Семинар 1
Задачи по математической статистике

Артемов А. В., Кондратьева Е. А.

8 сентября 2018 г.

1 Комбинаторика

Теория.

Правило сложения (правило «или») — одно из основных правил комбинаторики, утверждающее, что, если элемент А можно выбрать n способами, а элемент В можно выбрать m способами, то выбрать А или В можно $n + m$ способами.

Правило умножения (правило «и») — если элемент А можно выбрать n способами, и при любом выборе А элемент В можно выбрать m способами, то пару (А, В) можно выбрать $n \times m$ способами.

Размещения - размещением (из n по k) называется упорядоченный набор из k различных элементов из некоторого множества различных n элементов.

Число размещений из n по k, обозначаемое A_n^k равно убывающему факториалу:

$$A_n^k = \frac{n!}{(n-k)!}$$

При $k = n$ количество размещений равно количеству перестановок порядка n:

$$A_n^n = P_n = n!$$

По правилу умножения количество размещений с повторениями из n по k, обозначаемое A_n^-k, равно:

$$A_n^-k = n^k$$

Бинарное дерево поиска (англ. binary search tree, BST) — структура данных для работы с упорядоченными множествами. Бинарное дерево поиска обладает следующим свойством: если x — узел бинарного дерева с ключом k, то все узлы в левом поддереве должны иметь ключи, меньшие k, а в правом поддереве большие k.

Для каждого узла n в бинарном дереве поиска верны следующие утверждения:

- $n \leqslant l$, где l - значение потомка слева;
- $n \geqslant r$, где r это значение потомка справа;
- Левые и правые ветви дерева (поддеревья) также являются бинарными деревьями поиска.

Вычислительная (временная) сложность двоичного дерева поиска:

- $O(n)$ - расход памяти (в среднем случае);

- $O(\log n)$ - поиск (в среднем случае);
- $O(\log n)$ - удаление элемента (в среднем случае);
- $O(\log n)$ - добавление элемента (в среднем случае);

Сочетания - в комбинаторике сочетанием из n по k называется набор k элементов, выбранных из данного множества, содержащего n различных элементов.

Наборы, отличающиеся только порядком следования элементов (но не составом), считаются одинаковыми, этим сочетания отличаются от размещений.

Число сочетаний из n по k равно биномиальному коэффициенту:

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

Задача 1. Регистрационные номерные знаки Российской Федерации в пределах одного субъекта кодируются серией из трех букв и трех цифр. Буквы означают серию номерного знака, а цифры — номер. ГОСТом для использования на знаках разрешены 12 букв кириллицы, имеющие графические аналоги в латинском алфавите. Также, используются цифры от 0 до 9, причем, номера из трёх нулей быть не может. Определите общее количество комплектов регистрационных знаков, которое может быть изготовлено для каждого субъекта России.

Решение.
Исходя из условия, в рамках одного фиксированного цислового номера возможны 12^3 комбинаций букв, а в рамках одной фиксированной комбинации букв возможны $(10^3 - 1)$ комбинация цифр. Таким образом, общее колличество комплектов составляет $12^3 \times (10^3 - 1) = 1$ млн 726 тыс. 272 знака.

□

Задача 2. Подсчитать количество бинарных деревьев поиска, которые являются на самом деле односвязными списками (в каждом узле - по одному потомку):

(a) длиной 3 (узлы со значениями $1, 2, 3$);
(b) длиной 5 (2 узла со значением 0, 3 узла со значением 1).

Решение.

(a) Для BST длиной в 3 звена, в которых каждый узел имеет не более одного потомка, есть 3! способов упорядочения элементов. В зависимости от порядка, будет изменено размещение левых или правых потомков: в $1-2-3$, будут два правильных потомка, но в порядке $2-1-3$, получаем как левый, так и правый потомок элемента 2. Таким образом, $2-1-3$ и $2-3-1$ не являются вырожденными, в отличие от остальных комбинаций, так что есть $3!-2=4$ таких вырожденных дерева.

(b) Когда размер дерева увеличивается, вероятность того, что случайно выбранное дерево вырождается, становится исчезающе малой. Существует $5! = 120$ возможный комбинаций. Однако любая перестановка нулей в окончательном ответе не имеет значения, и аналогично для для единиц, поэтому конечный результат равен $5!/(2! \times 3!) = 10$.

□

Задача 3. У вас есть 6 книг, и вы хотите выбрать 3 из них. Однако, две из 6 книг - это разные издания одной и той же книги, и вы не хотите выбрать их вместе. Сколько существует вариантов выбора трех книг, которые соответсвуют данным условиям?

