



Машинное обучение: деревья решений и ансамбли

Эмили Драль
СВР, Москва 2020

Программа курса

Курс состоит из **5ти** блоков:

1. **Базовые** концепции машинного обучения
2. **Линейные** модели классификации и регрессии
3. **Деревья решений** в классификации и регрессии, **ансамбли моделей**
4. Обучение **без учителя** и частичное обучение
5. **Нейронные сети** и глубокое обучение, backpropagation, регуляризация и методы оптимизации

Деревья решений и ансамбли

1. Деревья решений
2. Ансамбли моделей
3. Случайный лес
4. Градиентный бустинг

Базовые концепты

Базовые концепты

Объекты и признаки:

- x – объект
- y – ответ
- $(f_1, f_2 \dots f_n)$ – признаки, описывающие объекты
- $F^{(l,n)}$ – матрица объект-признак
- X – пространство объектов
- Y – пространство ответов

Модель:

- $a: X \rightarrow Y$
- $a(x) = y$
- A – семейство моделей

Оценка качества

- $Q(a, X)$ – ошибки модели $a(x)$ на группе объектов X

Как построить модель?

1. Подготовить набор данных $X = (x_i, y_i)_{i=1, l}$
2. Выбрать семейство моделей A
3. Минимизировать ошибки модели $Q(a, X) \rightarrow$
за счет этого получить конкретную модель
 $a(x)$ из выбранного семейства A

Базовые концепты

Как построить модель?

1. Подготовить набор данных $X = (x_i, y_i)_{i=1, l}$
2. Выбрать семейство моделей A
3. Минимизировать ошибки модели $Q(a, X)$:
 - 3.1 выбрать гиперпараметры модели с помощью кросс-валидации
 - 3.2 зная гиперпараметры, подобрать параметры модели в результате минимизации $Q(a, X)$ на всей обучающей выборке

Базовые концепты

Как построить модель?

1. Подготовить набор данных $X = (x_i, y_i)_{i=1, l}$
2. Выбрать семейство моделей A
3. Минимизировать ошибки модели $Q(a, X)$:
 - 3.1 выбрать гиперпараметры модели $L(a, y)$, I_1 vs I_2 с помощью кросс-валидации
 - 3.2 зная гиперпараметры, подобрать параметры модели w_1, w_2, \dots, w_n в результате минимизации $Q(a, X)$ на всей обучающей выборке

Деревья решений

Простейшая выборка

Рассмотрим выборку объектов с одним признаком x :



Как подобрать порог по признаку в задаче бинарной классификации?

Простейшая выборка

Рассмотрим выборку объектов с одним признаком x :

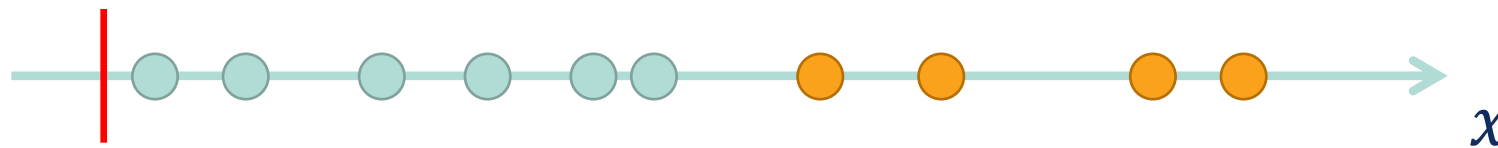


Как подобрать порог по признаку в задаче бинарной классификации?

Можно перебрать пороги, например, сдвигая на один пример.

Простейшая выборка

Рассмотрим выборку объектов с одним признаком x :

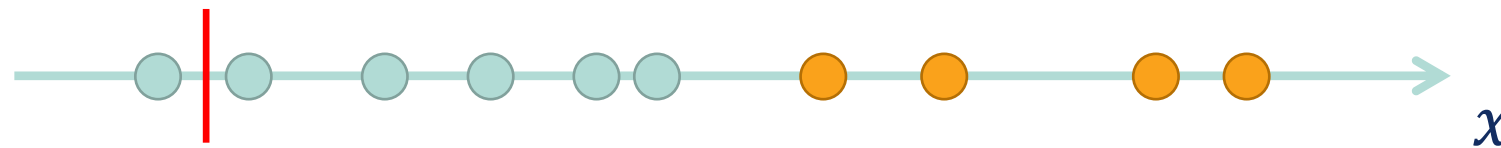


Как подобрать порог по признаку в задаче бинарной классификации?

Можно перебрать пороги, например, сдвигая на один пример.

Простейшая выборка

Рассмотрим выборку объектов с одним признаком x :

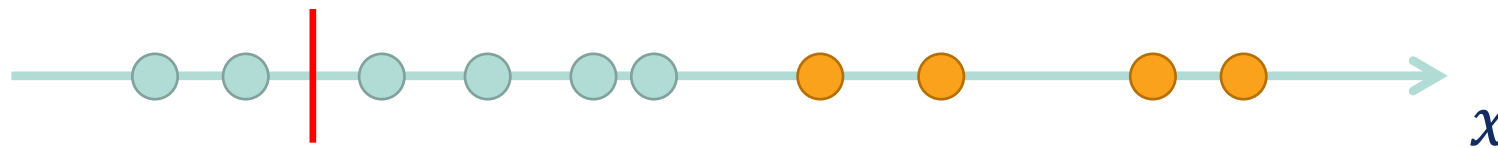


Как подобрать порог по признаку в задаче бинарной классификации?

Можно перебрать пороги, например, сдвигая на один пример.

Простейшая выборка

Рассмотрим выборку объектов с одним признаком x :

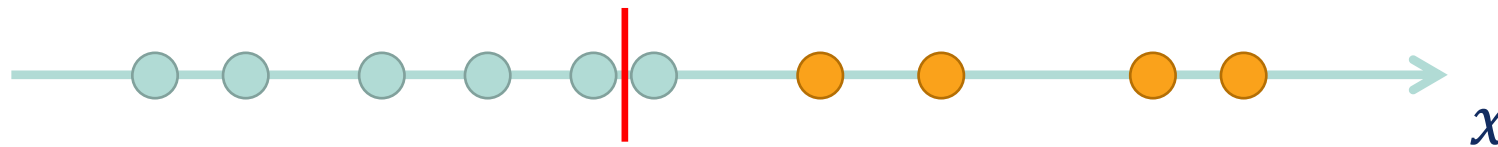


Как подобрать порог по признаку в задаче бинарной классификации?

Можно перебрать пороги, например, сдвигая на один пример.

Простейшая выборка

Рассмотрим выборку объектов с одним признаком x :

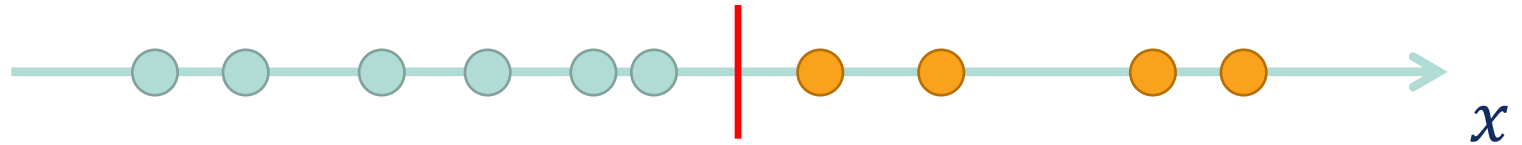


Как подобрать порог по признаку в задаче бинарной классификации?

Можно перебрать пороги, например, сдвигая на один пример.

Простейшая выборка

Рассмотрим выборку объектов с одним признаком x :

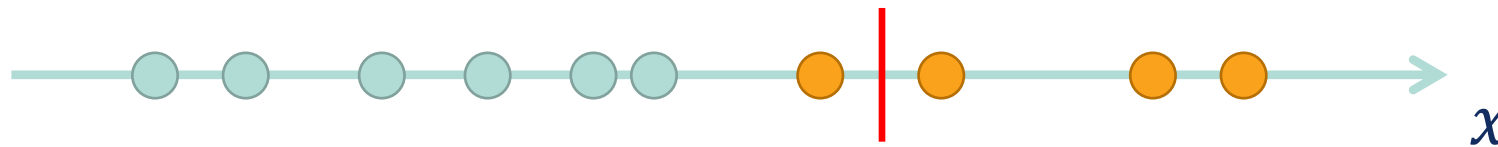


Как подобрать порог по признаку в задаче бинарной классификации?

Можно перебрать пороги, например, сдвигая на один пример.

Простейшая выборка

Рассмотрим выборку объектов с одним признаком x :

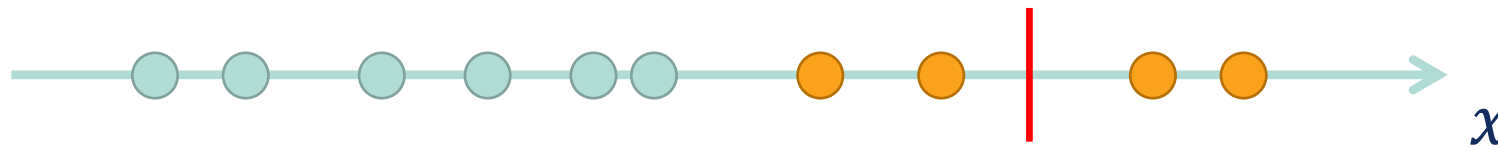


Как подобрать порог по признаку в задаче бинарной классификации?

Можно перебрать пороги, например, сдвигая на один пример.

Простейшая выборка

Рассмотрим выборку объектов с одним признаком x :



Как подобрать порог по признаку в задаче бинарной классификации?

Можно перебрать пороги, например, сдвигая на один пример.

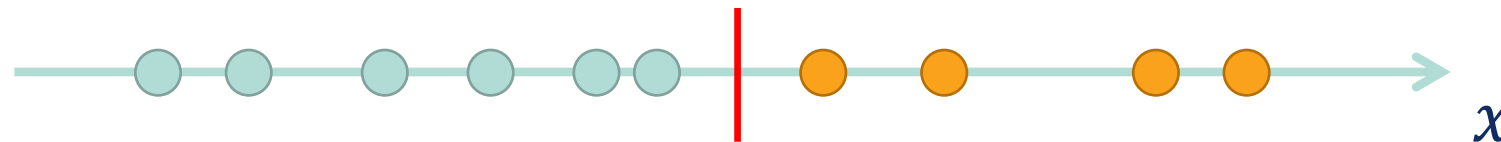
Простейшая выборка

Рассмотрим выборку объектов с одним признаком x :



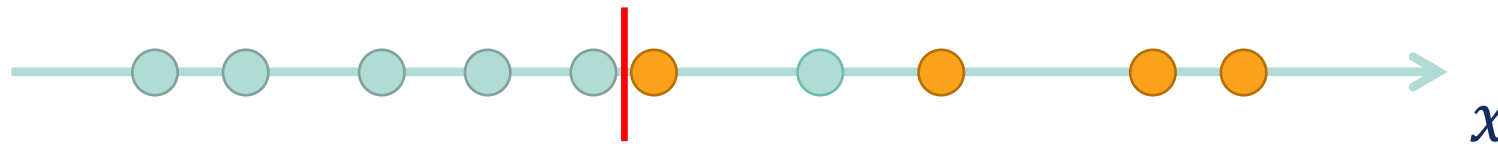
Как подобрать порог по признаку в задаче бинарной классификации?

