

Базовая математическая подготовка для Data Science

Лекция 1. Комбинаторика

Слипенчук Павел Владимирович

13.10.2023



Москва,
КиБ

1. Зачем нужна комбинаторика в DS
2. Комбинаторные принципы
3. Основные комбинаторные схемы
4. Бином Ньютона
5. Введение в теорию графов

Зачем нужна комбинаторика в DS

Комбинаторный анализ (Комбинаторика) — раздел математики, посвящённый решению задач выбора и расположения элементов некоторого, обычно конечного, множества в соответствии с установленными правилами (схемами).

Каждое такое правило определяет способ построения из элементов исходного множества некоторой конструкции, называемой *комбинаторной конфигурацией*.

Примеры задач комбинаторики

Простая задача. В школе танго 8 парней и 12 девушек. Сколько существует всевозможных пар?

$$8 \cdot 12 = 96$$

Сложная задача. Сколько потребуется бит данных, чтобы сохранить всевозможные первые 20 ходов в шахматах?

«Задача о зёрнах на шахматной доске»

Когда очень много кажется малым...

Согласно индийской легенде, брахман Сисса придумал шахматы (чатурангу). Правителю так понравилась игра, что он предложил Сиссе выбрать себе награду.

Хитрый Сисса попросил у правителя на первую клетку положить одно зёрнышко пшеницы, на вторую два, на третью четыре, на пятую восемь и так далее.

Правитель, не разбиравшийся в комбинаторике, быстро согласился, даже несколько обидевшись на столь "невысокую" цену.

Однако неделю спустя казначей доложил правителю что расплатиться невозможно, "разве что осушить моря и океаны и засеять всё пшеницей".

«Задача о зёрнах на шахматной доске»

$$C = 1 + 2 + 4 + 8 + 16 + \dots + 2^{63} = \sum_{i=0}^{63} 2^i = 2^{64} - 1 = \\ = 18\,446\,744\,073\,709\,551\,615$$

Много или мало 2^{64} зёрен пшеницы ?

В 2021 году было произведено в мире 770 млн.тонн пшеницы. Вес одного зерна примерно 50 миллиграмм.

Вопрос: сколько тысячелетий люди должны перестать есть хлеб, чтобы Сисса получил свою оплату? :)

Задач, аналогичной классической «Задаче о зёрнах...» в сфере Data Science чрезвычайно много.

Одна из целей комбинаторики: понимать что такое МНОГО, ОЧЕНЬ МНОГО.

Например:

1. «Давайте мониторить все видео на youtube с целью проверки на стеганографию!»
2. «ChatGPT скоро научиться отвечать на все вопросы!»

Road Map Data Science как мат.дисциплины

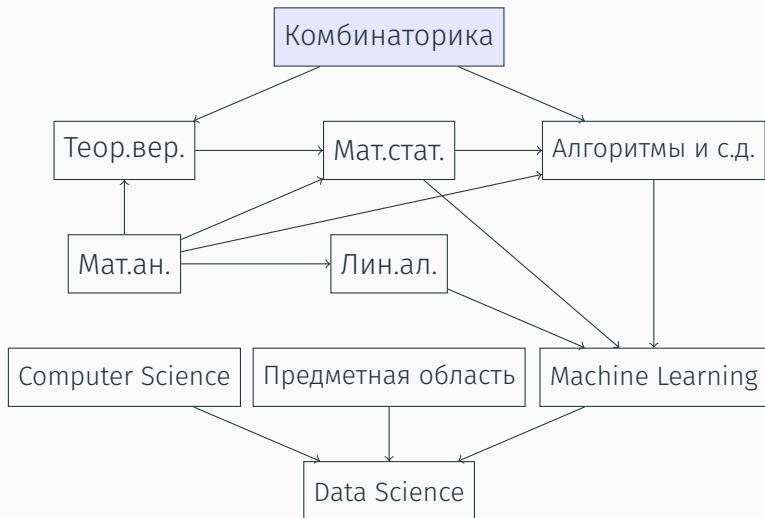


Рисунок 1: Место комбинаторики в DS

Комбинаторные принципы

Два основных правила

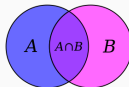
Правило суммы. (Правило «ИЛИ») Если элемент A можно выбрать a способами, а элемент B можно выбрать b способами, то выбрать A или B можно $a + b$ способами.

Правило умножения. (Правило «И») Если элемент A можно выбрать a способами, а элемент B можно выбрать b способами, то пару (A, B) можно выбрать $a \cdot b$ способами.

