АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

ЛЕКЦИЯ 11

Деревья в машинном обучении. Преобразование Фурье

Дерево разбивает пространство признаков, а преобразование Фурье описывает сигнал частотами.

11

ИЗ 15 ЛЕКЦИЙ



МОДУЛЬ 3 Динамическое программирование

1 КУРС / 2 АКАДЕМИЧЕСКИХ ЧАСА / PYTHON 3

Цель и содержание лекции

ЦЕЛЬ И РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

построить один узел дерева решений по размеченным данным, объяснить путь прогноза и вычислить спектр короткого сигнала дискретным преобразованием Фурье.

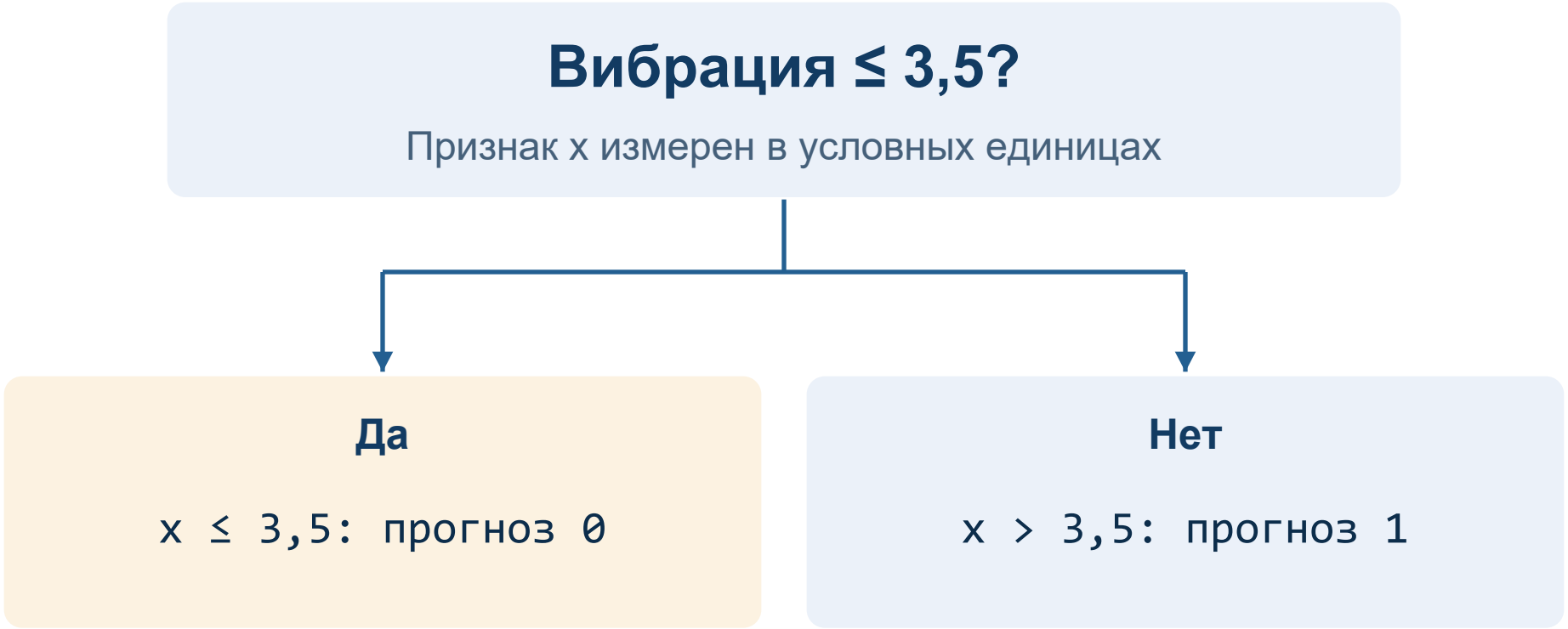
- Различать дерево поиска и дерево решений.
- Читать путь от корня дерева решений до листа.
- Интерпретировать частотные компоненты простого сигнала.

ПЛАН ЛЕКЦИИ

1. **Назначение дерева поиска и дерева решений**
2. **Выбор порога по критерию Джини и прогноз в листе**
3. **Дискретное преобразование Фурье и частотные коэффициенты**
4. **БПФ и интерпретация спектра**

Порог делит набор на две группы

СХЕМА



В листе хранится прогноз. Объект с x = 5 попадает в правую ветвь.