Если выборка разделима, оптимальный порог - между последним объектом одного класса и первым объектом:



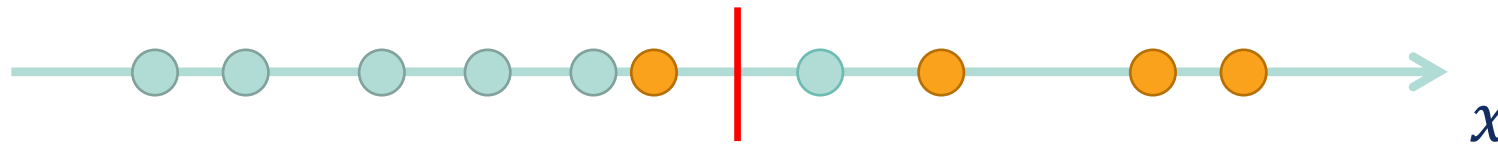
Простейшая выборка

Часто выборка не разделима и есть несколько неплохих порогов:



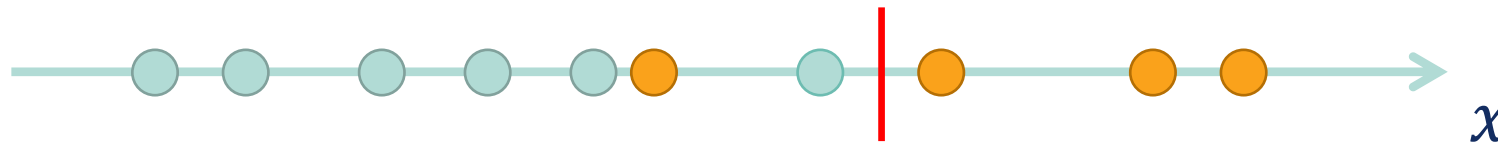
Простейшая выборка

Часто выборка не разделима и есть несколько неплохих порогов:



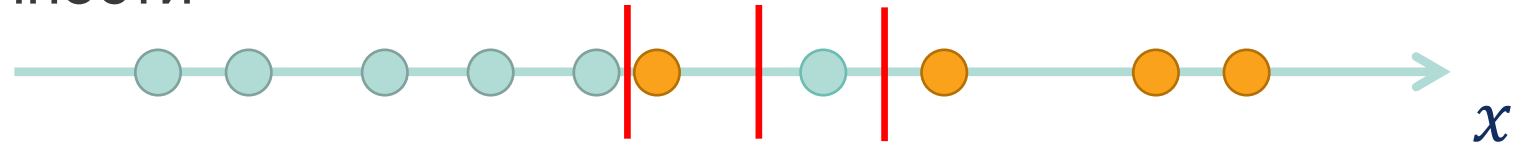
Простейшая выборка

Часто выборка не разделима и есть несколько неплохих порогов:



Простейшая выборка

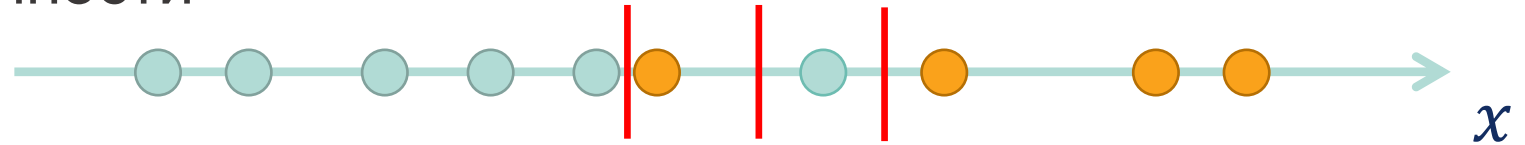
Вариант 1: потребовать от модели максимальной точности



и не ограничивать количество порогов, чтобы разделить выборку идеально

Простейшая выборка

Вариант 1: потребовать от модели максимальной точности

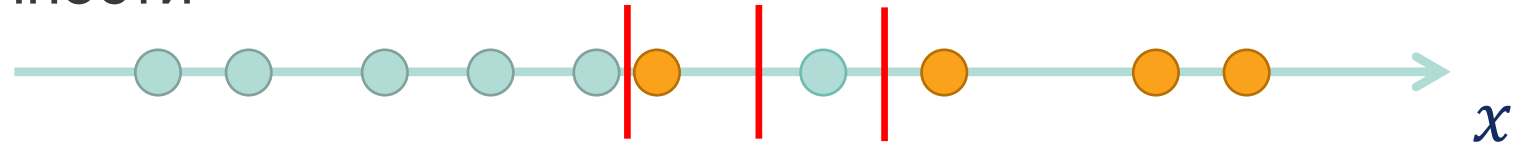


и не ограничивать количество порогов, чтобы разделить выборку идеально

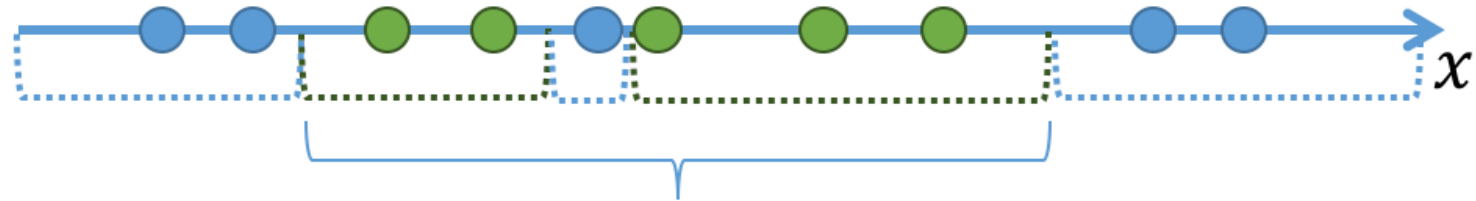
Проблема: запоминание выборки вместо обучения

Простейшая выборка

Вариант 1: потребовать от модели максимальной точности

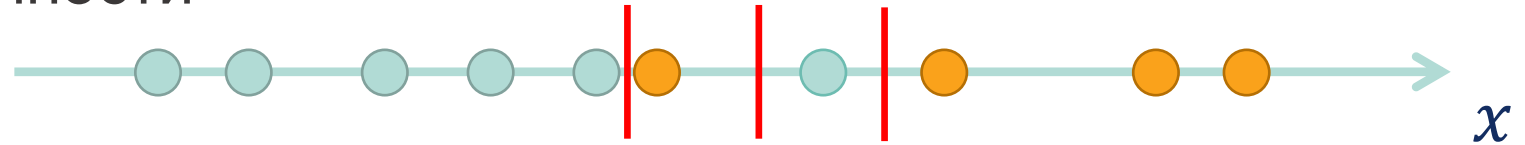


Вариант 2: разрешить объединение интервалов

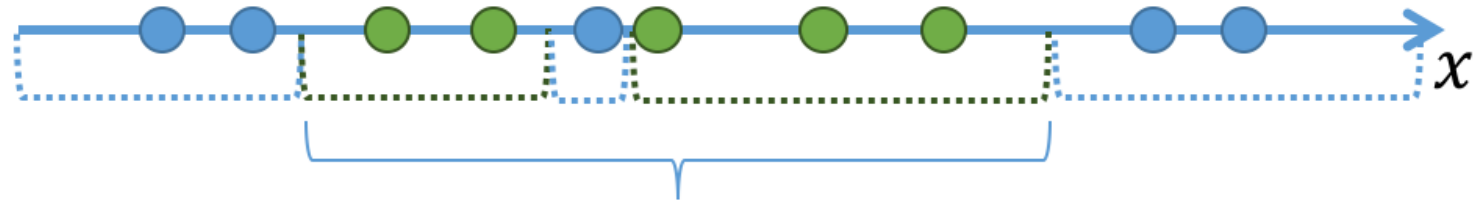


Простейшая выборка

Вариант 1: потребовать от модели максимальной точности



Вариант 2: разрешить объединение интервалов



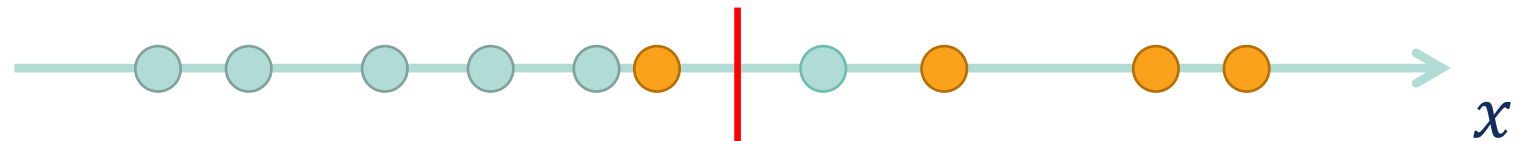
Такие интервалы можно строить последовательно

Простая выборка

Итак, выборка линейно не разделима

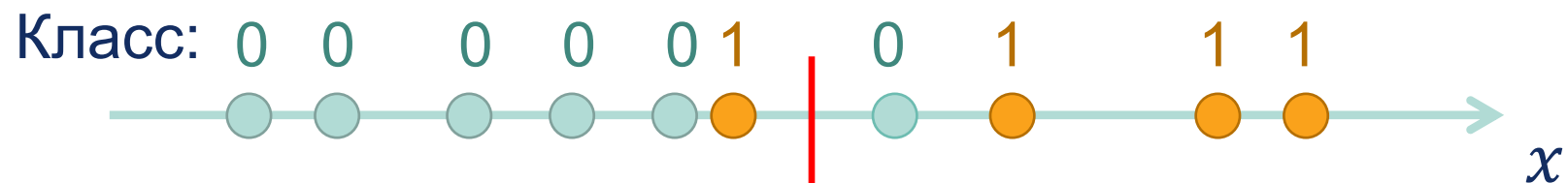


Требуется выбрать оптимальный порог:



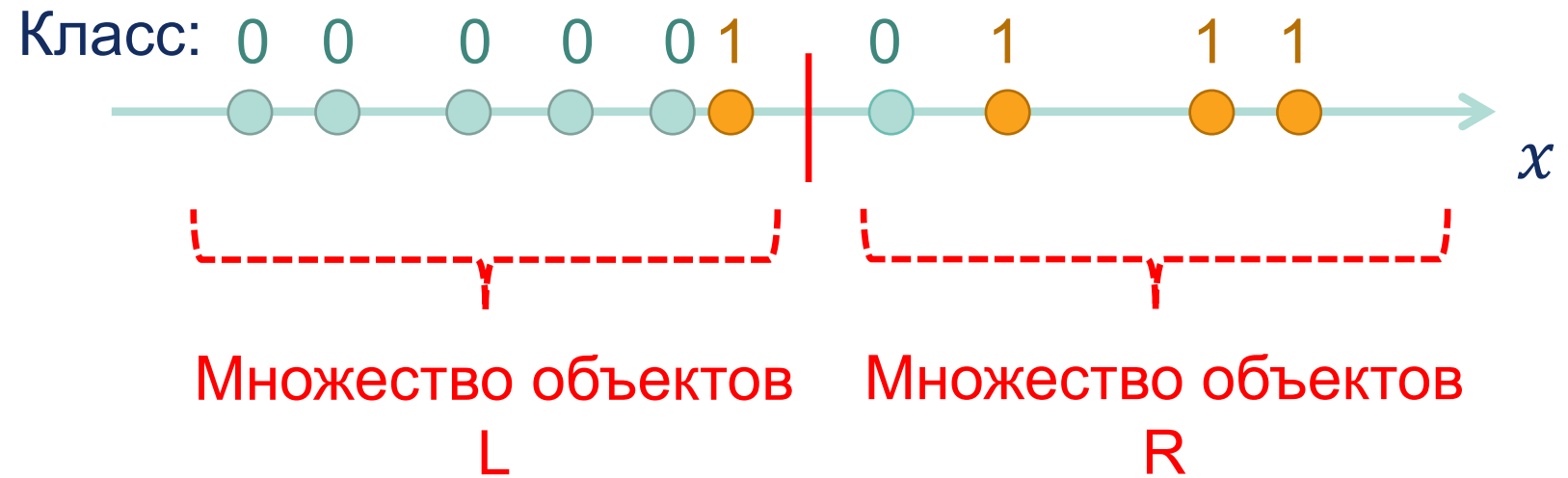
Как поставить задачу?

Задача оптимизации



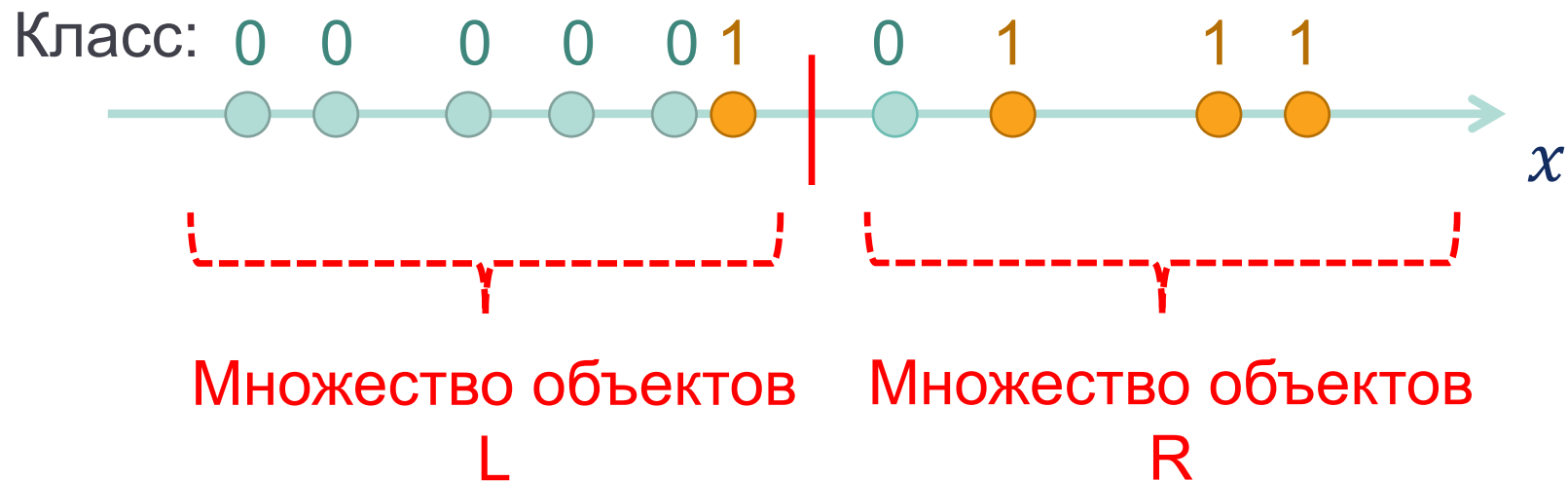
Дерево
решений

Задача оптимизации



Дерево
решений

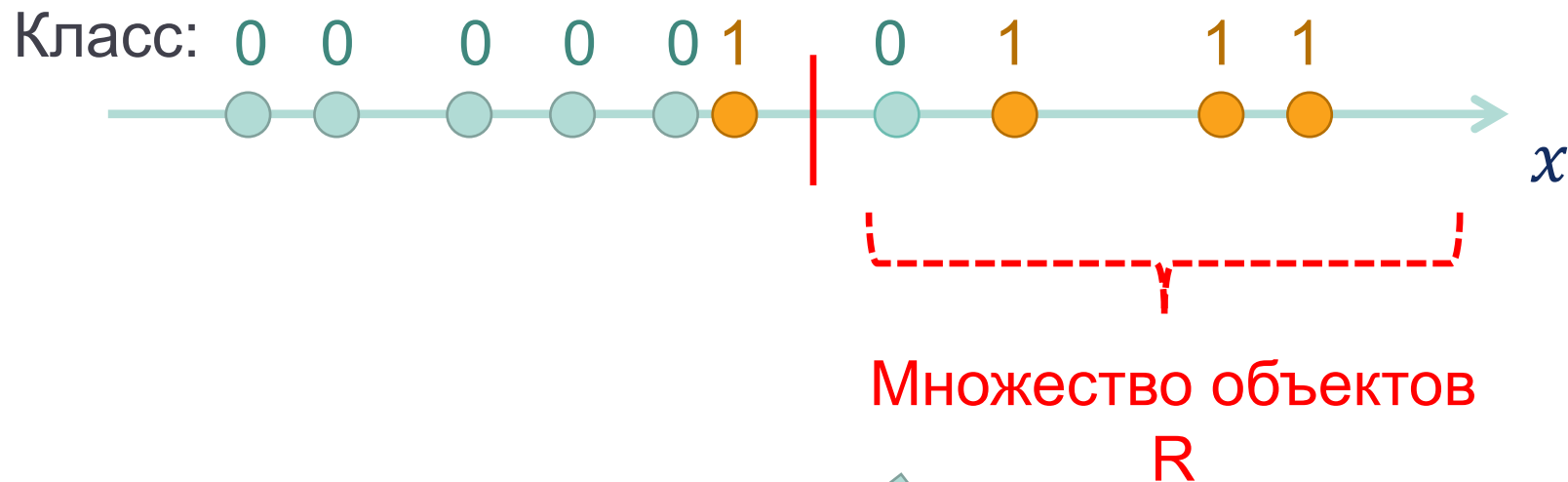
Задача оптимизации



Дерево
решений

Чтобы разделить классы хорошо – нужно, чтобы и в L и в R преобладал только один класс

Задача оптимизации

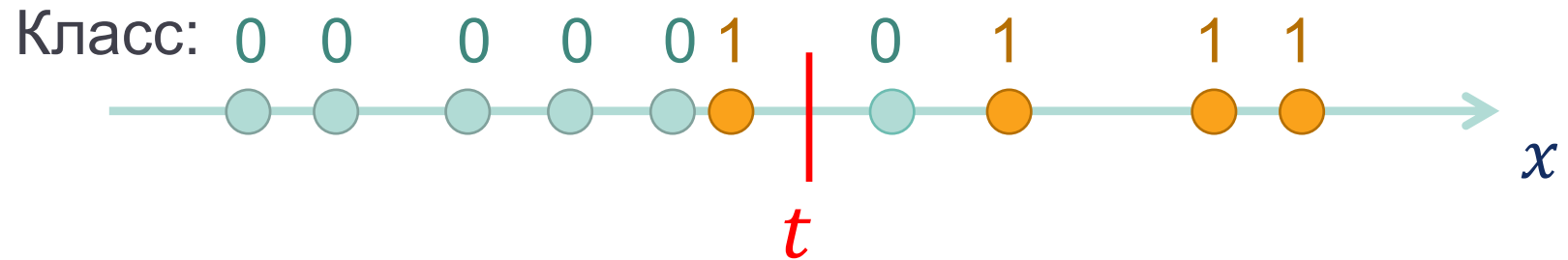


Пусть p_0 — доля класса 0 в R , а p_1 — доля класса 1 в R
В нашем примере $p_0 = \frac{1}{4}$, а $p_1 = \frac{3}{4}$

Как записать, что один из классов преобладает?

Дерево
решений

Задача оптимизации



Как записать, что один из классов должен преобладать в R?

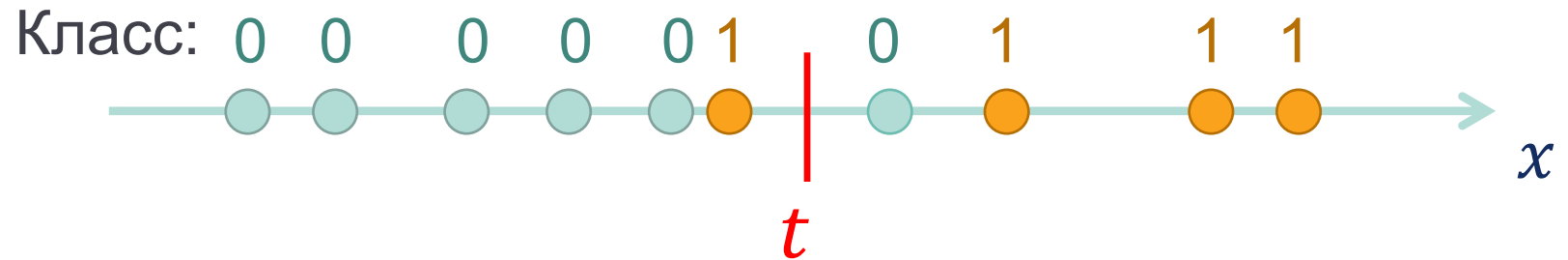
Например, так:

$$p_{max} = \max\{p_0, p_1\} \rightarrow \max_t$$

Или так:

$$1 - p_{max} \rightarrow \min_t$$

Задача оптимизации

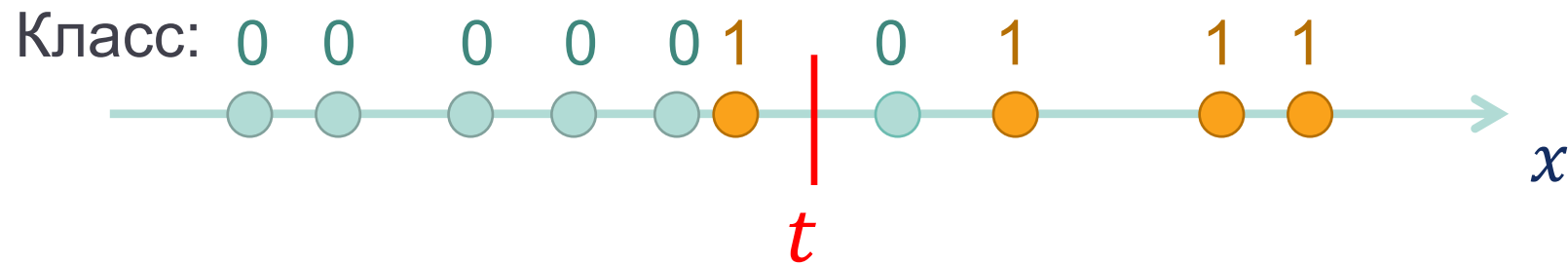


Другой вариант:

$$H(R) = -p_0 \ln p_0 - p_1 \ln p_1 \rightarrow \min_t$$

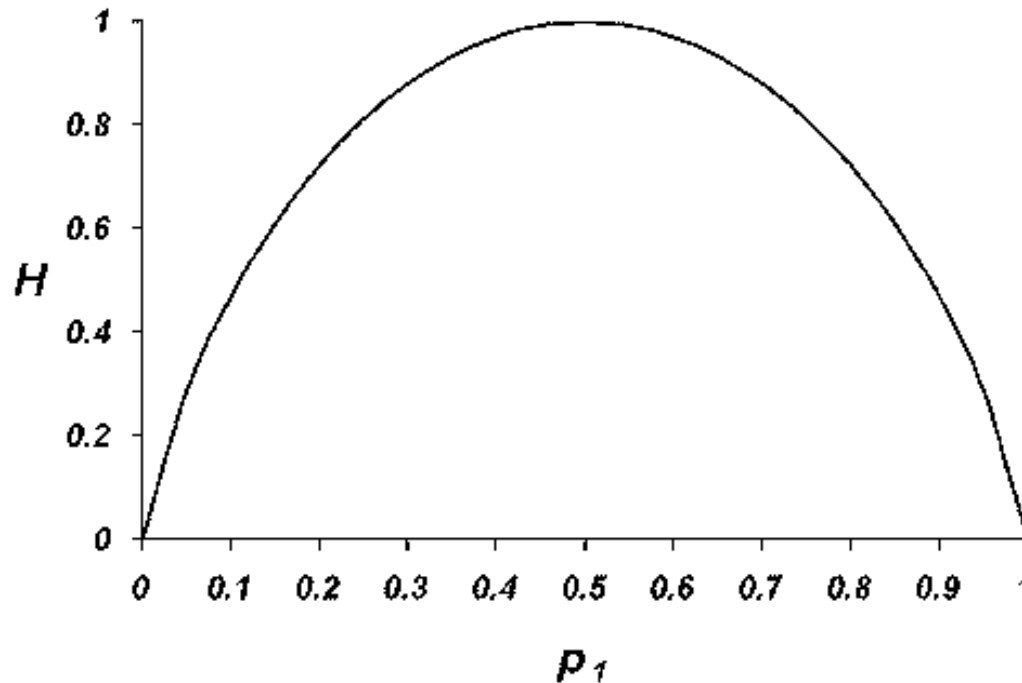
Дерево
решений

Задача оптимизации



Другой вариант:

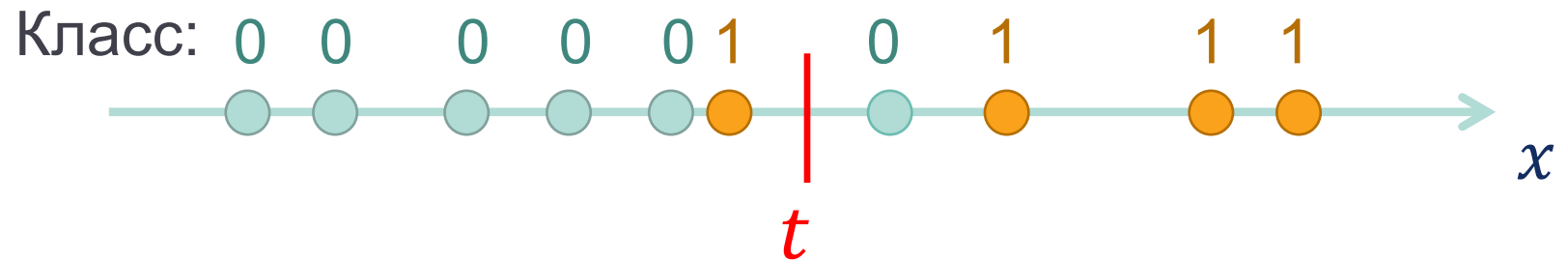
$$H(R) = -p_0 \ln p_0 - p_1 \ln p_1 \rightarrow \min_t$$



Дерево
решений

Дерево решений

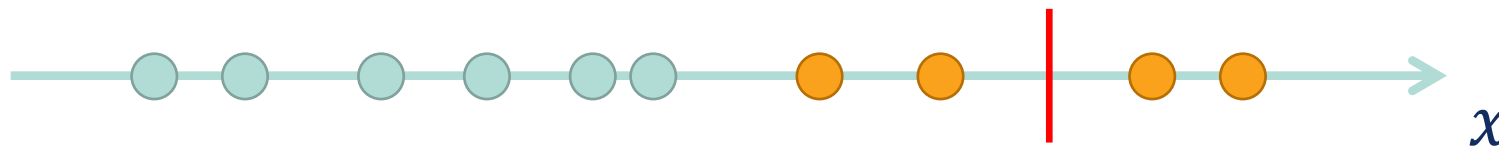
Задача оптимизации



Все это разные способы задать оптимизационную задачу, которую мы можем решить, перебирая порог t

Дерево решений

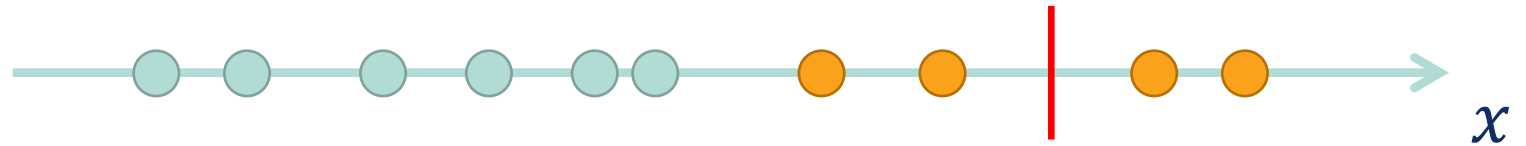
Но если смотреть только на R , можем разделить выборку так:



Здесь проблема возникает только в левой части, в правой части преобладает один класс

Дерево решений

Но если смотреть только на R , можем нечаянно разделить выборку так:



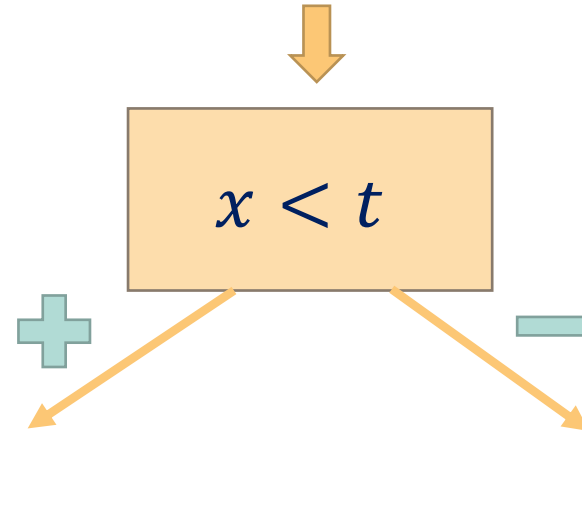
Здесь проблема возникает только в левой части, в правой части преобладает один класс

Значит надо учитывать обе части: R и L

Дерево решений

Оптимизация разбиения

Вся выборка (n объектов)

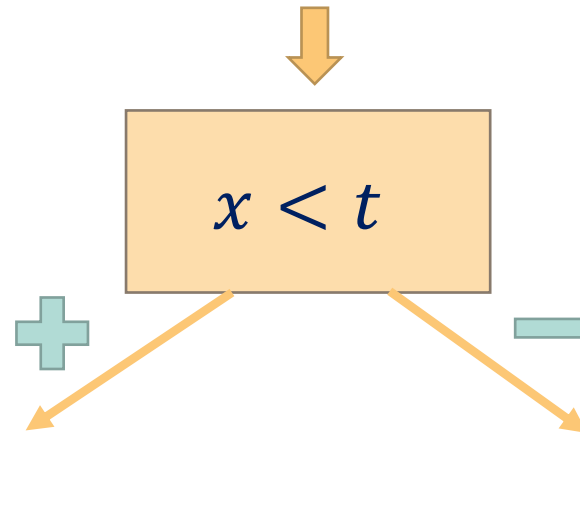


$$G(t) = H(L) + H(R) \rightarrow \min_t$$

$H(R)$ - мера «неоднородности»
(impurity) множества R

Оптимизация разбиения

Вся выборка (n объектов)



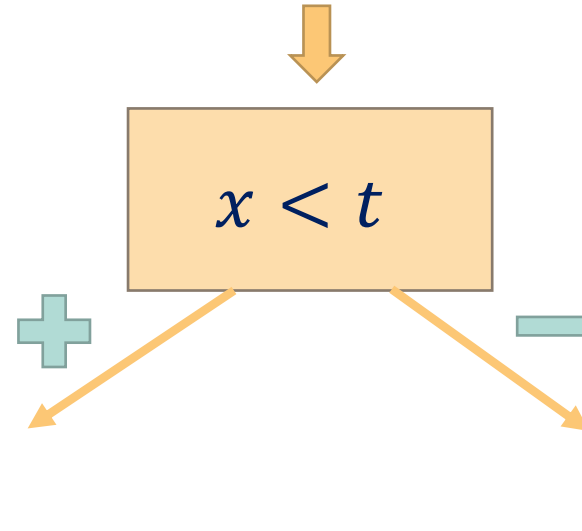
Дерево
решений

$$G(t) = H(L) + H(R) \rightarrow \min_t$$

Но что если L и R сильно разного размера?
Учтем это.

