

Разбор летучки

Лекция 4

Деревья принятия решений

Екатерина Тузова

Мотивирующий пример

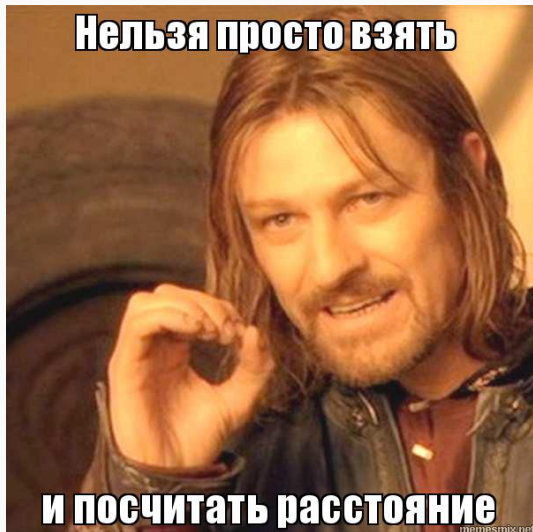
Мотивирующий пример



```
In [10]: pokemons.head()
```

```
Out[10]:
```

	Type_1	Type_2	isLegendary	Color	hasGender	Egg_Group_1	Egg_Group_2	hasMegaEvolution	Body_Style
0	Grass	Poison	False	Green	True	Monster	Grass	False	quadruped
1	Grass	Poison	False	Green	True	Monster	Grass	False	quadruped
2	Grass	Poison	False	Green	True	Monster	Grass	True	quadruped
3	Fire	NaN	False	Red	True	Monster	Dragon	False	bipedal_tailed
4	Fire	NaN	False	Red	True	Monster	Dragon	False	bipedal_tailed



$X^l = (x_i, y_i)_{i=1}^l$ - обучающая выборка.

Логическая закономерность – предикат $\beta : X \rightarrow \{0, 1\}$, который удовлетворяет двум требованиям:

$X^l = (x_i, y_i)_{i=1}^l$ - обучающая выборка.

Логическая закономерность – предикат $\beta : X \rightarrow \{0, 1\}$, который удовлетворяет двум требованиям:

1. Интерпретируемость

$X^l = (x_i, y_i)_{i=1}^l$ - обучающая выборка.

Логическая закономерность – предикат $\beta : X \rightarrow \{0, 1\}$, который удовлетворяет двум требованиям:

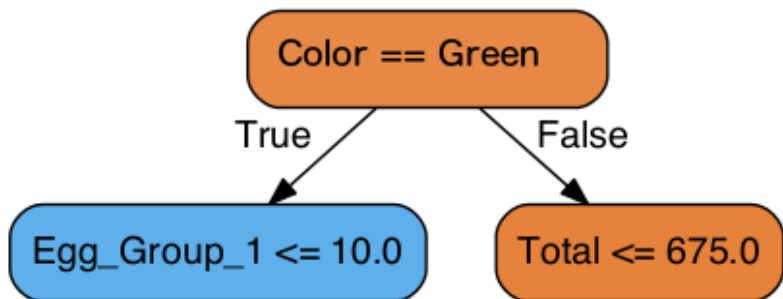
1. Интерпретируемость
2. Информативность относительно одного из классов $c \in Y$

$X^l = (x_i, y_i)_{i=1}^l$ – обучающая выборка.

Предикат $\beta : X \rightarrow \{0, 1\}$

Задача: Найти множество логических закономерностей \mathcal{B} по X^l .
Построить алгоритм $a(X, \mathcal{B}) \rightarrow y$, способный классифицировать произвольный объект $x \in X$.

1. Записывается на естественном языке
2. Зависит от небольшого числа признаков



Идея: Максимизировать количество правильно распознанных объектов класса c и при этом минимизировать количество объектов, ошибочно классифицированных как класс c

Идея: Максимизировать количество правильно распознанных объектов класса c и при этом минимизировать количество объектов, ошибочно классифицированных как класс c

$$tp(\beta) = \# \{x_i : \beta(x_i) = 1, y_i = c\} \rightarrow \max$$

Идея: Максимизировать количество правильно распознанных объектов класса c и при этом минимизировать количество объектов, ошибочно классифицированных как класс c

$$tp(\beta) = \# \{x_i : \beta(x_i) = 1, y_i = c\} \rightarrow \max$$

$$fp(\beta) = \# \{x_i : \beta(x_i) = 1, y_i \neq c\} \rightarrow \min$$

1. Какого вида закономерности $\beta(x)$ нужны?
2. Как определять информативность?
3. Как выбирать закономерности?
4. Как объединять закономерности в алгоритм?

