

Решающие деревья

Виктор Владимирович Китов

МГУ им.Ломоносова, ф-т ВМиК, кафедра ММП.

I семестр 2015 г.

Определение решающего дерева

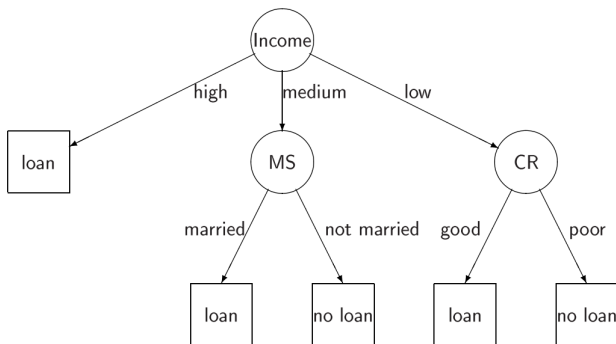
- Прогнозирование осуществляется деревом T :
 - направленный граф
 - без циклов
 - с одной корневой вершиной

Определение решающего дерева

- каждому узлу t соответствуют функции от признаков $Q_t(x^1, x^2, \dots, x^D)$
 - чаще всего проверяется значение отдельного признака $Q_t(x^1, x^2, \dots, x^D) = x^{i(t)}$
- исходящим ребрам $r_1(t), \dots, r_{K(t)}(t)$ соответствуют области значений $S_1(t), \dots, S_{K(t)}(t)$ для $Q_t(x^1, \dots, x^D)$, такие что:
 - $S_1(t), \dots, S_{K(t)}(t)$ покрывают все множество значений Q_t и $S_i \cap S_j = \emptyset \forall i \neq j, i, j \in \{r_1(t), \dots, r_{K(t)}(t)\}$.
 - чаще всего $K(t) = 2$, $S_1 = \{x^{i(t)} \leq \text{threshold}(t)\}$, $S_2 = \{x^{i(t)} > \text{threshold}(t)\}$.
 - варианты: $S_j = \{l_j < x \leq h_j\}$ либо $S = \{v_k\}$, где $\{v_1, v_2, \dots\}$ -множество уникальных значений $x^{i(t)}$.

Определение решающего дерева

- множество вершин делится на:
 - внутренние вершины $int(T)$, каждая из которых имеет ≥ 2 дочерних вершины
 - внешние вершины $terminal(T)$, которые не содержат потомков, но которым сопоставлены прогнозные значения.



Процесс прогнозирования

- Каждому листу сопоставлен прогноз некоторой константой:
 - классификация: номер класса
 - регрессия: значение.
- Процесс прогнозирования для дерева T :
 - $t = \text{root}(T)$
 - пока t не является листом:
 - рассчитать $Q_t(x)$
 - определить номер множества j , среди $S_1(t), \dots, S_{K(t)}(t)$, куда попадет $Q_t(x)$: $Q_t(x) \in S_j(t)$
 - перейти по ребру $r_j(t)$ к j -му потомку t :
 $t = (j\text{-й потомок } t)$.
 - вернуть прогноз, сопоставленный листу t .

Спецификация решающего дерева

- Определить решающее правило: $Q_t(x)$, $K(t)$ и $S_1(t), \dots, S_{K(t)}(t)$.
- Определить критерий останова (когда делаем вершину листом дерева).
- Определить сопоставление прогнозов каждому листу.

Алгоритм ID3

ПАРАМЕТРЫ:

Z —подмножество обучающих объектов $(x_1, y_1), \dots, (x_n, y_n)$

F —множество рассматриваемых признаков

ФУНКЦИЯ ID3(Z, F):

создать вершину дерева $root$

если все $y_i \in Z$ принадлежат одному классу c_k :

вернуть $root$ в качестве листа с классом c_k

если $F = \emptyset$:

вернуть $root$ в качестве листа с классом c_j ,
где c_j —самый частый класс среди Y .

иначе:

$f^* = \arg \max_f \text{InformationGain}_{f \in F}(Z, f)$

соотнести вершине $root$ признак f^*

для каждого значения v_i признака f^* :

создать исходящую ветвь $edge(v_i)$, которой
сопоставлено значение v_i

взять подмножества $Z(v_i) = \{Z_k, \text{ где } k \text{ такие, что } f_k^* = v_i\}$

если $Z(v_i) = \emptyset$, то

ветви $edge(v_i)$ соотнести лист с классом,
равным самому частому классу в Z .

