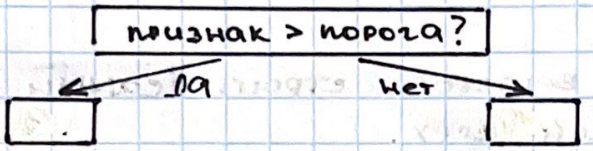
ЛЕКЦИЯ. РЕШАЮЩИЕ ДЕРЕВЬЯ

Решающие деревья - нелинейные модели, это деревья в математическом смысле (ориентированные графы). Как правило рассматриваются бинарные деревья: в вершине стоит некоторое условие, которое мы проверяем, и есть 2 ветки: одна если условие выполняется, другая - если нет, а в листьях дерева стоят предсказания

РЕШАЮЩЕЕ ДЕРЕВО - то бинарное дерево, в котором:

1) каждой вершине v приписана функция (предикат) $\beta_v : X \to \{0;1\}$



2) каждой листовой вершине v приписан прогноз $c_v \in Y$ (для классификации - класс или вероятность класса, для регрессии - действительное значение целевой переменной)

Минус дерева - легко переобучается.

Почти для любой выборки можно построить решающее дерево, не допускающее на ней ни одной ошибки (исключение - если есть одинаковые объекты с разными ответами). Такое дерево скорее всего будет переобученным.

Критерии информативности

В каждой вершине оптимизируем функционал $Q(x,j,t)$

Пусть R - множество объектов, попадающих в вершину на данном шаге, а R_l и R_r - объекты, попадающие в левую и правую ветки после разбиения.

Цель: хотим, чтобы после разбиения объектов на 2 группы внутри каждой группы было как можно больше объектов одного класса.

$H(R)$ – критерий информативности – мера неоднородности целевых (разнообразие) переменных внутри группы R.

Чем меньше разнообразие целевой переменной внутри группы, тем меньше значение $H(R)$. То есть хотим:

$$H(R_l) \to \min, \quad H(R_r) \to \min$$

$$Q(R, j, t) = H(R) - \left(\frac{|R_l|}{|R|} H(R_l) + \frac{|R_r|}{|R|} H(R_r) \right) \to \max_{j,t}$$

$H(R)$ в задаче регрессии:

Если в качестве функции потерь мы берём квадратичную ошибку, то

$$H(R) = \min_{c \in R} \frac{1}{|R|} \sum_{(x_i, y_i) \in R} (y_i - c)^2$$

Её минимум достигается при $c = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} y_i$, т.е.

в листе предсказывается среднее значение целевой переменной на объектах, попавших в лист.

Значит, информативность $H(R)$ в листе – это дисперсия целевой переменной (для объектов, попавших в этот лист). Чем меньше дисперсия, тем меньше разброс целевой переменной объектов, попавших в лист.

$H(R)$ в задаче классификации:

Решаем задачу классификации с K классами: $1, 2, \dots, K$.

Пусть p_k - доля объектов класса k, попавших в вершину:

$$p_k = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} [y_i = k]$$

Пусть k_* - самый представительный класс в данной вершине:

$$k_* = \arg\max_k p_k$$

Ошибка классификации:

$$H(R) = \min_{c \in} \frac{1}{|R|} \sum_{(x_i, y_i) \in R} [y_i \neq c]$$

Критерий Джини:

Будем в каждой вершине в качестве ответа выдавать не класс, а распределение вероятностей классов:

$c = (c_1, \dots, c_K)$, $\sum_i c_i = 1$

Качество распределения можно измерить с помощью критерия Бриера:

$$H(R) = \min_c \frac{1}{|R|} \sum_{(x_i, y_i) \in R} \sum_{k=1}^{K} (c_k - [y_i = k])^2 \quad (*)$$

Минимальное значение функционала $H(R)$ достигается на векторе, состоящем из долей классов: $c_* = (p_1, \dots, p_K)$

На векторе c_* функционал $(*)$ переписывается в виде:

$$H(R) = \sum_{k=1}^{K} p_k (1 - p_k) \text{ - критерий Джини}$$

Энтропийный критерий:

Запишем логарифм правдоподобия:

$$H(R) = \min_c \left(-\frac{1}{|R|} \sum_{(x_i, y_i) \in R} \sum_{k=1}^{K} [y_i = k] \log c_k \right) \quad (*)$$

На векторе $c_* (p_1, \dots, p_K)$ функционал $(*)$ записывается в виде:

$$H(R) = -\sum_{k=1}^{K} p_k \log p_k$$ — энтропия

Критерии останова

1) ограничение максимальной глубины дерева (max_depth)
2) ограничение минимального числа объектов в листьях (min_samples_leaf)
3) ограничение максимального числа листьев в дереве
4) останов в случае, если все объекты в листе одного класса
5) требование, что функционал качества при дроблении увеличивался как минимум на 5%

Стрижка дерева (pruning)

Алгоритм:

1) строится переобученное дерево (в каждом листе один объект)
2) производится оптимизация его структуры с целью уменьшения переобучения

Стрижка – альтернатива критериям останова.

Cost-complexety pruning:

Пусть $R(T)$ – ошибка на дереве T. Введём регуляризованный функционал:

$$R_\alpha(T) = R(T) + \alpha(T), \text{ где}$$

$|T|$ – количество вершин в дереве.

Тогда при построении дерева и оптимизации функционала $R_\alpha(T)$ дерево не будет иметь большое количество вершин, а потому, будет менее переобученным

Плюсы решающих деревьев:

1) чёткие правила классификации (интерпретируемые предикаты, например, "возраст > 25")
2) деревья решений легко визуализируются, то есть хорошо интерпретируются
3) быстро обучаются и выдают прогноз
4) малое число параметров

Минусы решающих деревьев:

1) очень чувствительны к шумам в данных, модель сильно меняется при небольшом изменении обучающей выборки
2) разделяющая граница имеет свои ограничения (состоит из гиперплоскостей)
3) необходимость борьбы с переобучением (стрижка или какой-либо из критериев останова)
4) проблема поиска оптимального дерева (NP-полная задача, поэтому на практике используется жадное построение дерева)