведением, то убираем первоначальное



# Анализ критерия ветвления CART

## Преимущества:

- интерпретируемость (визуализация)
- вычислительная простота прогнозирования
- отбор признаков
- работает для признаков разной природы
  - обрабатывает вещественные, упорядоченные и бинарные признаки
  - прогноз инвариантен к монотонным преобразованиям признака для  $Q_t(x) = x^{i(t)}$ :

$$x^{i(t)} \leq h \Leftrightarrow f\left(x^{i(t)}\right) \leq f(h) \quad \forall \uparrow f(\cdot)$$

## Недостатки:

- константные прогнозы в листьях
  - можно в листах ассоциировать  $\hat{y} = f_t(x)$ , а не константу  $\hat{y}_t$ .
- много вершин - для описания наклонных границ к осям

# Содержание

- 1 Понятие решающего дерева
- 2 Правила спуска
- 3 Выбор критерия ветвления**
- 4 Назначение прогнозов листьям
- 5 Критерий остановки



## Определение критерия ветвления

- Из узла  $t$  перейти в: 
$$\begin{cases} \text{левого потомка } t_L, & \text{если } x^{\hat{t}} \leq \hat{h}_t \\ \text{правого потомка } t_R, & \text{если } x^{\hat{t}} > \hat{h}_t \end{cases}$$

## Определение критерия ветвления

- Из узла  $t$  перейти в: 
$$\begin{cases} \text{левого потомка } t_L, & \text{если } x^{\hat{t}} \leq \hat{h}_t \\ \text{правого потомка } t_R, & \text{если } x^{\hat{t}} > \hat{h}_t \end{cases}$$
- Определим  $\phi(t)$  - ф-цию неопределенности (критерий информативности, impurity function)
  - измеряет степень неоднозначности  $u$  для объектов в узле  $t$ .

## Определение критерия ветвления

- Из узла  $t$  перейти в: 
$$\begin{cases} \text{левого потомка } t_L, & \text{если } x^{\hat{t}_t} \leq \hat{h}_t \\ \text{правого потомка } t_R, & \text{если } x^{\hat{t}_t} > \hat{h}_t \end{cases}$$
- Определим  $\phi(t)$  - ф-цию неопределенности (критерий информативности, impurity function)
  - измеряет степень неоднозначности  $y$  для объектов в узле  $t$ .
- Качество разбиения  $t$ :

$$\Delta\phi(t) = \phi(t) - \phi(t_L)\frac{N(t_L)}{N(t)} - \phi(t_R)\frac{N(t_R)}{N(t)}$$

## Определение критерия ветвления

- Из узла  $t$  перейти в: 
$$\begin{cases} \text{левого потомка } t_L, & \text{если } x^{\hat{i}_t} \leq \hat{h}_t \\ \text{правого потомка } t_R, & \text{если } x^{\hat{i}_t} > \hat{h}_t \end{cases}$$
- Определим  $\phi(t)$  - ф-цию неопределенности (критерий информативности, impurity function)
  - измеряет степень неоднозначности  $y$  для объектов в узле  $t$ .
- Качество разбиения  $t$ :

$$\Delta\phi(t) = \phi(t) - \phi(t_L)\frac{N(t_L)}{N(t)} - \phi(t_R)\frac{N(t_R)}{N(t)}$$

- Оптимизация CART (регрессия, классификация): выбрать признак  $x_i$  и порог  $h$ , максимизирующие  $\Delta\phi(t)$ :

$$\hat{i}_t, \hat{h}_t = \arg \max_{i,h} \Delta\phi(t)$$

## Функция информативности

- Регрессия:

- пусть  $I = \{i_1, \dots, i_K\}$  - множество индексов объектов узла  $t$ .  
Можно определить  $\phi(t)$  как

$$\phi(t) = \arg \min_{\hat{y}} \frac{1}{|I_t|} \sum_{i \in I_t} (y_i - \hat{y})^2$$

$$\phi(t) = \arg \min_{\hat{y}} \frac{1}{|I_t|} \sum_{i \in I_t} |y_i - \hat{y}|$$