иначе:

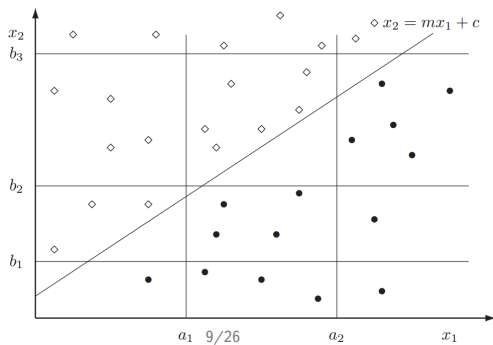
сопоставить ребру $e(v_i)$ вершину $ID3(Z(v_i), F - \{f^*\})$

Наиболее распространенное решающее правило

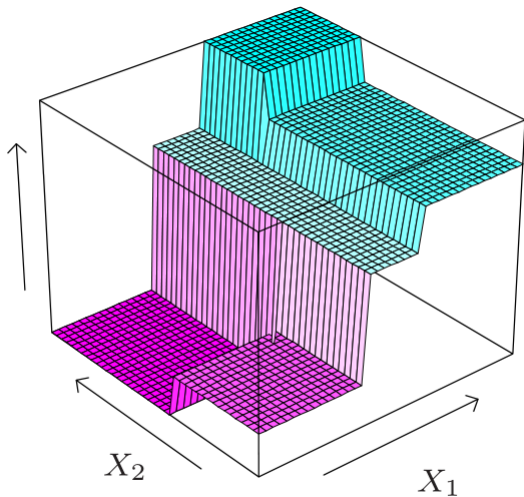
- Чаще всего:
 - в качестве Q_t берут $Q_t(x) = x^{i(t)}$
 - $K(t) = 2 \forall t \in \text{int}(T)$, где $\text{int}(T)$ - множество внутренних вершин дерева.
 - $S_1(t) = \{x^{i(t)} \leq \text{threshold}(t)\}$, $S_2(t) = \{x^{i(t)} > \text{threshold}(t)\}$
 - $\text{threshold}(t) \in \{x_1^{i(t)}, x_2^{i(t)}, \dots, x_N^{i(t)}\}$
 - применимо только для вещественных, порядковых и бинарных признаков
 - в случае дискретных признаков - их необходимо преобразовать в бинарные через one-hot-encoding.

Анализ решающего правила

- Преимущества:
 - простота
 - интерпретируемость
- Недостатки
 - много узлов может потребоваться, если разделяющая кривая не параллельна осям координат:

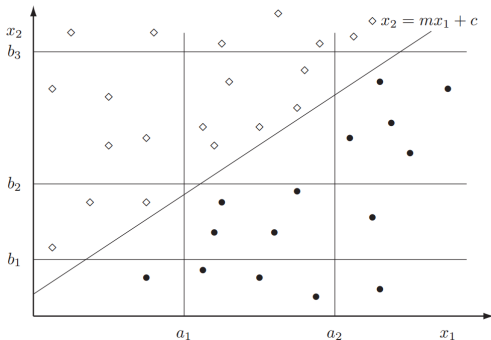


Кусочно-постоянное решение



Более общие решающие правила

- Вместо отдельного признака, $Q(x)$ может быть линейной $Q_t(x) = \langle a_t, x \rangle$ или нелинейной функцией.
 - тоже кусочно-постоянное решение, но с др. границами
 - менее интерпретируемое решение
 - зато может оказаться значительно меньше вершин!



Критерий останова

- Противоречие: смещение/дисперсия
 - наиболее разросшиеся деревья - переобучение
 - слишком простые - недообучение
- Подходы к остановке
 - основанные на правиле: сравниваем критерий с порогом
 - строим до самого низа, а потом обрезаем лишнее (pruning)

Правильный подход к остановке

- Делаем вершину листом, если критерий больше или меньше порога.
- Варианты критерия:
 - глубина дерева
 - количество наблюдений в вершине
 - минимальное количество наблюдений в одной из дочерних вершин
 - разделимость классов
 - изменение разделимости классов после дробления