Оптимизация разбиения

Вся выборка (n объектов)



Дерево
решений

$$G(t) = \frac{|L|}{n} H(L) + \frac{|R|}{n} H(R) \rightarrow \min_t$$

Оптимизация разбиения

$H(R)$ — мера «неоднородности» множества R

Дерево
решений

Дерево решений

Оптимизация разбиения

$H(R)$ — мера «неоднородности» множества R

Варианты этой функции:

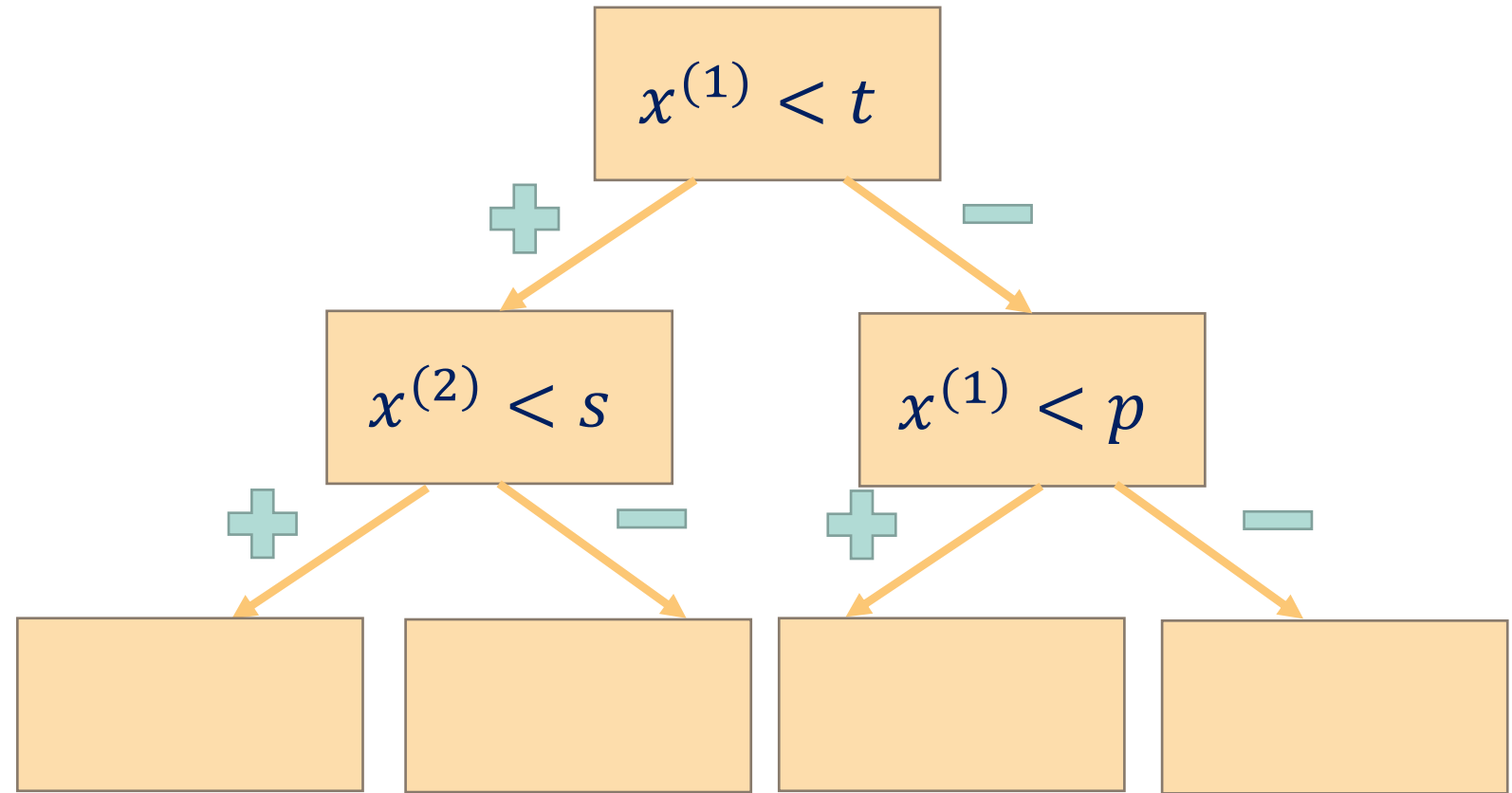
1) Misclassification criteria: $H(R) = 1 - \max\{p_0, p_1\}$

2) Entropy criteria: $H(R) = -p_0 \ln p_0 - p_1 \ln p_1$

3) Gini criteria: $H(R) = 1 - p_0^2 - p_1^2 = 2p_0p_1$

Обобщение для N признаков

Дерево
решений



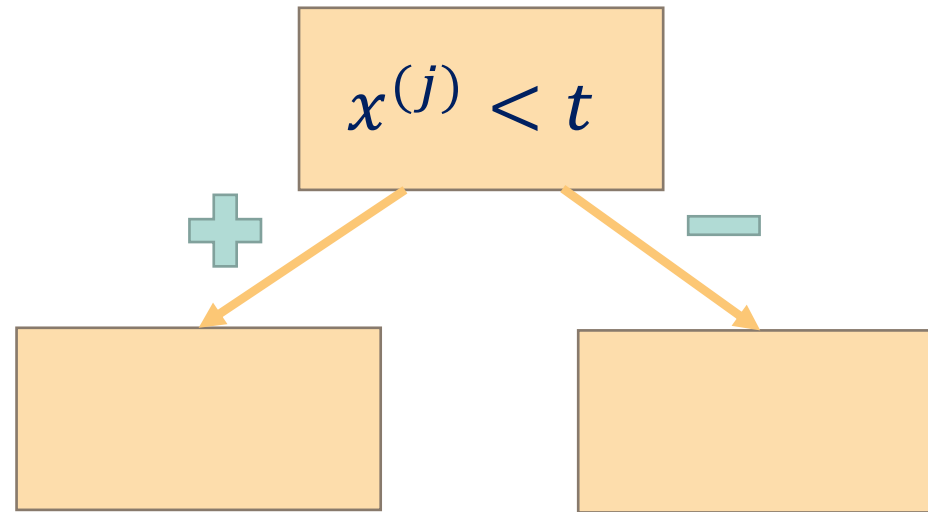
Рекурсивное построение

$$x^{(j)} < t$$

Дерево
решений

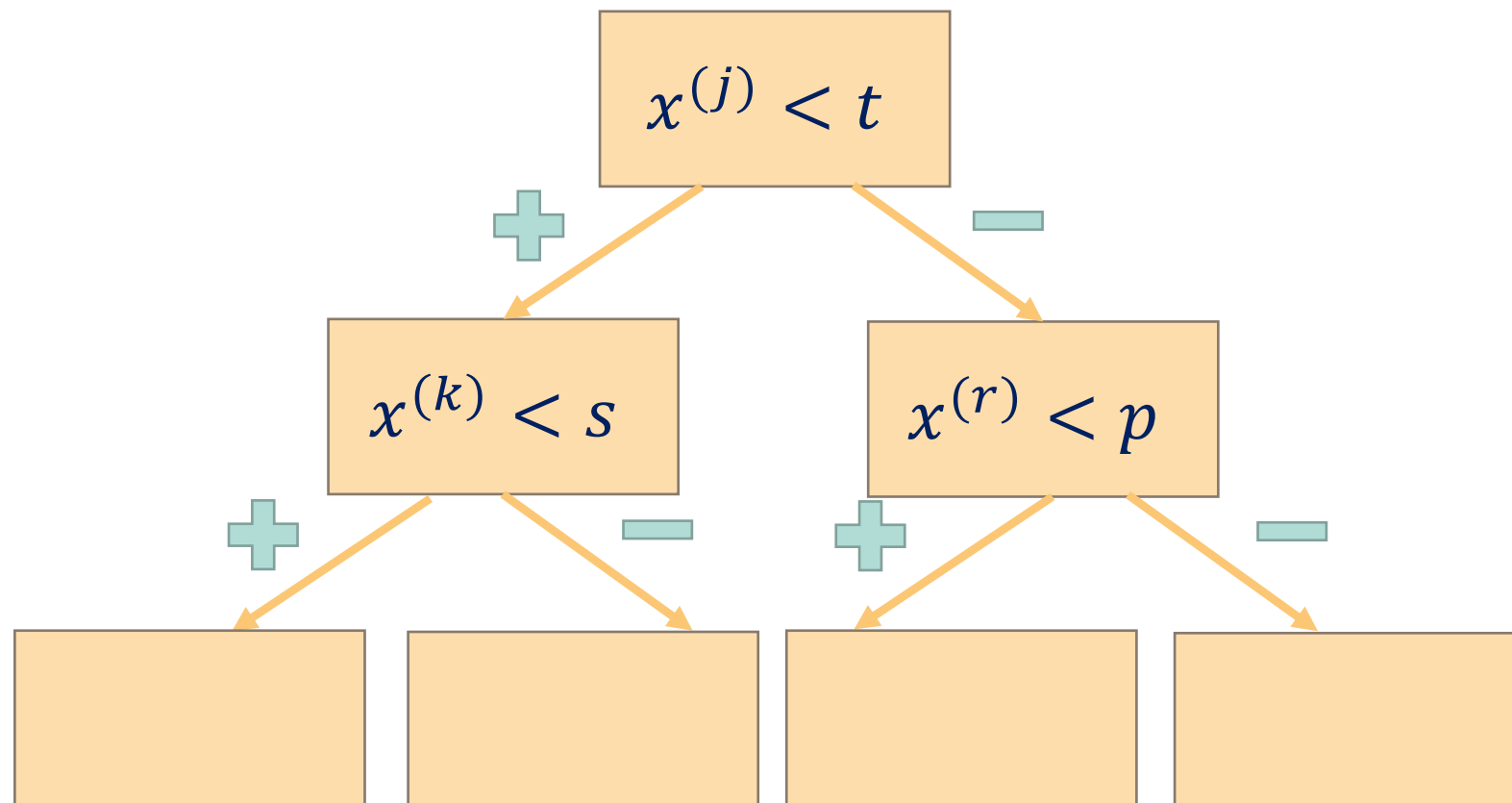
Рекурсивное построение

Дерево
решений



Рекурсивное построение

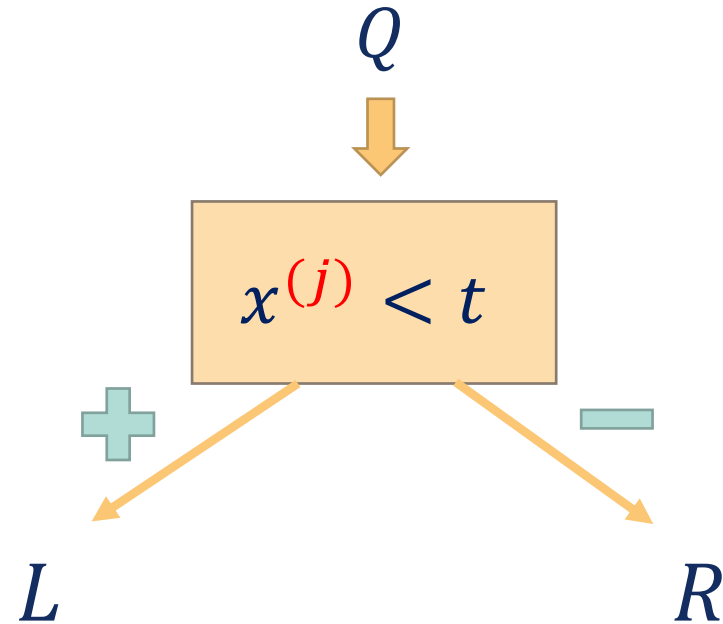
Дерево
решений



Процесс можно продолжать в тех узлах, в
которые попадает достаточно много объектов