## Функция информативности

- Регрессия:

- пусть  $I = \{i_1, \dots, i_K\}$  - множество индексов объектов узла  $t$ .  
Можно определить  $\phi(t)$  как

$$\phi(t) = \arg \min_{\hat{y}} \frac{1}{|I_t|} \sum_{i \in I_t} (y_i - \hat{y})^2 = \frac{1}{|I_t|} \sum_{i \in I_t} (y_i - \text{mean}(I_t))^2$$

$$\phi(t) = \arg \min_{\hat{y}} \frac{1}{|I_t|} \sum_{i \in I_t} |y_i - \hat{y}| = \frac{1}{|I_t|} \sum_{i \in I_t} |y_i - \text{median}(I_t)|,$$

# Функции информативности для классификации

- Классификация:

- $p_1, \dots, p_C$  - вероятности классов в узле  $t$ .

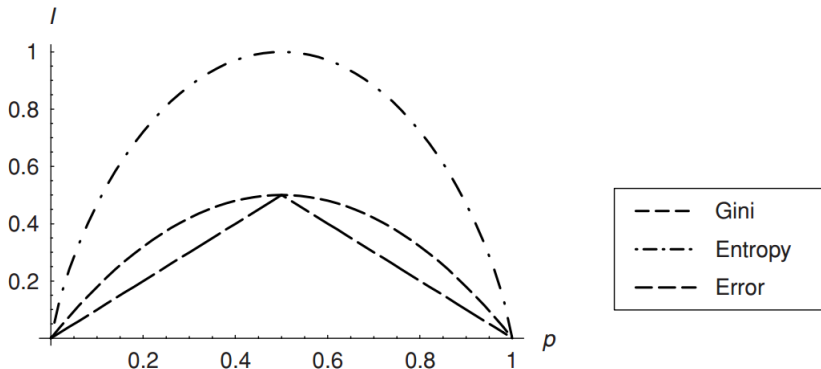
- Функция информативности  $\phi(t) = \phi(p_1, p_2, \dots, p_C)$  должна удовлетворять:

- $\phi$  определена для  $p_j \geq 0$  и  $\sum_j p_j = 1$ .
- $\phi$  достигает максимума при  $p_j = 1/C$ ,  $k = 1, 2, \dots, C$ .
- $\phi$  достигает минимума при  $\exists j : p_j = 1, p_i = 0 \forall i \neq j$ .
- $\phi$  симметрична относительно  $p_1, p_2, \dots, p_C$ .

## Визуализация основных функции информативности

Функции информативности для  $y \in \{+1, -1\}$  с

$p(y = +1|x) = p$  и  $p(y = -1|x) = 1 - p$ .





## Функция неопределённости: классификационная ошибка

- **Классификационная ошибка:** как часто ошибаемся при константном прогнозе?

$$\begin{aligned}\phi(t) &= \min_{\hat{y}} \frac{1}{|I_t|} \sum_{i \in I_t} \mathbb{I}[y_i \neq \hat{y}] \\ &= \frac{1}{|I_t|} \sum_{i \in I_t} \mathbb{I}[y_i \neq y_{most.common}] \\ &= 1 - \hat{p}_{max}\end{aligned}$$

## Функция неопределённости: критерий Джини

- Критерий Джини: оценка Бриера<sup>1</sup>

$$\begin{aligned}
 \phi(t) &= \min_{p: \sum_c p_c = 1} \frac{1}{|I_t|} \sum_{i \in I_t} \|p - p_i^{true}\|^2 = \\
 &= \min_{p: \sum_c p_c = 1} \frac{1}{|I_t|} \sum_{i \in I_t} \sum_{c=1}^C (p_c - \mathbb{I}[y_i = c])^2 = \\
 &= \sum_{i=1}^C \hat{p}_i (1 - \hat{p}_i) = 1 - \sum_{i=1}^C \hat{p}_i^2
 \end{aligned}$$

- Это вероятность ошибки при случайном угадывании с вероятностями  $p(\hat{y} = 1) = \hat{p}_1, \dots, p(\hat{y} = C) = \hat{p}_C$

---

<sup>1</sup> Докажите оптимальность выборочных оценок вероятностей классов и финальный вид критерия.