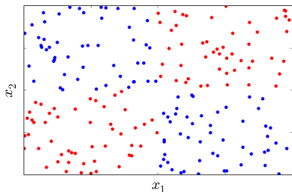
Обсуждение правилых подходов

Преимущества:

- простота
- эффективность

Недостатки:

- требует спецификации порога
- субоптимальна: последующие дробления могут оказаться лучше, но процесс построения дерева до них не дойдет
 - пример субоптимальности правил (основанных на делимости классов):



Функция смешанности классов

- Пусть $u(t)$ -область точек, которые попадают в узел t .
Вероятности классов в узле t :

$$p(\omega_j | x \in u(t)) = p(\omega_j | t) \approx \frac{N_j(t)}{N(t)}$$

- Функция смешанности (impurity function):

$$I(t) = \phi(p(\omega_1 | t), \dots, p(\omega_C | t))$$

- $\phi(q_1, q_2, \dots, q_C)$ определена для $q_j \geq 0$ и $\sum_j q_j = 1$.
- ϕ достигает максимума при $q_j = 1/C$ для всех j .
- ϕ достигает минимума, когда $\exists j : q_j = 1, q_i = 0$ для всех $i \neq j$.
- ϕ симметричная функция от q_1, q_2, \dots, q_C .

Типичные функции смешанности классов

- Критерий Джини

- вероятность сделать ошибку, сопоставляя класс случайно с вероятностями $[p(\omega_1|t), \dots, p(\omega_C|t)]$

$$I(t) = \sum_i p(\omega_i|t)(1 - p(\omega_i|t)) = 1 - \sum_i [p(\omega_i|t)]^2$$

- Энтропия

- мера неопределенности дискретной случайной величины

$$I(t) = - \sum_i p(\omega_i|t) \ln p(\omega_i|t)$$

- Ошибка классификации

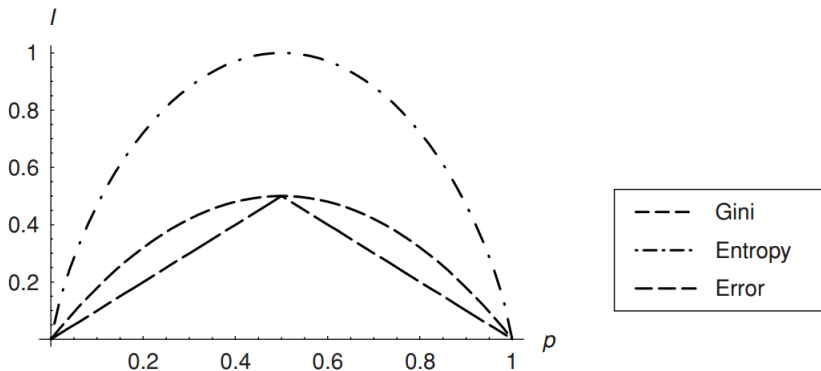
- показывает частоту ошибок при классификации самым частым классом

$$I(t) = 1 - \max_i p(\omega_i|t)$$

Типичные функции смешанности классов

Функции смешанности в случае 2х классов:

$$p = p(\omega_1|t), \quad 1 - p = p(\omega_2|t).$$



Критерий ветвления

- Выбрать правило ветвления, максимизирующее снижение смешанности классов:

$$\Delta I(t) = I(t) - \sum_{i=1}^S I(t_i) \frac{N_j(t)}{N(t)}$$

где S - число ветвей, t_1, \dots, t_S дочерние вершины для t , $N(t)$ - число наблюдений в t , а $N_j(t)$ - число наблюдений класса ω_j в t .

- Если $I(t)$ - энтропия, то $\Delta I(t)$ называется *information gain*.

Назначение классов листовым вершинам

- Регрессия:

$$\hat{y} = \arg \min_{\mu} \sum_{i: x_i \in u(t)} (y_i - \mu)^2 = \frac{1}{N_t} \sum_{i: x_i \in u(t)} y_i,$$

где $N = |\{x_i : x_i \in u(t)\}|$

- могут использоваться и др. функции потерь.
- Классификация:
 - Пусть $\lambda(y, \hat{y})$ - цена сопоставления объекта класса y классу \hat{y}
 - Сопоставление класса с минимальной совокупной ценой

$$\hat{y} = \arg \min_y \sum_{i: x_i \in u(t)} \lambda(y_i, y)$$

- Для $\lambda(y, \hat{y}) = \mathbb{I}[y \neq \hat{y}]$ будет сопоставляться наиболее частый класс в подвыборке листа.