Решение. Всего существует $\binom{6}{3} = 20$ сочетаний из 3 книг, выбранных из данных 6-ти. Принимая во внимания 2 книги, которые являются изданиями одной (1-ое и 2-ое издание), можем разделить задачу выбора на 3 возможных случая:

(a) Случай, когда выбраны 1-ое издание и две другие книги, таких сочетаний возможно $\binom{4}{2}$;

(b) Случай, когда выбраны 2-ое издание и две другие книги, таких сочетаний возможно $\binom{4}{2}$;

(c) Случай, когда не выбрано ни одно из изданий, таких сочетаний возможно $\binom{4}{3}$.

Таким образом, общее количество сочетаний: $2 \times \binom{4}{2} + \binom{4}{3} = 16$. □

2 Вероятность

Теория.

Аксиомы вероятности:

- Аксиома 1: $0 \leqslant P(E) \leqslant 1$;
- Аксиома 2: $P(S) = 1$;
- Аксиома 3: Если E и F взаимоисключающие события ($E \cap F = \oslash$), then $P(E) + P(F) = P(E \cup F)$;

Для любой последоветльности взаимоисключающих событий $E_1, E_2, ...$

$$P(\bigcup_{i=1}^{\infty} E_i) = \sum_{i=0}^{\infty} P(E_i)$$

где $\cap$ – пересечение (произведение) событий, $\cup$ – объединение событий.
Определение 1. Пространство выборки - это совокупность всех возможных результатов эксперимента.
Примеры некоторых пространств выборки:

- При переворачивании монеты пространство выборки: $S = \{H, T\}$;
- При переворачивании двух разных монет пространство выборки:
  $S = \{(H; H); (H; T); (T; H); (T; T)\}$;
- При подбрасывании кубика пространство исходов: $S = \{1; 2; 3; 4; 5; 6\}$;
- Пространства выборки не обязательно должны быть конечными. Например, количество писем, отправленных за день: $S = \{1; 2; 3; 4; 5; ...\}$;
- Они также могут быть плотными наборами. Например, количество часов, потраченных на просмотр видео на Youtube за день: $S = \{x | x \in R, 0 \leqslant x \leqslant 24\}$.

Примеры некоторых событий:

- Монета перевернулась орлом: $E = \{H\}$;
- При переворачивании двух разных монет выпало более $\geqslant 1$ орла: $E = \{(H;H);(H;T);(T;H)\}$;
- При подбрасывании кубика получили $E = \{1;2;3;\}$;
- Пространства выборки не обязательно должны быть конечными. Например, количество писем, отправленных за день, равно $E = \{1;2;3;...;20\}$.

Условная вероятность - вероятность события E возникает при условии, что какое-то другое событие F уже произошло. Выражается, как $P(E|F)$.

$$P(E|F) = \frac{P(EF)}{P(F)}$$

В этом случае пространство выборки сводится к тем параметрам, которые соответствуют F или $S \cap F$, а пространство событий таким же образом сводится к $E \cap F$. Таким образом, в случае одинаково вероятных результатов $P(E|F) = |E{\cap}F|/|S{\cap}F| = |E{\cap}F|/|F|$, $F \subset S$.

Если $P(F) = 0$, то условная вероятность не определена, поскольку утверждение: $P(E)$, учитывая, что F произошло не имеет смысла, когда F невозможно.

Правило цепи, также известное, как правило умножения:

$$P(E_1,E_2,E_3...E_n) = P(E_1)P(E_2|E_1)P(E_3|E_2E_1)...P(E_n|E_1E_2...E_{n-1})$$

Или другая форма записи:

$$\mathrm{P}(\cap_{i=1}^{n} E_i) = \prod_{i=1}^{n} P(E_i|\cap_{j=1}^{i-1} E_j)) = P(E_1)p(E_2|E_1)P(E_3|E_1 \cup E_2)...P(E_n|E_1 \cap ... \cap E_{n-1})$$

Задача 1. Какова стойкость пароля (pincode для разблокировки) iPhone, учитывая, что в нем 4 цифры (a). Сколько понадобится комбинаций, чтобы расшифровать пароль, если на экране остались отпечатки пальцев на 4 (b) и 3 местах (c). В последнем случае, это значит, что 1 цифра пароля повторяется дважды.

Решение.

(a) Для четырехзначных паролей возможно $10^4 = 10000$ комбинаций с повторами;

(b) Если известны 4 цифры, которые используются единажды в пароле, число возможный комбинаций $4! = 24$;

(c) Когда известно, что одна из цифр в четырехзначном пароле повторяется (обозначим три используемые цифры, как a, b и c), достаточно найти количество комбинаций для повтора одной из них и умножить на 3. Допустим, посторяется c, тогда из 4 цифр в пароле, нам нужно выбрать ещё 2, помимо повторяющейся цифры c, то есть $4!/2! = 12$. Умножая полученное значение на 3, получаем $12 \times 3 = 36$;

Таким образом, можно отметить, что пароль с одним повтором немного надежнее, чем без повторов.