Виды правил

– Пороговое условие

$$\beta(x) = [x^j \leq a_j] \text{ или } [a_j \leq x^j \leq b_j]$$

- Пороговое условие

$$\beta(x) = [x^j \leq a_j] \text{ или } [a_j \leq x^j \leq b_j]$$

- Конъюнкция из J пороговых условий

$$\beta(x) = \bigwedge_{j \in J} [a_j \leq x^j \leq b_j]$$

- Пороговое условие

$$\beta(x) = [x^j \leq a_j] \text{ или } [a_j \leq x^j \leq b_j]$$

- Конъюнкция из J пороговых условий

$$\beta(x) = \bigwedge_{j \in J} [a_j \leq x^j \leq b_j]$$

- Синдром – выполнение не менее d условий из J

$$\beta(x) = \left[\sum_{j \in J} [a_j \leq x^j \leq b_j] \geq d \right]$$

Как собрать классификатор
из закономерностей?

Идея:

Возьмем $\beta_1(x), \beta_2(x), \dots, \beta_T(x)$ закономерностей и будем по порядку применять на объекте. Как только предикат β_i сработал – вернем соответствующий класс c_i .

Идея:

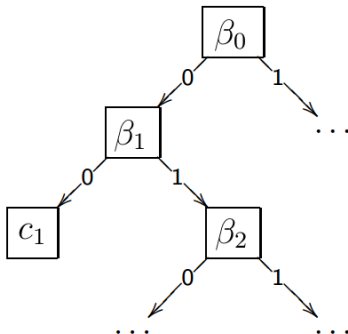
Возьмем $\beta_1(x), \beta_2(x), \dots, \beta_T(x)$ закономерностей и будем по порядку применять на объекте. Как только предикат β_i сработал – вернем соответствующий класс c_i .

Каждое правило принимает окончательное решение \Rightarrow ошибка правила равна ошибке всего алгоритма

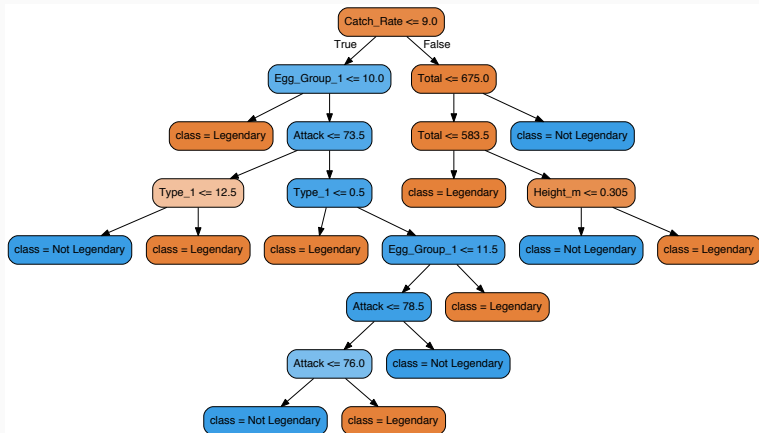
Бинарное решающее дерево

Бинарное решающее дерево – алгоритм классификации $a(x, \beta)$, задающийся бинарным деревом:

- $\forall v \in V_{inner} \rightarrow \beta_v : X \rightarrow \{0, 1\}, \beta \in \mathcal{B}$
- $\forall v \in V_{leaf} \rightarrow$ имя класса $c_v \in Y$



Пример решающего дерева



Алгоритм построения ID3

```
1 function LEARNID3( $U, \mathcal{B}$ )
2   if все объекты из  $U$  лежат в одном классе  $c \in Y$  then
3     return новый лист  $v$ ,  $c_v = c$ 
4    $\beta^* = \max_{\beta \in \mathcal{B}} I(\beta, U)$ 
5    $U_{left} = \{x \in U : \beta^*(x) = 0\}$ 
6    $U_{right} = \{x \in U : \beta^*(x) = 1\}$ 
7   if  $U_{left} = \emptyset$  или  $U_{right} = \emptyset$  then
8     return  $v$ ,  $c_v = \text{Majority}(U)$ 
9   Создать новую внутреннюю вершину  $v$ :  $\beta_v = \beta^*$ 
10   $L_v = \text{LearnID3}(U_{left}, \mathcal{B})$ 
11   $R_v = \text{LearnID3}(U_{right}, \mathcal{B})$ 
12  return  $v$ 
```