Обозначения в методе CART

- Пусть T -дерево, а \tilde{T} -множество листовых вершин для T
- Определим $R(t) = \frac{M(t)}{N}$, для $t \in \tilde{T}$.
 - $M(t)$ - количество ошибок на валидационном множестве
 - N -количество элементов на валидационном множестве.
- Определим частоту ошибок на валидационном множестве

$$R(T) = \sum_{t \in \tilde{T}} R(t),$$

- и цену дерева T :

$$R_\alpha(T) = \sum_{t \in \tilde{T}} R_\alpha(t) = R(T) + \alpha |\tilde{T}|,$$

- Условие, когда дерево T_t с корнем t имеет такую же цену, как сам корень t :

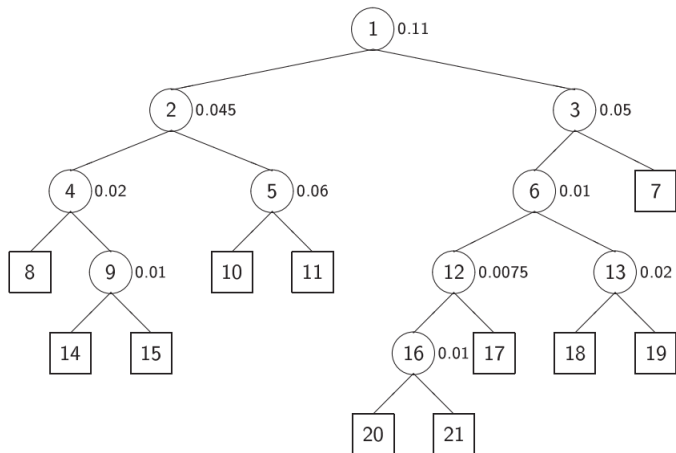
$$R_\alpha(t) = R_\alpha(T_t) \Rightarrow \alpha_t = \frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1}$$

Обрезка дерева в методе CART

- строим дерево, пока классы не становятся полностью разделимыми, получаем дерево T .
- строим иерархию вложенных поддеревьев T_0, T_1, \dots, T_K , повторяя процедуру пока не останемся только с корнем T :
 - заменяем дерево с наименьшим α_t его корнем t
 - пересчитываем $R(T_t)$, $R_\alpha(T_t)$ и α_t для всех предков t .
- Для каждого вложенного дерева $T_0, T_1, \dots, T_{|T|}$ вычисляем ф-ции цены $R(T_0), R(T_1), \dots, R(T_K)$.
- Выбираем дерево T_i , дающее наименьшую цену

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

Пример



Пример

	α_k	$ \tilde{T}^k $	$R(T^k)$
1	0	11	0.185
2	0.0075	9	0.2
3	0.01	6	0.22
4	0.02	5	0.25
5	0.045	3	0.34
6	0.05	2	0.39
7	0.11	1	0.5

Работа с пропущенными признаками

Если нужный признак отсутствует:

- усреднение прогнозов по каждой ветви с весами \propto числу наблюдений из TS , идущих в каждую ветвь.
- surrogate splits
- заполнение пропущенных значений
 - среднее
 - прогноз по остальным признакам
- отдельная ветвь «значение пропущено».

Важность признаков, рассчитанная по дереву

Важность признака x^d для предсказания y можно считать:

- по доле наблюдений, охватываемых правилом с участием x^d в дереве
- по тому, насколько правила с x^d улучшили разделимость классов
 - с учетом доли охвата правил с участием x^d

Анализ решающих деревьев в целом

- Преимущества решающих деревьев:
 - простота
 - интерпретируемость
 - встроенный отбор признаков
 - работает одновременно с дискретными и непрерывными признаками
 - прогноз инвариантен к монотонным преобразованиям признаков для $Q_t(x) = x^{i(t)}$
 - не нужна нормализация
- Недостатки решающих деревьев:
 - если разделяющая кривая не параллельна осям координат, то может потребоваться много вершин
 - субоптимальность (пример XOR)
 - нет онлайнности - для новых наблюдений требуется полная перестройка всего дерева.