## Функция неопределённости: энтропия

- **Энтропия:** -логарифм правдоподобия оптимальных вероятностей классов<sup>2</sup>

$$\begin{aligned}\phi(t) &= \min_{p: \sum_c p_c = 1} -\frac{1}{|I_t|} \left( \sum_{i \in I_t} \sum_{c=1}^C \ln p_c^{\mathbb{I}[y_i=c]} \right) = \\ &= \min_{p: \sum_c p_c = 1} -\frac{1}{|I_t|} \left( \sum_{i \in I_t} \sum_{c=1}^C \mathbb{I}[y_i = c] \ln p_c \right) = -\sum_{i=1}^C \hat{p}_i \ln \hat{p}_i\end{aligned}$$

- Это среднее количество информации  $= -\ln p_y$ , которое получаем, узнав класс  $y$ .

---

<sup>2</sup> Докажите оптимальность выборочных оценок вероятностей классов и финальный вид критерия.

## Комментарии

- Алгоритм  $\hat{i}_t, \hat{h}_t = \arg \max_{i,h} \Delta \phi(t)$  применяется рекурсивно при построении дерева сверху вниз.
- жадный алгоритм, см. только на 1 шаг вперед (глобально неоптимальный, зато быстрый)
  - можно заглядывать на 2 шага вперед
- Сложность вычисления  $\phi(t)$ :  $O(N)$ ,  $O(N)$  значений порога,  $D$  признаков.
- Сложность настройки:  $O(N^2 D)$ . Как можно её сократить?

## Комментарии

- Алгоритм  $\hat{i}_t, \hat{h}_t = \arg \max_{i,h} \Delta \phi(t)$  применяется рекурсивно при построении дерева сверху вниз.
- жадный алгоритм, см. только на 1 шаг вперед (глобально неоптимальный, зато быстрый)
  - можно заглядывать на 2 шага вперед
- Сложность вычисления  $\phi(t)$ :  $O(N)$ ,  $O(N)$  значений порога,  $D$  признаков.
- Сложность настройки:  $O(N^2 D)$ . Как можно её сократить?
  - экономный пересчет  $\phi(t)$ 
    - при смещении порога 1 объект меняет вершину
  - дискретизация признака
    - квантили: 0.1, 0.2, ... 0.9

## Содержание

- 1 Понятие решающего дерева
- 2 Правила спуска
- 3 Выбор критерия ветвления
- 4 Назначение прогнозов листьям**
- 5 Критерий остановки

# Оптимальный прогноз в листьях: регрессия

- Регрессия:

$$\hat{y} = \arg \min_f \sum_{i: x_i \in t} \mathcal{L}(f - y_i)$$

- Например<sup>3</sup>

$$\mathcal{L}(u) = u^2 : \hat{y} = \text{mean}_{i: x_i \in t} \{y_i\}$$

$$\mathcal{L}(u) = |u| : \hat{y} = \text{median}_{i: x_i \in t} \{y_i\}$$

---

<sup>3</sup> Докажите оптимальность среднего и медианы для соответствующих ф-ций потерь.

## Оптимальный прогноз в листьях: классификация

В практических задачах классификации типы ошибок приводят к разным штрафам.

- например, при определении болен пациент или здоров.

Определим матрицу штрафов<sup>4</sup>  $\Lambda \in \mathbb{R}^{C \times C}$ , где

$\lambda_{ij} = \text{cost}(\hat{y} = j | y = i)$ :

		прогноз		
		$\hat{y} = 1$	$\dots$	$\hat{y} = C$
факт	$y = 1$	$\lambda_{11}$	$\dots$	$\lambda_{1C}$
	$\dots$	$\dots$	$\dots$	$\dots$
	$y = C$	$\lambda_{C1}$	$\dots$	$\lambda_{CC}$

<sup>4</sup>Как эта матрица будет выглядеть в случае единичных потерь за любой тип ошибки?