Критерии информативности

$$tp(\beta) = \# \{x_i : \beta(x_i) = 1, y_i = c\} \rightarrow \max$$

$$tn(\beta) = \# \{x_i : \beta(x_i) = 0, y_i \neq c\} \rightarrow \max$$

$$fp(\beta) = \# \{x_i : \beta(x_i) = 1, y_i \neq c\} \rightarrow \min$$

$$fn(\beta) = \# \{x_i : \beta(x_i) = 0, y_i = c\} \rightarrow \min$$

$$tp(\beta) = \# \{x_i : \beta(x_i) = 1, y_i = c\} \rightarrow \max$$

$$tn(\beta) = \# \{x_i : \beta(x_i) = 0, y_i \neq c\} \rightarrow \max$$

$$fp(\beta) = \# \{x_i : \beta(x_i) = 1, y_i \neq c\} \rightarrow \min$$

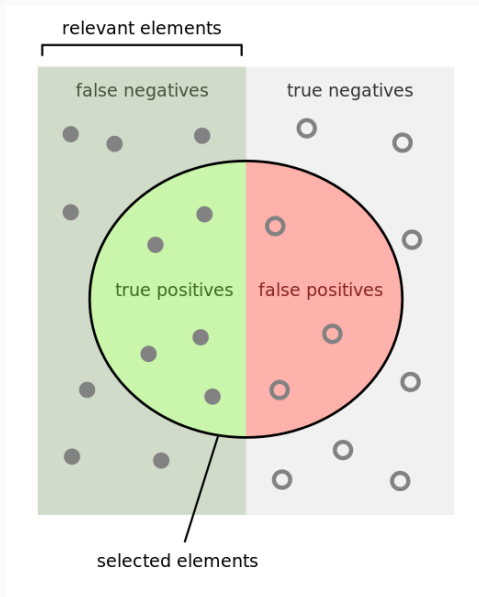
$$fn(\beta) = \# \{x_i : \beta(x_i) = 0, y_i = c\} \rightarrow \min$$

$$Precision = \frac{tp}{tp+fp}$$

$$Recall = TPR = \frac{tp}{tp+fn}$$

$$FPR = \frac{fp}{fp+tn}$$

Precision-Recall



Пример свертки двух критериев

Пусть число примеров искомого класса 200 и число остальных объектов 100

tp	fp	$tp - fp$	$tp - 5fp$	$Precision$
50	0	50	50	1
100	50	50	-150	0.6
50	9	41	5	0.84
5	0	5	5	1

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

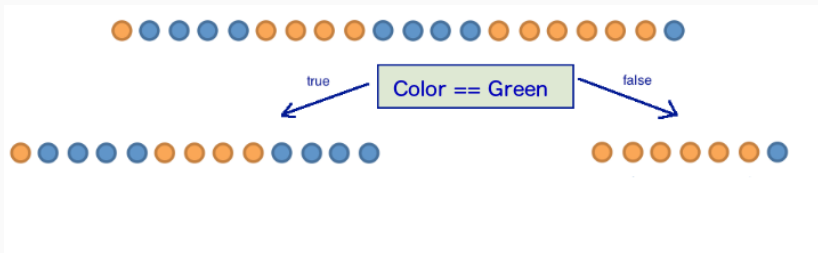
$$H(U) = - \sum_{i=1}^C p_i \log_2 p_i$$

p_i – процентное соотношение объектов класса i в выборке U

Энтропия Шеннона

$$H(U) = - \sum_{i=1}^C p_i \log_2 p_i$$

p_i – процентное соотношение объектов класса i в выборке U



Прирост информации – уменьшение энтропии.

$$H = - \sum_{i=1}^C p_i \log_2 p_i$$

$$IGain(U, x^j) = H(U) - \sum_v \frac{|U_v|}{|U|} H(U_v)$$

$$v \in values(x^j) \quad U_v = \{x \in U | x^j = v\}$$

+ Интерпретируемость и простота классификации

- + Интерпретируемость и простота классификации
- + Допустимы разнотипные данные и данные с пропусками

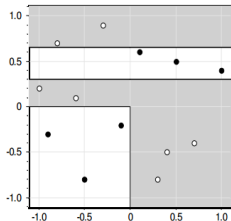
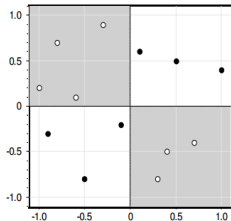
- + Интерпретируемость и простота классификации
- + Допустимы разнотипные данные и данные с пропусками
- + Не бывает отказов от классификации