Дерево поиска и дерево решений

01 ТЕМА

В двоичном дереве поиска узел содержит ключ. Меньшие ключи находятся в левом поддереве, большие — в правом. Сбалансированные варианты поддерживают логарифмическую высоту, благодаря чему поиск и обновления выполняются за `O(log n)`. Балансировка меняет форму дерева, сохраняя ключи и возможность их поиска. Например, в индексе библиотеки по номеру книги дерево возвращает запись с заданным номером. В базах данных задачу индексации также решают B-деревья, узлы которых могут содержать несколько ключей и иметь больше двух потомков.

Дерево решений получает признаки объекта и выдает прогноз. Внутренний узел проверяет условие вида «температура выше 25 градусов?», а лист хранит результат. В классификации результатом может быть наиболее частый класс среди обучающих объектов, дошедших до листа, в регрессии с квадратичной ошибкой — среднее значение целевой переменной. Один и тот же признак может встречаться на нескольких уровнях, а разные узлы могут проверять разные признаки.

Дерево поиска и дерево решений

01 ПРОДОЛЖЕНИЕ

При обучении алгоритм подбирает условия по наблюдениям с известными ответами. При прогнозировании новый объект проходит по одной ветви в каждом узле. Стоимость одного прогноза составляет $O(h)$, где h — высота дерева. Качество разделений определяет форму дерева, поэтому высота зависит от данных и настроек. Путь от корня к листу можно записать как последовательность условий и объяснить пользователю, какие признаки повлияли на ответ.

От данных к первому разбиению

Рассмотрим учебные измерения вибрации двигателя. Признак `x` — максимальная вибрация в условных единицах. Метка `y` равна `1`, если техник после измерения обнаружил неисправность, и `0` в остальных случаях. Возьмем обучающие пары `(1, 0)`, `(2, 0)`, `(3, 0)`, `(4, 1)`, `(5, 0)` и `(6, 1)`.

Критерий Джини измеряет неоднородность классов в узле. Из единицы вычитают сумму квадратов долей классов. В чистом листе значение равно нулю. У корня четыре нуля и две единицы, поэтому `G = 1 - (4/6)² - (2/6)² = 4/9`. Для каждого порога считаем Джини отдельно слева и справа, затем берем среднее с весами по числу объектов. Порог выбирают по наименьшему взвешенному значению. Без весов чистый лист из одной строки получал бы такой же вклад в расчет, как большой лист.

От данных к первому разбиению

02 ПРОДОЛЖЕНИЕ

Порог	Метки слева	Метки справа	Взвешенный Джини
2,5	0, 0	0, 1, 0, 1	1/3
3,5	0, 0, 0	1, 0, 1	2/9
5,5	0, 0, 0, 1, 0	1	4/15

Порог `3.5` дает выигрыш `4/9 - 2/9 = 2/9`. При `x = 5` путь ведет в правый лист с прогнозом `1`, хотя известная метка этой строки равна `0`. Можно продолжить разбиение правого листа, но одиночные примеры могут отражать случайный шум. Чтобы оценить переносимость правил, прогнозы проверяют на отложенных наблюдениях. Ограничения на глубину и на минимальный размер листа помогают управлять сложностью модели.

От данных к первому разбиению

02 ПРИМЕР PYTHON

11_01.py / фрагмент 1 из 2

```
from collections import Counter


def gini(labels):
    return 1 - sum((count / len(labels)) ** 2
                   for count in Counter(labels).values())


def split_score(rows, threshold):
    left = [label for value, label in rows if value <= threshold]
    right = [label for value, label in rows if value > threshold]
    score = (len(left) * gini(left) + len(right) * gini(right)) / len(rows)
    return score, left, right
```

От данных к первому разбиению

02 ПРИМЕР PYTHON

11_01.py / фрагмент 2 из 2

```
rows = [(1, 0), (2, 0), (3, 0), (4, 1), (5, 0), (6, 1)]
values = sorted({value for value, _ in rows})
thresholds = [(a + b) / 2 for a, b in zip(values, values[1:])]
threshold = min(thresholds, key=lambda t: split_score(rows, t)[0])
score, left, right = split_score(rows, threshold)
print(f"Порог {threshold}, Джини {score:.3f}")
for value in (2, 5):
    leaf = left if value <= threshold else right
    print(value, Counter(leaf).most_common(1)[0][0])
```