□

Задача 2. Какова вероятность того, что на вечеринке из n человек нет двух людей, которые родились в один день . Подсчитать для вечеринок размером в $n = [23, 75{,}100, 150]$ человек, вне зависимости от года рождения. Определить вероятность, что на вечеринке нет человека, который родился с Вами в 1 день, для $n = [23{,}190{,}253]$, вне зависимости от года рождения.

Решение.

$$|S| = (365)^n, |E| = (365)(364)...(365 - n + 1)$$

$$P(no\ match) = (365)(364)(365 - n + 1)/(365)^n$$

$$n = 23 : P(no\ match) < 0.5$$

$$n = 75 : P(no\ match) < 0.00033(3)$$

$$n = 100 : P(no\ match) < 1/3000000$$

$$n = 150 : P(no\ match) < 1/3000000000000000$$

Тогда вероятность того, что на вечеринке нет человека у которого с Вами один день рождения:

$$|S| = (365)^n, |E| = (364)^n$$

$$P(no\ match) = (364)^n/(365)^n$$

$$n = 23 : P(no\ match\ with\ yours) \approx 0.938$$

$$n = 190 : P(no\ match\ with\ yours) \approx 0.5938$$

$$n = 253 : P(no\ match\ with\ yours)\ \approx 0.4995$$

□

3 Формула Байеса

Теория.

Задача 1. Рассмотрим тест на ВИЧ, известно что он эффективен в 98% и имеет частоту ложноположительных результатов 1%. Известно, что 1 человек из 200 в США ВИЧ-положителен, посчитать вероятность события E, что пациент получил положительный результат теста, когда F вероятность того, что у пациента действительно ВИЧ. Тогда вероятность $P(E|F)$ или вероятность истинно позитивного результата:

Решение.

$$\mathsf{P}(F|E) = \frac{\mathsf{P}(E|F)\,\mathsf{P}(F)}{\mathsf{P}(E|F)\,\mathsf{P}(F) + \mathsf{P}(E|\sim F)P(\sim F)}$$

Где $\mathsf{P}(E|F) = 0.98$, $\mathsf{P}(E|\ F) = 0.01$, $\mathsf{P}(\ E|\ F) = 0.99$, $\mathsf{P}(\ E|F) = 0.02$.

$$\mathsf{P}(F|E) = \frac{(0.98)(0.005)}{(0.98)(0.005) + (0.01)(1 - 0.005)} \approx 0.330$$

Интересно заметить, что несмотря на то, что тест имеет такую высокую точность, вероятность получить истинно положительный результат не столь велика. Это обсуловлено небольших колличеством пациентов с ВИЧ, по отношению ко всй популяции, поэтому вероятность ложно положительного результата более вероятна, чем истинно-позитивного.

□

Задача 2. Кроме того, известно, что 60% всех почтовых сообщений — спам. Из сообщений, являющихся спамом 90% содержат подозрительный заголовок. Из сообщений, не являющихся спамом, "подозрительный"заголовок содержат 20% писем. Подсчитать вероятность того, что почтовое сообщение является спамом, при условии, что оно содержит поддельный заголовок $(P|E)$.

Решение.

$$\mathsf{P}(F|E) = \frac{\mathsf{P}(E|F)\,\mathsf{P}(F)}{\mathsf{P}(E|F)\,\mathsf{P}(F) + \mathsf{P}(E|\sim F)P(\sim F)}$$

$$\mathsf{P}(F|E) = \frac{(0.9)(0.6)}{(0.9)(0.6) + (0.2)(0.4)} \approx 0.871$$

4 Последовательность независимых испытаний

Задача 1. Компьютер генерирует последовательность бит, причем каждый бит является единицей с вероятностью p и нулем с вероятностью $1-p$. Подсчитать вероятность того, что первые n бит последовательности — единицы, а $n+1$-й бит — нуль.

Решение. Пусть $b_i \in \{\mathtt{0}, \mathtt{1}\}$ – значение i-того бита. Искомая вероятность

$$\mathrm{P}(b_1 = \mathtt{1}, b_2 = \mathtt{1}, \ldots, b_n = \mathtt{1}, b_{n+1} = \mathtt{0}) = \mathrm{P}(b_1 = \mathtt{1})\mathrm{P}(b_2 = \mathtt{1})\cdots\mathrm{P}(b_n = \mathtt{1})\mathrm{P}(b_{n+1} = \mathtt{0}),$$

поскольку испытания независимы. Так как согласно условию $\mathrm{P}(b_i = \mathtt{0}) = 1-p$, $\mathrm{P}(b_i = \mathtt{0}) = p$, $i = 1, 2, \ldots$, то в результате

$$\mathrm{P}(\underbrace{\mathtt{11}\ldots\mathtt{1}}_{\text{n раз}}\mathtt{0}) = p^n(1-p).$$

□

Задача 2. m строк добавляются в хеш-таблицу, содержащую n корзинок, так, что вероятность попасть в каждую из корзинок одинаковая. Подсчитать вероятность, что после добавления всех строк первая корзинка останется пустой.