- + Интерпретируемость и простота классификации
- + Допустимы разнотипные данные и данные с пропусками
- + Не бывает отказов от классификации
- + Трудоёмкость линейна по длине выборки

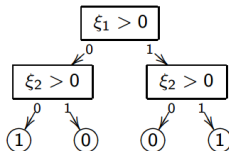
- Жадный ID3 сильно переобучается

- Жадный ID3 сильно переобучается
- Высокая чувствительность к шуму, к составу выборки, к критерию информативности

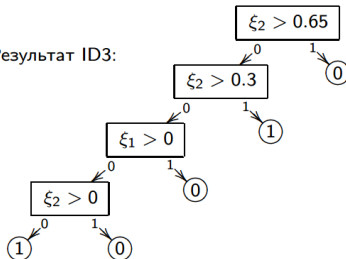
- Жадный ID3 сильно переобучается
- Высокая чувствительность к шуму, к составу выборки, к критерию информативности
- Чем дальше v от корня, тем меньше надёжность выбора β_v , c_v



Оптимальное дерево для задачи XOR:



Результат ID3:



Подрезание дерева C4.5

X^k – независимая контрольная выборка, $k \approx 0.5l$

Для всех $v \in V_{inner}$:

U_v = подмножество объектов X^k , дошедших до v

Если $U_v = \emptyset$:

Вернуть новый лист v , $c_v = \text{Majority}(U)$

Вычислить число ошибок четырьмя способами:

$r(v)$ – поддеревом, растущим из вершины v

$r_L(v)$ – левой дочерней вершины L_v

$r_R(v)$ – правой дочерней вершины R_v

$r_c(v)$ – к классу $c \in Y$

В зависимости от того, какое из них минимально:

Сохранить поддерево v

Заменить поддерево v поддеревом L_v

Заменить поддерево v поддеревом R_v

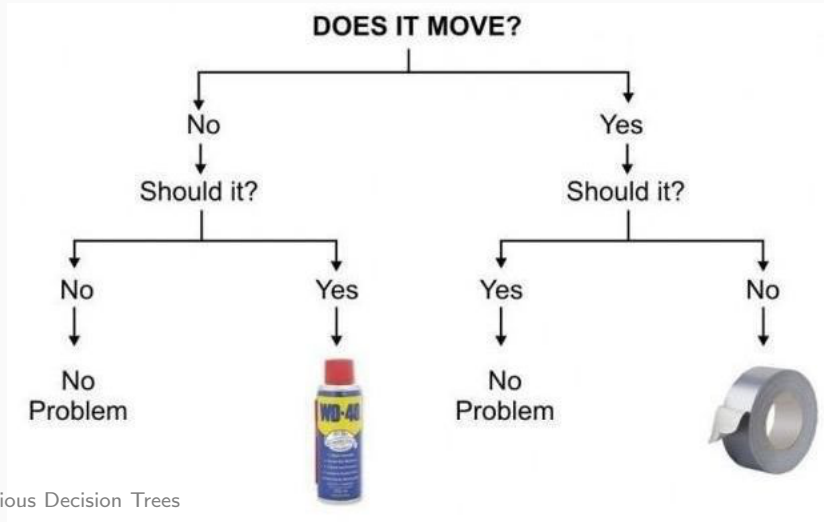
Заменить v листом $c_v = \min_{c \in Y} r_c(v)$

Небрежные решающие деревья

Идея: Сделаем дерево сбалансированным. Для этого нужно для всех узлов одного уровня использовать одинаковое условие ветвления.

Небрежные решающие деревья

Идея: Сделаем дерево сбалансированным. Для этого нужно для всех узлов одного уровня использовать одинаковое условие ветвления.



Идея: Можно использовать результаты нескольких алгоритмов, а не одного.

Голосование деревьев классификации, $Y = \{-1, +1\}$

$$a(t) = \text{Majority}(b_t(x))$$

- Каждое дерево $b_t(x)$ обучается по случайной выборке с повторениями
- В каждой вершине предикат выбирается из случайного подмножества n предикатов

Вопросы?

Что почитать по этой лекции

- G. James, D. Witten, T. Hastie, R. Tibshirani "An Introduction to Statistical Learning" Chapter 8
- Воронцов "Логические алгоритмы классификации"

На следующей лекции

- Байесовские методы классификации
- Вероятностная постановка задачи
- Оптимальный Байесов классификатор
- Наивность
- Максимальное правдоподобие
- Разные распределения