РЕЗУЛЬТАТ ЗАПУСКА

```
Порог 3.5, Джини 0.222
2 0
5 1
```

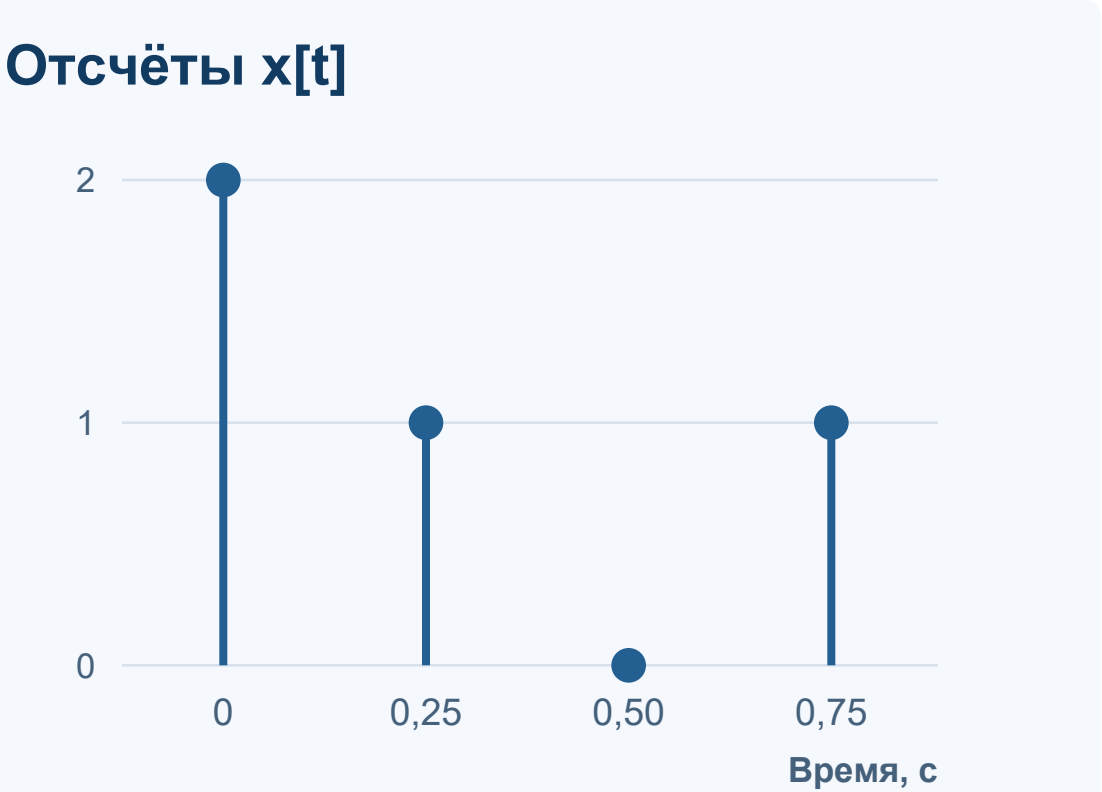
От данных к первому разбиению

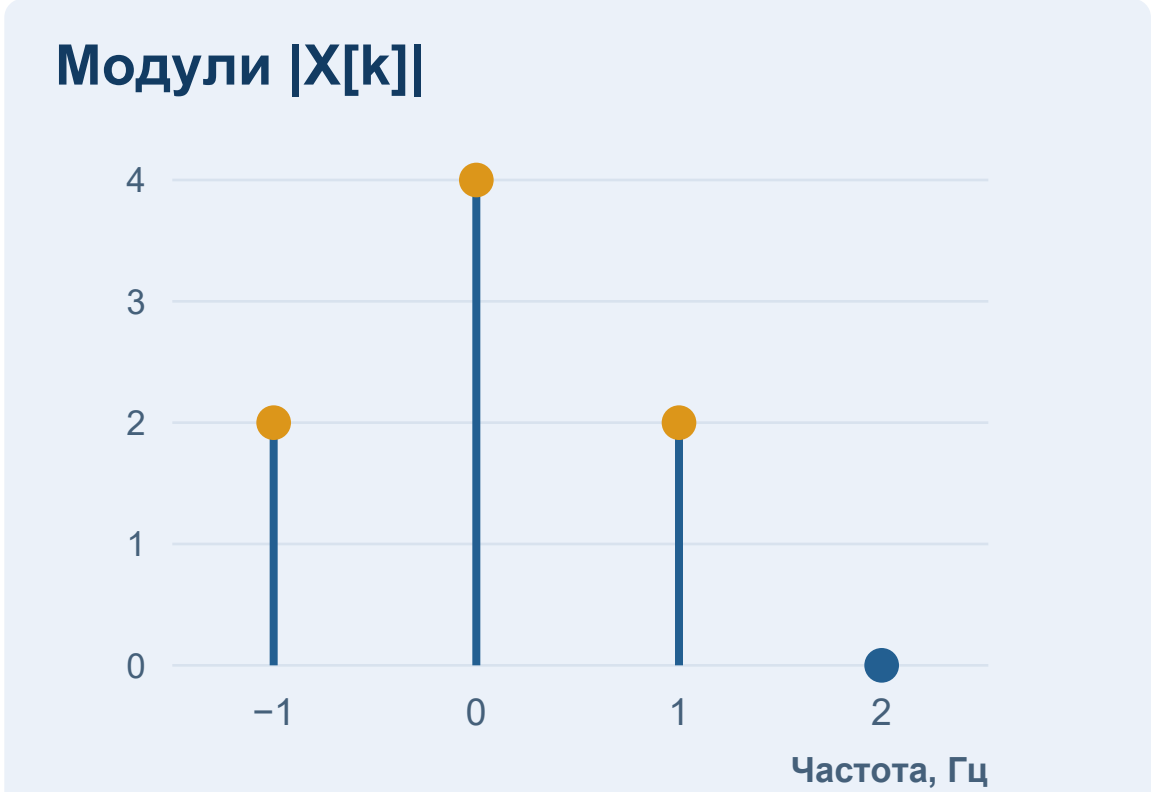
02 ПРОДОЛЖЕНИЕ

Пример обучает дерево глубины один. Нужны как минимум два различных конечных значения признака, а метки должны быть пригодны для `Counter`. Пороги берутся между соседними уникальными значениями, поскольку внутри каждого такого интервала разбиение одинаково. Для `n` строк и `q` порогов код выполняет `O(nq)` работы после сортировки и использует `O(n)` дополнительной памяти. Можно ускорить перебор, проходя по отсортированным строкам и обновляя счетчики классов. При равенстве частот в листе следует явно определить политику выбора класса.

Четыре отсчёта сигнала и их спектр

СХЕМА





Постоянная часть даёт пик при 0 Гц. Пара пиков при ±1 Гц соответствует одной вещественной гармонике.

От отсчетов сигнала к спектру

03 ТЕМА

Вибрацию можно описать числом для каждого момента времени. Для периодических явлений информативны также частоты. Например, датчик дает четыре равномерно расположенных отсчета `x = [2, 1, 0, 1]`, по одному каждые `0.25` секунды. Временная последовательность имеет среднее `1` и выглядит как один цикл косинуса, сдвинутого вверх на единицу.

Дискретное преобразование Фурье (ДПФ) превращает `N` отсчетов в `N` комплексных коэффициентов. Коэффициент `X[k]` — сумма произведений `x[t]` на `exp(-2πi·k·t/N)` по индексам `t` от `0` до `N - 1`. Здесь `i` — мнимая единица, а комплексный множитель задает фазу пробной волны. Если отсчеты согласуются с этой волной, вклады складываются. При рассогласовании они частично или полностью компенсируются. Модуль `abs(X[k])` измеряет величину коэффициента, а его аргумент задает фазу.

От отсчетов сигнала к спектру

03 ПРОДОЛЖЕНИЕ

Для `k = 0` все множители равны единице, поэтому `X[0] = 4`. Для `k = 1` множители равны `1`, `-i`, `-1`, `i`, значит сумма равна `2 - i + 0 + i = 2`. Для `k = 2` получаем `2 - 1 + 0 - 1 = 0`. При вещественных отсчетах коэффициенты `X[k]` и `X[N-k]` комплексно сопряжены для `1 ≤ k < N`. Здесь `X[3] = 2`. Вместе `k = 1` и `k = 3` описывают одну вещественную гармонику с амплитудой `2·abs(X[1])/N = 1`.