Дерево решений

Рекурсивное построение



$$G(j, t) = \frac{|L|}{|Q|} H(L) + \frac{|R|}{|Q|} H(R) \rightarrow \min_{j, t}$$

Дерево решений

Рекурсивное построение

$H(R)$ — мера «неоднородности» множества R

Варианты этой функции:

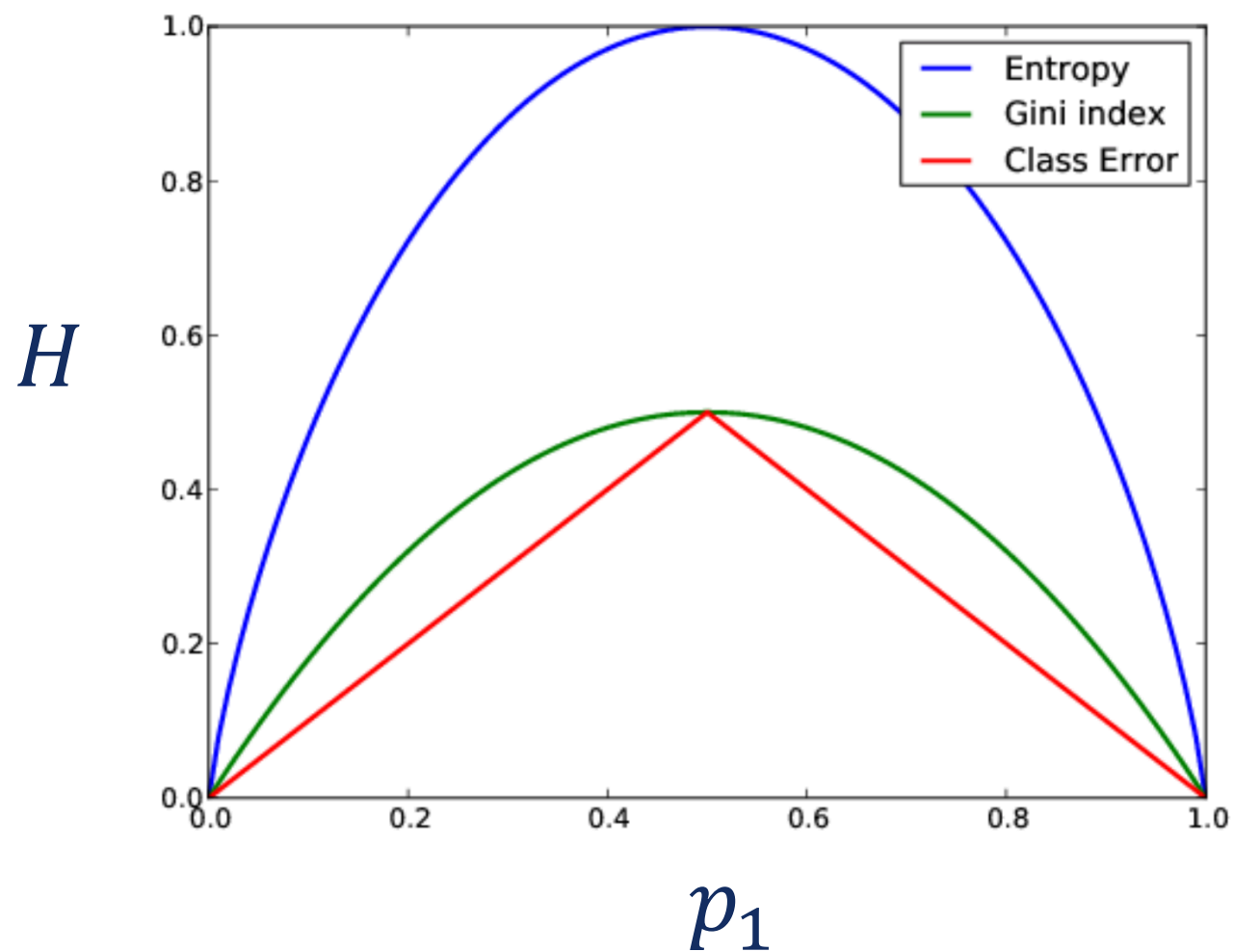
1) Misclassification criteria: $H(R) = 1 - \max\{p_0, p_1\}$

2) Entropy criteria: $H(R) = -p_0 \ln p_0 - p_1 \ln p_1$

3) Gini criteria: $H(R) = 1 - p_0^2 - p_1^2 = 2p_0p_1$

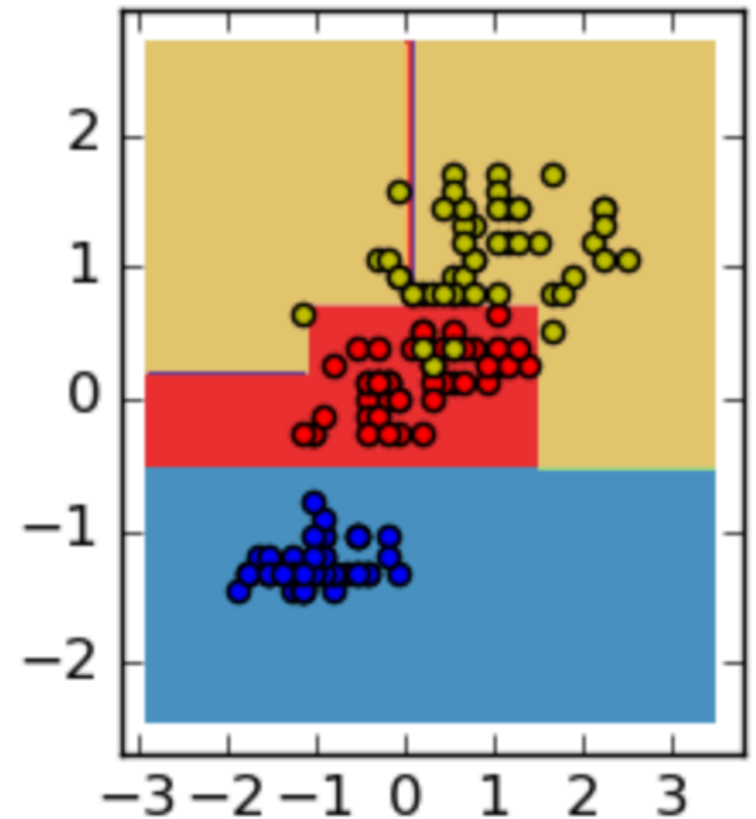
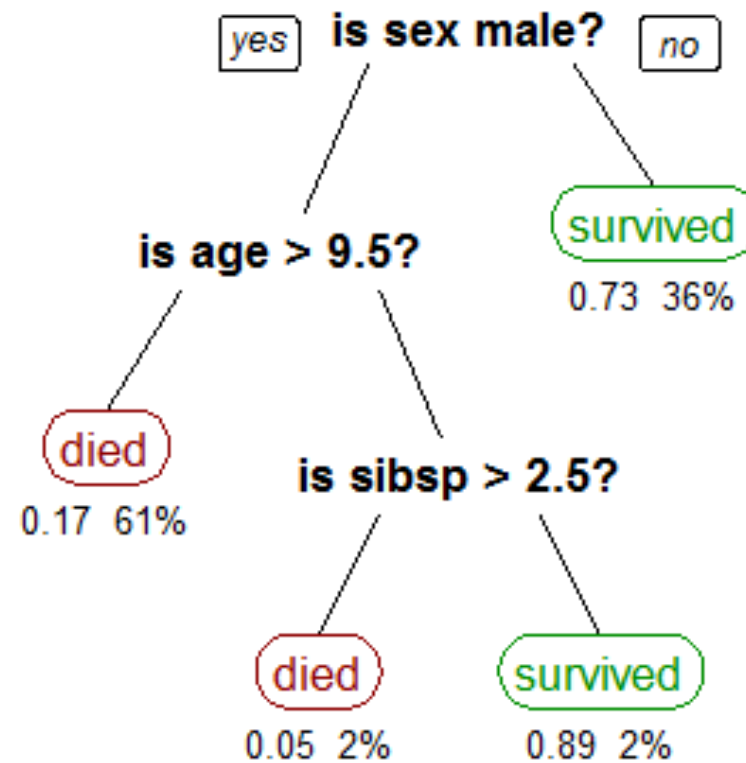
Дерево решений

Критерии разбиений



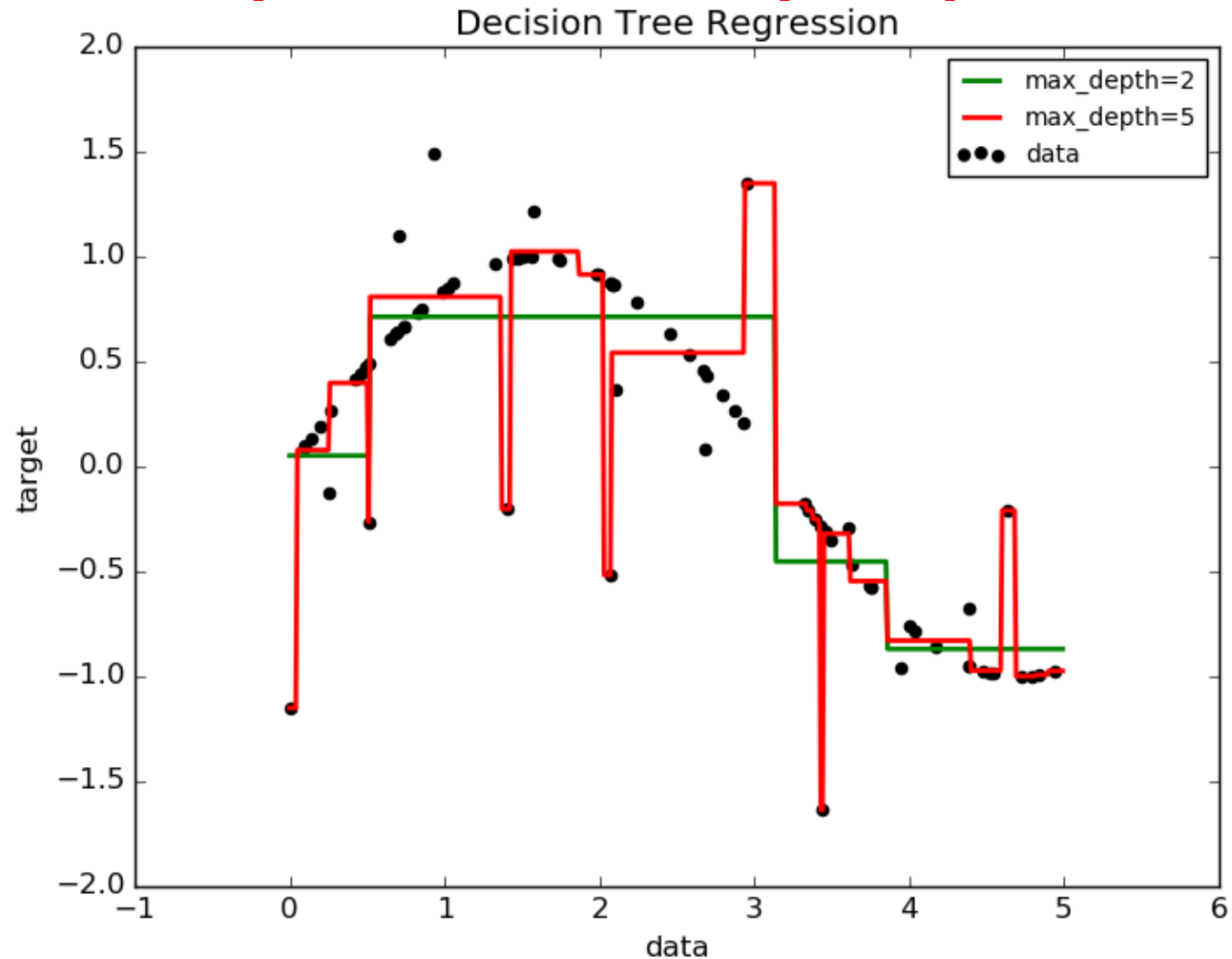
Дерево решений

Дерево решений



Дерево решений

Дерево решений: регрессия



В каждом листе дерево отвечает
некоторой константой

Дерево решений

Деревья решений

Область применения:

- базовый алгоритм в ансамбле
- очень небольшие выборки
- алгоритм для интерпретации сложной модели

Ограничения:

- сильнейшее переобучение

Ансамбли моделей

Ансамбли моделей

Способы комбинирования моделей

- Bagging
- Stacking
- Blending
- Boosting

Ансамбли моделей

Bagging

Bagging = Bootstrap aggregation

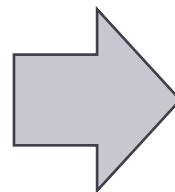
| № | значение |
|---|----------|
| 1 | |
| 2 | |
| 3 | |
| | |
| | |
| | |
| N | |

Ансамбли моделей

Bagging

Bagging = Bootstrap aggregation

| № | значение |
|---|----------|
| 1 | |
| 2 | |
| 3 | |
| | |
| | |
| | |
| N | |



| № | значение |
|----|----------|
| 1 | |
| 25 | |
| 1 | |
| | |
| | |
| | |
| | |

| № | значение |
|----|----------|
| 67 | |
| 24 | |
| 13 | |
| | |
| | |
| | |

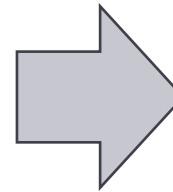
| № | значение |
|---|----------|
| 9 | |
| 9 | |
| 9 | |
| | |
| | |
| | |

Ансамбли моделей

Bagging

Bagging = Bootstrap aggregation

| № | значение |
|---|----------|
| 1 | |
| 2 | |
| 3 | |
| | |
| | |
| | |
| N | |



| № | значение |
|----|----------|
| 1 | |
| 25 | |
| 1 | |
| | |
| | |
| | |

| № | значение |
|----|----------|
| 67 | |
| 24 | |
| 13 | |
| | |
| | |

| № | значение |
|---|----------|
| 9 | |
| 9 | |
| 9 | |
| | |
| | |
| | |

По схеме выбора с возвращением, генерируем M обучающих выборок такого же размера, обучаем на них модели и усредняем результат

Ансамбли моделей

Stacking

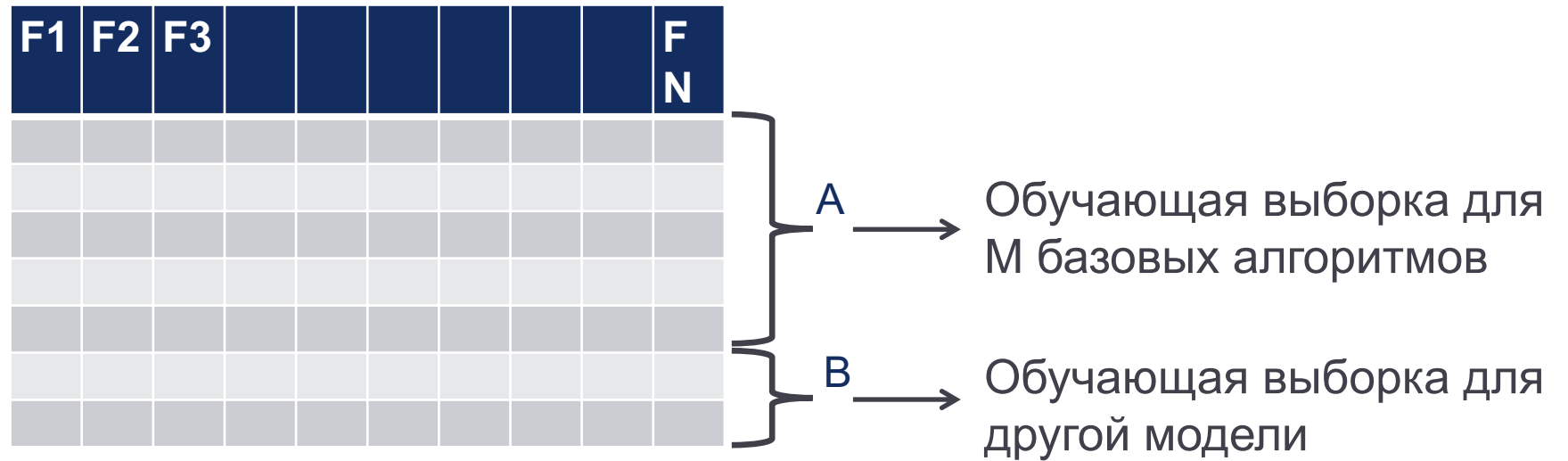
Обучающая выборка:

[illegible]

Ансамбли моделей

Stacking

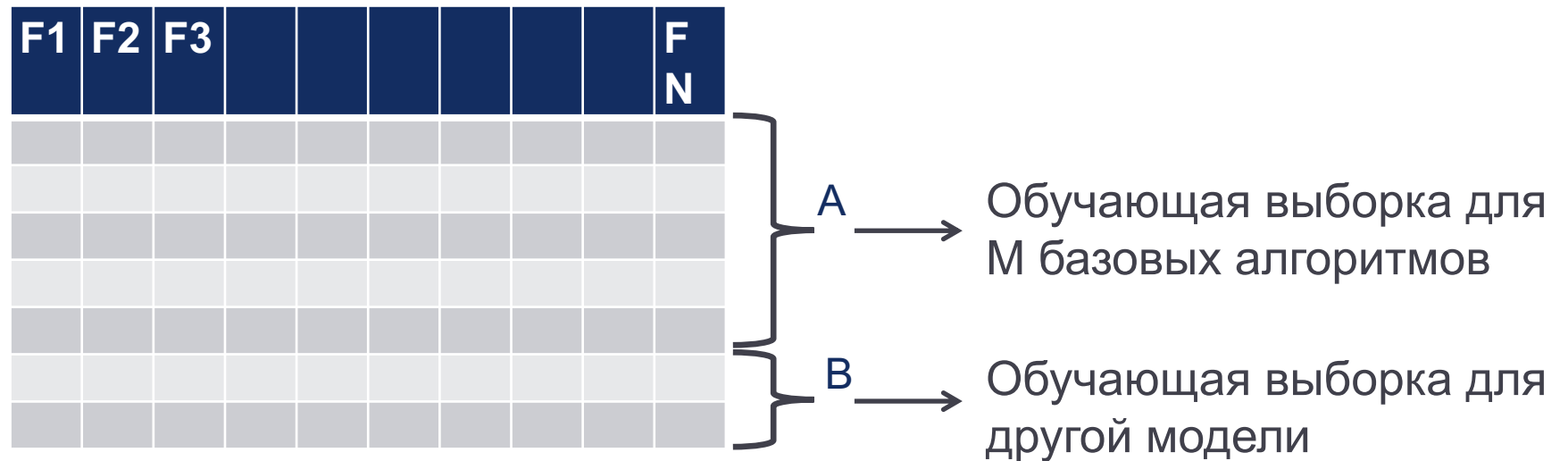
Обучающая выборка:



Ансамбли моделей

Stacking

Обучающая выборка:



Обучаем М базовых
алгоритмов на
выборке А



Считаем их
прогнозы на
выборке В

Ансамбли моделей

Stacking

Обучающая выборка:

| F1 | F2 | F3 | | | | | | | F N |
|----|----|----|--|--|--|--|--|--|-----|
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

A

Обучающая выборка для
М базовых алгоритмов

B

Обучающая выборка для
другой модели

Обучаем М базовых
алгоритмов на
выборке A

Считаем их
прогнозы на
выборке B

| B1 | B2 | | | BM |
|----|----|--|--|----|
| | | | | |
| | | | | |

Обучаем другую
модель (например,
линейную регрессию)

$$a(x) = \sum_{t=1}^T \alpha_t b_t(x)$$

Ансамбли моделей

Blending

Смесь нескольких сильных классификаторов:

$$a(x) = \sum_{t=1}^T \alpha_t b_t(x)$$

+ веса неотрицательны и дают в сумме единицу

Ансамбли моделей

Blending

Смесь нескольких сильных классификаторов:

$$a(x) = \sum_{t=1}^T \alpha_t b_t(x)$$

+ веса неотрицательны и дают в сумме единицу

Преимущества и недостатки:

- Очень прост идейно, хорошо работает, логичен
- Иногда надо перебирать веса или использовать дискретную оптимизацию
- Не всегда композиция в виде взвешенной суммы — то, что надо. Иногда нужна более сложная композиция


Ансамбли моделей

Blending

Бустинг – жадное построение взвешенной суммы базовых алгоритмов $h_k(x)$

Blending

Бустинг – жадное построение взвешенной суммы базовых алгоритмов $h_k(x)$


$$a(x) = \beta_1 h_1(x)$$


$$a(x) = \beta_1 h_1(x) + \beta_2 h_2(x)$$




...


Ансамбли моделей

Blending

Бустинг – жадное построение взвешенной суммы базовых алгоритмов $h_k(x)$

$$a(x) = \sum_{t=1}^T \beta_t h_t(x)$$


$$a(x) = \beta_1 h_1(x)$$


$$a(x) = \beta_1 h_1(x) + \beta_2 h_2(x)$$



...