## Оптимальный прогноз в листьях: классификация

В случае общих потерь  $\lambda_{ij} = \text{cost}(\hat{y} = j | y = i)$

$$\hat{y} = \arg \min_j \sum_{i \in I_t} \lambda_{y(i),j} = \arg \min_j N_t \sum_{c=1}^C p_c \lambda_{cj}$$

В случае  $\lambda_{ij} = \lambda \mathbb{I}[i \neq j]$ :

$$\begin{aligned} \hat{y} &= \arg \min_j N_t \sum_{c=1}^C p_c \lambda \mathbb{I}[i \neq j] = \arg \min \sum_{c \neq j} p_c \\ &= \arg \min_j (1 - p_j) = \arg \max_j p_j \end{aligned}$$

# Содержание

- 1 Понятие решающего дерева
- 2 Правила спуска
- 3 Выбор критерия ветвления
- 4 Назначение прогнозов листьям
- 5 Критерий остановки
  - Остановка, основанная на правилах
  - Алгоритм обрезки в CART

## Критерий остановки

- Сложность модели должна соответствовать сложности данных:
  - слишком глубокие деревья -> переобучение
    - в крайнем случае: 1 лист содержит 1 объект, нет обобщающей способности.
  - слишком мелкие деревья -> недообучение
- Необходимо выбрать оптимальную глубину при построении дерева.
- Подходы к остановке построения:
  - основанные на правилах
  - обрезка деревьев (pruning)

## 5 Критерий остановки

- Остановка, основанная на правилах
- Алгоритм обрезки в CART

## Остановка, основанная на правилах

- Остановка, когда критерий больше порога.
- Варианты критерия:
  - глубина дерева
  - #объектов в узле
  - минимальное #объектов в дочерних узлах после разбиения
  - значение информативности  $\phi(t)$
  - изменение информативности  $\Delta\phi(t)$  после разбиения

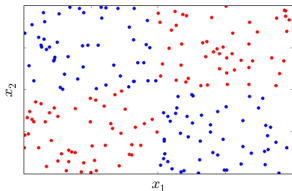
## Анализ подхода

Преимущества:

- простота
- интерпретируемость

Недостатки:

- нужно подбирать порог
- изменение информативности  $\Delta\phi(t)$  неоптимально:  
последующие разбиения могут привести к большему  $\Delta\phi(t)$ :



## 5 Критерий остановки

- Остановка, основанная на правилах
- Алгоритм обрезки в CART

# Алгоритм обрезки в CART

- Дерево строится до самого низа.
- Простой подход - обрезать снизу вверх по валидации.
- Алгоритм обрезки CART:
- Определим:
  - $T_t$  поддерево с корнем  $t$
  - $\tilde{T}, \tilde{T}_t$  - множество листьев  $T$  и  $T_t$
  - $R(T)$  - мера ошибок дерева  $T$  (#ошибок классификации / сумма квадратов ошибок)
  - $R_\alpha(T)$  - со штрафом за сложность ( $+\alpha$  за лист).

потери за ошибки :  $R(T) = \sum_{\tau \in \tilde{T}} R(\tau)$

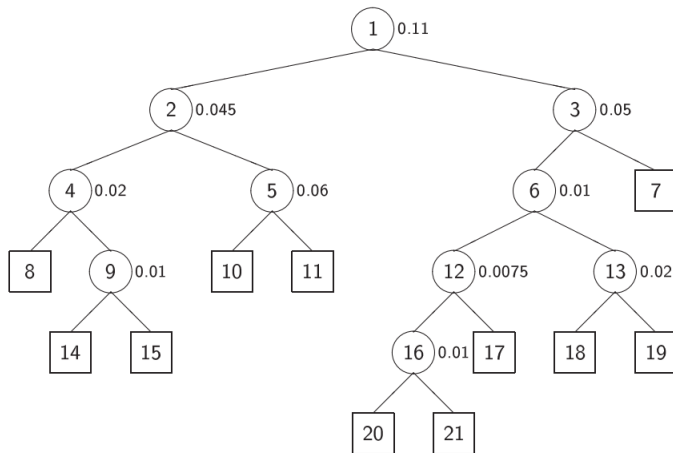
ошибки и сложность:  $R_\alpha(T) = \sum_{\tau \in \tilde{T}} R_\alpha(\tau) = R(T) + \alpha |\tilde{T}|$