Формула включения-исключения

Посчитаем мощность множества (количество элементов в множестве) вида $|A \cup B|$

$$|A \cup B| = |A| + |B| - |A \cap B| \quad (1)$$



Принцип Дирихле

«Если кролики рассажены в клетки и число кроликов больше числа клеток, то хотя бы в одной из клеток находится более одного кролика.»

«Если число клеток больше числа кроликов, то как минимум одна клетка пуста.»

Ряд комбинаторных вещей мы не рассматриваем:

1. биективное доказательство;
2. производящие функции;
3. рекуррентные соотношения.

Самостоятельно их освойте после прохождения DSIS направления.

Основные комбинаторные схемы

Размещение (выборка без возвращения)

Нужно из n различных элементов выбрать **упорядоченный** ряд из k элементов. Элементы берутся не более одного раза.

(Согласно принципу Дирихле, очевидно, что $k \leq n$, иначе задача не имеет решения.)

Сколько существует таких способов?

Пример. Есть n различных действий для атаки. Сколько существует вариантов сделать атаку из k действий?

$$A_n^k = n \cdot (n - 1) \cdot (n - 2) \cdot \dots \cdot (n - k + 1) = \frac{n!}{(n - k)!} \quad (2)$$

Как вывести эту формулу?

Докажите строго формулу:

$$A_n^k = n \cdot (n - 1) \cdot (n - 2) \cdot \dots \cdot (n - (k - 1)) \quad (3)$$

Подсказка: используйте правило умножения, здравый смысл и метод математической индукции

Перестановка – частный случай *размещения*, когда $n = k$

$$P_n = A_n^n \quad (4)$$

Пример. Сколько существует способ в шифротексте из n блоков переставить эти блоки?

Нужно из n различных элементов выбрать **упорядоченный** ряд из k элементов. При этом элементы можно выбирать более одного раза.

$$\overline{A}_n^k = n^k \quad (5)$$

выведите эту формулу.

Нужно из n различных элементов выбрать **не упорядоченный** ряд из k элементов. Элементы берутся не более одного раза.

$$C_n^k = \frac{n!}{(n-k)!k!} \quad (6)$$

Давайте выведем эту формулу

...

C_n^k это то же самое, что и A_n^k , за исключением того, что порядок НЕ важен. Значит если C_n^k умножить на число перестановок P_k , то согласно правилу умножения получим A_n^k

Значит:

$$A_n^k = C_n^k \cdot P_k = C_n^k \cdot A_k^k$$

следовательно верно:

$$C_n^k = \frac{A_n^k}{A_k^k}$$

Далее нужно расписать формулу. **сделайте это дома.**

Нужно из n различных элементов выбрать **не упорядоченный** ряд из k элементов. При этом элементы можно брать более одного раза.

$$\overline{C}_n^k = C_{n+k-1}^{n-1} \quad (7)$$

Попробуйте доказать это самостоятельно, используя подсказку из методички Жуковых на стр.6 Если не получится – на ближайшем очном занятии распишем.

Мы не рассматриваем числа стирлинга 1-го и 2-го родов.

Самостоятельно их освойте после прохождения DSIS
направления.

Бином Ньютона

Бином Ньютона:

$$(x + y)^n = \sum_{k=0}^{k=n} C_n^k \cdot x^{n-k} \cdot y^k \quad (8)$$

Доказательство

$$\begin{aligned}(x + y)^2 &= x^2 + 2xy + y^2 = C_2^0 x^2 y^0 + C_2^1 xy + C_2^2 x^0 y^2 \\(x + y)^3 &= x^3 + 3x^2 y + 3xy^2 + y^3 = C_3^0 x^3 y^0 + C_3^1 x^2 y + C_3^2 xy^2 + C_3^3 x^0 y^3\end{aligned}$$

дома самостоятельно распишите $(x + y)^4$ и $(x + y)^5$

База индукции есть.

Предположим, что верно

$$(x + y)^{(n-1)} = \sum_{k=0}^{k=n-1} C_{(n-1)}^k \cdot x^{n-1-k} \cdot y^k \quad (9)$$

тогда нужно из формулы (9) вывести формулу (8)

сделайте это!

Бином Ньютона полезен для доказательства разных формул.

Например подставим $x = 1$ и $y = 1$ в формулу (8). Получим:

$$\begin{aligned}(1 + 1)^n &= \sum_{k=0}^{k=n} C_n^k \cdot 1^{n-k} \cdot 1^k \\ 2^n &= \sum_{k=0}^{k=n} C_n^k\end{aligned}\tag{10}$$

Введение в теорию графов

Теория графов... очень обширная.

И при этом достаточно простая с большим количеством качественных материалов.

Мы эту часть пройдем полностью заочно. Наша цель – получением минимальных навыков, чтобы «убрать страх» и в будущем достаточно быстро решать любые практические задачи из теории графов

Ниже – темы для самостоятельного исследования.