Ансамбли моделей


Blending

Бустинг – жадное построение взвешенной суммы базовых алгоритмов $h_k(x)$

$$a(x) = \sum_{t=1}^T \beta_t h_t(x)$$

$h_k(x)$ – как правило, решающие деревья небольшой глубины или линейные модели


$$a(x) = \beta_1 h_1(x)$$


$$a(x) = \beta_1 h_1(x) + \beta_2 h_2(x)$$

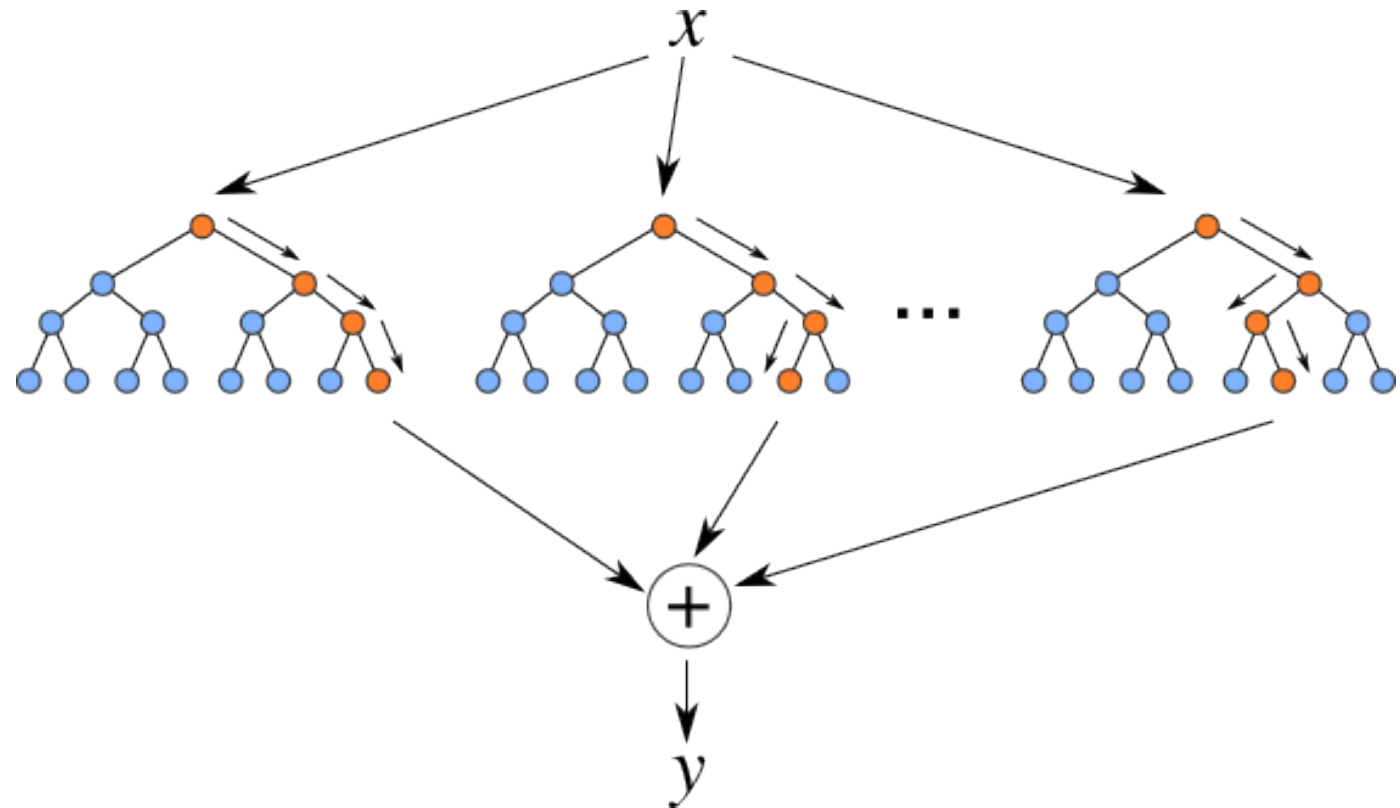


...

Random forest

Random forest

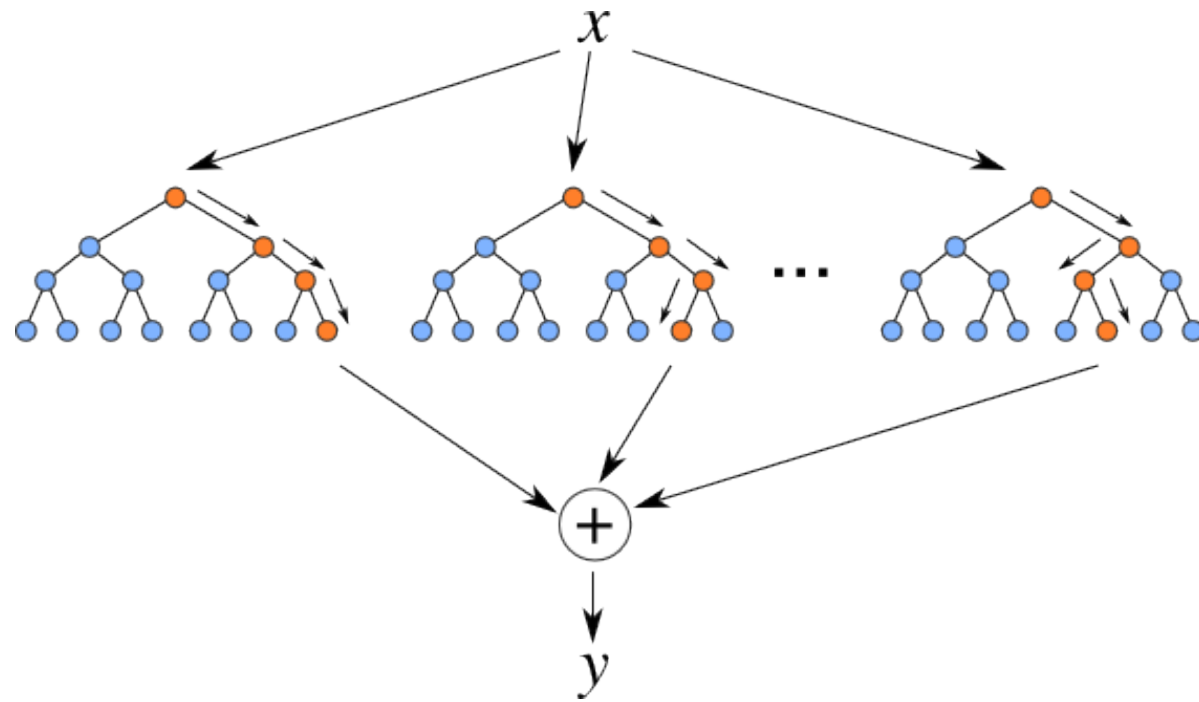
Random
Forest



Random Forest

Random forest: построение

1. Генерируем M выборок на основе имеющейся
2. Строим на них деревья с рандомизированными разбиениями в узлах: выбираем k случайных признаков и ищем наиболее информативное разбиение по ним
3. При прогнозировании усредняем ответ всех деревьев



Random Forest

Random Forest

- Позволяет обучать модель распределено
- Умеренно интерпретируем
- Устойчив к переобучению

Gradient Boosting

Gradient Boosted Decision Trees

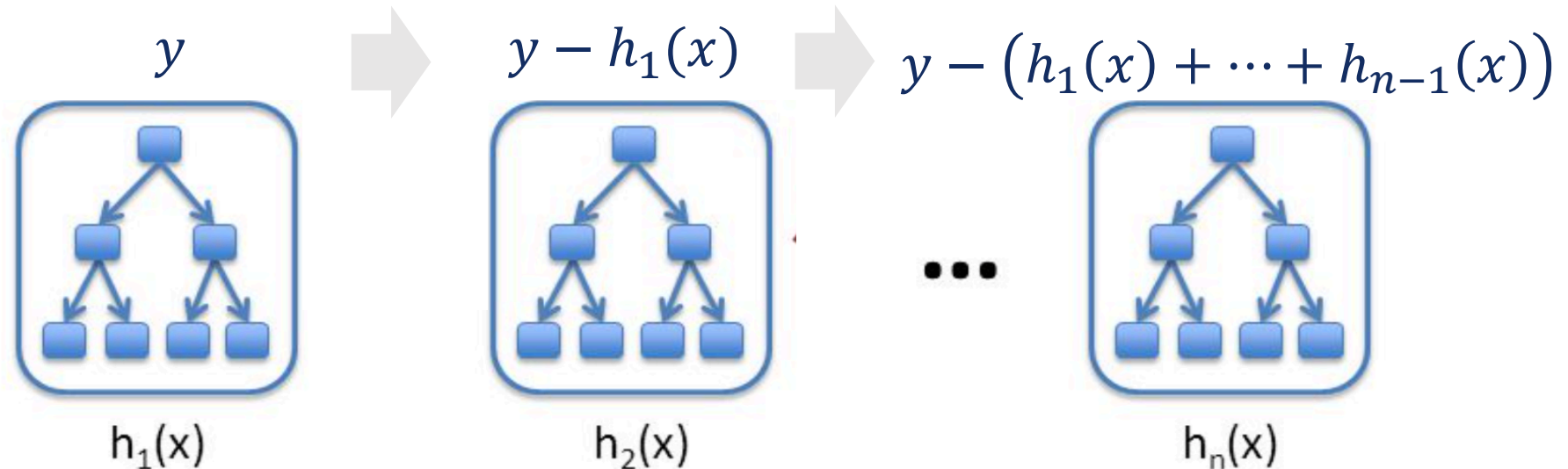
$$h(x) = h_1(x) + \cdots + h_n(x)$$

GBDT

Gradient Boosted Decision Trees

$$h(x) = h_1(x) + \cdots + h_n(x)$$

GBDT



GBDT

Gradient Boosted Decision Trees

- Каждое новое дерево $h_k(x)$ обучаем на ответы $y_i - h_i$
 h_i - прогноз всей композиции на i -том объекте на предыдущей итерации
- Коэффициент β_k перед новым деревом подбираем с помощью численной оптимизации ошибки

Gradient Boosted Decision Trees

GBDT

$$a(x) = \sum_{t=1}^T \beta_t h_t(x)$$

Идея: будем каждый следующий алгоритм выбирать так, чтобы он приближал антиградиент ошибки

$$h_t(x) \approx -\frac{\partial Q(\hat{y}, y)}{\partial \hat{y}}$$

GBDT

Gradient Boosted Decision Trees

1. Обучаем первый базовый алгоритм h_1 , $\beta_1 = 1$
2. Повторяем в цикле по t от 2 до T :

обучаем h_t на ответы $y_i - a_{t-1}(x_i)$

выбираем β_t

GBDT

Gradient Boosted Decision Trees

1. Обучаем первый базовый алгоритм h_1 , $\beta_1 = 1$
2. Повторяем в цикле по t от 2 до T :

обучаем h_t на ответы $y_i - a_{t-1}(x_i)$

выбираем β_t

Стратегии выбора β_t :

- всегда равен небольшой константе
- как в методе наискорейшего спуска
- уменьшая с ростом t

GBDT

Gradient Boosted Decision Trees

- Позволяет очень точно приблизить восстанавливаемую функцию или разделяющую поверхность классов
- Плохо интерпретируем
- Композиции могут содержать десятки тысяч базовых моделей и долго обучаться
- Переобучение на выбросах при избыточном количестве классификаторов

Машинное обучение: деревья решений и ансамбли

Спасибо!
Эмили Драль