- Целесообразность построения  $T_t$  вместо  $t$  - определим из условия  $R_{\alpha_t}(T_t) = R_{\alpha_t}(t)$ :

$$R(T_t) + \alpha |\tilde{T}_t| = R(t) + \alpha \Rightarrow \alpha_t = \frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1}$$



# Пример вычисления $\alpha_t$



## Алгоритм обрезки

- 1 Строим дерево  $T$  до самого низа (пока в листьях не останутся объекты с одинаковым откликом).
- 2 Строим систему вложенных поддеревьев  $T = T_0 \supset T_1 \supset \dots \supset T_{|T|}$  содержащих  $|T|, |T| - 1, \dots, 1$  узлов, повторяя процедуру:
  - заменить  $T_t$  с самым малым  $\alpha_t$  её корнем  $t$
  - пересчитать  $\alpha_t$  для всех предков  $t$ .
- 3 Для деревьев  $T_0, T_1, \dots, T_{|T|}$  рассчитаем их потери на валидации  $R(T_0), R(T_1), \dots, R(T_{|T|})$ .
- 4 Выберем  $T_i$ , достигающее минимального штрафа:

$$i = \arg \min_i R(T_i)$$

## Обработка пропущенных значений

Если проверяемый признак отсутствует:

- заполнить пропуски:
  - вещественные: средним, медианой
  - категориальные: модой или значением "пропущено"
- Можно предсказывать пропущенные значения по др. признакам (использовано в CART)
- C4.5: спускаемся из  $t$  по всем дочерним вершинам  $t_1, \dots, t_S$ , потом усредняем прогнозы с весами:

$$N(t_1)/N(t), N(t_2)/N(t), \dots N(t_S)/N(t)$$

## Важность признаков: mean decrease in impurity

- Важность признаков по изменению критерия информативности (mean decrease in impurity, MDI).

## Важность признаков: mean decrease in impurity

- Важность признаков по изменению критерия информативности (mean decrease in impurity, MDI).
  - рассмотрим признак  $f$
  - пусть  $T(f)$ -множество всех вершин, использующих  $f$  в функции ветвления
  - эффективность разбиения в  $t$ :

$$\Delta\phi(t) = \phi(t) - \sum_{c \in \text{children}(t)} \frac{N(c)}{N(t)} \phi(c)$$

- значимость  $f$ :

$$\sum_{t \in T(f)} N(t) \Delta\phi(t)$$

- Поощряет признаки с большим количеством уникальных значений.

## Важность признаков: mean decrease in impurity

В sklearn:

- важность рассчитывается метом `clf.feature_importances_`
- недостатки:
  - вычисляется на обучающей выборке
    - если модель переобучается на признаке, важность высока, но вклад в точность прогнозов мал.

## Преимущества решающих деревьев

Преимущества решающих деревьев:

- нелинейная модель с гибкой настройкой сложности
  - по глубине и др. критериям
- вычислительная эффективность прогнозов
- интерпретируемость (для неглубоких деревьев)
- встроенный обзор признаков
- инвариантны к масштабу признаков
- инвариантны к монотонным преобразованиям признаков
- обладают универсальной применимостью к признакам разной природы
  - бинарные, вещественные, порядковые категориальные
  - категориальные  $\rightarrow$  бинарные (one-hot) или вещественные (mean-value encoding)
  - категориальные  $\rightarrow$  порядковые, упорядочив категории по  $\bar{y}$  при условии категории
- позволяют вычислять важность признаков

## Недостатки решающих деревьев

### Недостатки решающих деревьев:

- сравнительно невысокая точность:
  - в силу "жадной настройки" сверху вниз (глобально неоптимальность)
    - ошибки вблизи наклонных границ между классами
    - ранняя остановка по правилу может рано останавливаться
  - точность повышается применением композиций решающих деревьев
- отсутствует динамичная подстройка под потоковые данные
- обобщение за пределы обучающей выборки константой