На каждую часть нужно потратить 1-2, может быть 3 вечера, не более.

Поиск материалов – часть задания. (Если не находите удобный и понятный – спрашивайте в чате)

По каждому пункту – удобный ДЛЯ ВАС конспект от руки !

Часть 1. Определения, классические задачи

1. Формальное математическое определение графа:
 $G = (V, E)$, V – множество вершин, E – множество рёбер.
Ориентированный и неориентированный графы.
Представление графа таблицей. Связный граф.
Циклический и ациклический граф
2. Дерево
3. Двудольный граф
4. Классическая «задача о Кёненгсбергских мостах»
5. Классическая «задача о четырёх красках»

Простая задача на графы «Кто пьёт воду и кто держит зебру?»

На улице стоят пять домов, расположенные в ряд друг за другом.

Каждый из пяти домов окрашен в свой цвет, их жители имеют различное гражданство, владеют разными животными, пьют разные напитки и майнят разные криптовалюты.

1. Англичанин живёт в красном доме.
2. У испанца есть собака.
3. В зелёном доме пьют кофе.
4. Украинец пьёт чай.
5. Зелёный дом стоит сразу справа от белого дома.
6. Тот, кто майнит Bitcoin, разводит улиток.
7. В жёлтом доме майнят Ethereum.
8. В центральном доме пьют молоко.
9. Норвежец живёт в первом доме.
10. Сосед того, кто майнит Stellar, держит лису.
11. В доме по соседству с тем, в котором держат лошадь, майнят Ethereum.
12. Тот, кто майнит IOTA, пьёт апельсиновый сок.
13. Японец майнит Monero.
14. Норвежец живёт рядом с синим домом.

Вопросы

1. Кто держит зебру?
2. Кто пьёт воду?

Часть 2. Графы как структуры данных

1. Двоичное дерево поиска (binary search tree, BST)
2. Код Шенона-Фано;
3. Код Хаффмана.
4. Алгоритм Дейкстры

По просьбе трудящихся каждую эту тему можем рассмотреть отдельно. Собирайте инициативные группы, пишите в чат.

Часть 3*. Практическое решение задачи Коммивояжёра

1. P задачи, NP задачи и NP полные задачи
2. Задача Коммивояжёра
3. Метод ветвей и границ
4. Алгоритм Литтла

Домашняя работа

1. заведите толстую тетрадь в клетку, удобную ручку и маркеры.
2. прочитайте ещё раз лекцию. Перепишите формулы.
3. задания, выделенные **ЭТИМ ЦВЕТОМ** решите в тетради (ручкой по бумаге!). Если трудно, дождитесь результатов других ребят и спишите ПО УМНОМУ. Если не понятно – напишите в личку тому, кто решил
4. сфоткайте отсканируйте и выложите это в топик. Помогите 1-2 ребятам, кто к вам обратился. После – перенаправьте к тем, кому уже объяснили
5. на вопросы для самопроверки ответьте усно. Там где нужно что-либо считать – решите на бумаге, потом перенесите в тетрадь.
6. всё что выходит за рамки курса выделено **ЭТИМ ЦВЕТОМ**. В конце тетрадки сделайте главу OutOf Scope. И просто перепишите названия. Прочитайте в википедии О ЧЁМ это, чтобы просто иметь хоть какое-то представление. В будущем – разберётесь более предметно.

Задачи

Задачи желательно попробовать решить самому. Если не получается – посмотреть у товарища, но ПРОРАБОТАТЬ их ВСЕ!

Докажите *свойство симметрии* биномиальных коэффициентов, то есть:

$$C_n^k = C_n^{n-k} \quad (11)$$

Докажите *рекуррентное соотношение*:

$$C_n^k = C_{n-1}^{k-1} + C_{n-1}^k \quad (12)$$

Если n четное, то используя Бином Ньютона докажите:

$$C_n^0 + C_n^2 + C_n^4 + C_n^6 + \dots + C_n^n = C_n^1 + C_n^3 + \dots + C_n^{n-1} \quad (13)$$

Если n – нечётное, то как измениться формула (13)?

Задача (*) Свёртка Вандермонда

Опциональное ДЗ для особо продвинутых.

Попробуйте доказать Свёртку Вандермонда:

$$\sum_{j=0}^{j=n} C_n^j C_m^{k-j} = C_{n+m}^k \quad (14)$$

Список материалов

Методичка: «Элементы комбинаторики» (А.Е.Жуков, Д.А.Жуков, 2014)

Калькулятор биномиальных коэффициентов: <https://www.omnicalculator.com/math/binomial-coefficient>

Стандартная библиотека Python itertools: <https://docs.python.org/3/library/itertools.html>