От отсчетов сигнала к спектру

03 ПРИМЕР PYTHON

11_02.py / фрагмент 1 из 2

```
from cmath import exp
from math import pi


def dft(samples):
    size = len(samples)
    return [sum(value * exp(-2j * pi * k * t / size)
                for t, value in enumerate(samples))
            for k in range(size)]


def inverse_dft(spectrum):
    size = len(spectrum)
    return [sum(value * exp(2j * pi * k * t / size)
                for k, value in enumerate(spectrum)) / size
            for t in range(size)]
```

От отсчетов сигнала к спектру

03 ПРИМЕР PYTHON

11_02.py / фрагмент 2 из 2

```
samples = [2, 1, 0, 1]
spectrum = dft(samples)
restored = inverse_dft(spectrum)
print([round(abs(value), 6) for value in spectrum])
print([round(value.real) for value in restored])
assert all(abs(a - b) < 1e-9 for a, b in zip(samples, restored))
```

РЕЗУЛЬТАТ ЗАПУСКА

```
[4.0, 2.0, 0.0, 2.0]
[2, 1, 0, 1]
```

От отсчетов сигнала к спектру

03 ПРОДОЛЖЕНИЕ

Обратное преобразование меняет знак в экспоненте и делит сумму на `N`. Проверка с допуском учитывает округления арифметики. После удаления коэффициентов сигнал восстанавливается приближенно. Например, если сохранить только `X[0]`, обратное преобразование даст `[1, 1, 1, 1]`.

Зачем нужно быстрое преобразование Фурье

04 ТЕМА

Прямое вычисление обрабатывает `N` вкладов для каждого из `N` коэффициентов, то есть работает за `O(N²)`. Быстрое преобразование Фурье (БПФ) вычисляет те же коэффициенты за `O(N log N)`. При длине `N`, равной степени двойки, рекурсивный вариант сначала преобразует отсчеты с четными и нечетными индексами. Затем он объединяет два спектра, умножая нечетную часть на фазовые множители. Значения для индексов `k` и `k + N/2` используют одну пару промежуточных коэффициентов с разными знаками. На каждом уровне суммарное число операций линейно, а глубина деления логарифмическая.

Для анализа реальных сигналов сначала проверяют, равномерно ли расположены отсчеты и известна ли частота дискретизации `fs`. Шаг между соседними частотами спектра равен `fs/N`. В нашем примере `fs = 4 Гц`, поэтому индекс `k = 1` соответствует `1 Гц`, а `k = 3` — `-1 Гц` в двустороннем спектре. При конечной записи сигнал с частотой между отсчетами спектра дает вклад в несколько коэффициентов. При слишком редкой дискретизации разные непрерывные частоты могут дать одинаковые отсчеты. По этим причинам высота спектрального пика требует контекста измерений.

Зачем нужно быстрое преобразование Фурье

04 ПРОДОЛЖЕНИЕ

Спектральные признаки можно передать в дерево решений. Например, узел сравнивает мощность вибрации в заданной полосе с порогом. Преобразование вычисляет признаки из сырых отсчетов, а дерево учится связывать эти признаки с размеченными состояниями двигателя.

Что нужно уметь после лекции

ИТОГИ

- Дерево решений подбирает правила по данным и делает прогноз по пути к листу.
- Взвешенный Джини оценивает, насколько разбиение уменьшило смешение классов.
- ДПФ представляет отсчеты через частотные коэффициенты. БПФ ускоряет их расчет за счет повторного использования промежуточных результатов.

> Дерево разбивает пространство признаков, а преобразование Фурье описывает сигнал частотами.

Проверьте понимание

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как меняется смысл узла при переходе от дерева поиска к дереву решений?
2. Зачем умножать неоднородность дочернего узла на долю объектов в нем?
3. Как связаны `X[0]` и среднее значение сигнала?
4. За счет чего вариант БПФ с делением пополам сокращает число операций?

Самостоятельная работа

ПРАКТИКА

1. Посчитайте взвешенный Джини при порогах `1.5` и `4.5`. Подтвердите выбор порога `3.5`.
2. Запустите код для сигнала `[1, -1, 1, -1]`. Определите ненулевой коэффициент и его частоту при `fs = 4 Гц`.
3. Замените коэффициенты `X[1]` и `X[3]` нулями в исходном примере. Восстановите сигнал и объясните изменение.

Источники и документация

ЧТЕНИЕ

ОСНОВНОЕ ЧТЕНИЕ

А. Бхаргава. Грокаем алгоритмы

СПб.: Питер, 2017. Глава 11. «Деревья» и «Преобразование Фурье». С. 254–258, 259.

ДОПОЛНИТЕЛЬНО

Scikit-learn. Decision Trees, математическое описание критериев классификации. https://scikit-learn.org/stable/modules/tree.html

Johnson S. G., Frigo M. Implementing FFTs in Practice. https://math.mit.edu/~stevenj/papers/JohnsonFr08-burrus.pdf