Решение. Пусть событие E заключается в том, что в первую корзинку захешировалась хотя бы одна строка. (Мы ищем, таким образом, $1 - \mathrm{P}(E)$.) Обозначим F_i событие, которое заключается в том, что строка i *не* захешировалась в первую корзинку ($i \in \{1, \ldots, m\}$). Вероятность этого события $\mathrm{P}(F_i) = 1 - \frac{1}{n}$. Тогда событие $\cap_{i=1}^m F_i$ соответствует тому, что ни одна из строк не захешировалась в первую корзинку, причем вероятность $\mathrm{P}(\cap_{i=1}^m F_i)$ этого события соответствует $1 - \mathrm{P}(E)$:

$$\mathrm{P}(E) = 1 - \mathrm{P}(\cap_{i=1}^m F_i) = 1 - \prod_{i=1}^m \mathrm{P}(F_i) = 1 - \left(1 - \frac{1}{n}\right)^m.$$

□

5 Независимость

Задача 1. Два компьютера соединены n роутерами, включенными параллельно, причем вероятность безотказной работы каждого из которых p_i. Подсчитать вероятность существования работающего пути между компьютерами.

Решение. Пусть E – событие, которые заключается в том, что между компьютерами существует работающий путь (мы ищем $\mathrm{P}(E)$.) Тогда

$$\begin{aligned}\mathrm{P}(E) &= 1 - \mathrm{P}(\text{все роутеры сломаны}) = \\ &= 1 - \prod_{i=1}^n \mathrm{P}(\text{роутер } i \text{ сломан}) = 1 - \prod_{i=1}^n \mathrm{P}(1 - p_i).\end{aligned}$$

□

6 Дискретные распределения

6.1 Биномиальное распределение

Задача 1. n бит пересылаются по сети, причем вероятность для каждого бита инвертироваться при пересылке равна p. Подсчитать вероятность получения корректируемого сообщения, если для коррекции используется 3 бита, и допустима лишь одна ошибка при пересылке.

Теория. Задача связана с применениями т.н. корректирующих кодов (error-correcting codes, коды Хэмминга) для передачи сообщений по сетям с помехой. Простая интуиция, связанная с корректирующими кодами, заключается в следующем[1]. Предположим, что мы хотим передать сообщение из одного бита – числа `0`, и существует некоторая вероятность $p < 1$ инверсии бита при передаче. Если мы вместо одного бита `0` будем посылать, например, три `000` и считывать полученное сообщение как `0`, если в нем большинство нулей (т.е. сообщения `000`, `001`, `010`, `100` считаются как `0`), то вероятность получить ошибочное сообщение (два и более бита инвертировались) будет равна $p^3 + C_3^2 p^2 (1-p)$, что меньше p, таким образом, надежность канала передачи повышена (за счет передачи избыточной информации).

Чуть более общая ситуация заключается в следующем (Hamming(7,4)[2]). Например, пусть у нас есть какое-то сообщение длиной 4 бита. Пусть к этому сообщению

[1] Соответствующий пример можно найти на Википедии: https://en.wikipedia.org/wiki/Error_correction_code#How_it_works

[2] Для более полного понимания всей общей процедуры призываем ознакомиться с материалом https://en.wikipedia.org/wiki/Hamming(7,4)

мы добавим 3 т.н. бита четности (parity bits). Наша цель будет состоять в том, чтобы выбрать эти биты четности так, чтобы получить 3 множества по 4 бита с четным числом единиц. Если при этом какой-то один бит инвертировался при передаче, его можно найти (и даже скорректировать), взяв пересечение множеств с нечетным числом единиц и дополнения множеств с четным числом единиц (т.е. корректных множеств).

Решение. Пусть X – число инвертированных при пересылке бит. Т.к. биты инвертируются независимо, то $X \sim \mathrm{Bin}(n+3, p)$. Поскольку согласно условию возможно скорректировать одну ошибку при пересылке, то допустимо появление 0 или 1 инверсии бит в передаваемом сообщении. Вероятность этого события равна

$$\mathrm{P}(X=0) + \mathrm{P}(X=1) = (1-p)^{n+3} + \frac{1}{n+3} p(1-p)^{n+2}.$$

Если, например, $n = 4$ и каждый бит инвертируется с вероятностью $p = 0.1$ (что значительно больше, чем происходит в практике), то $X \sim \mathrm{Bin}(7, 0.1)$ и можно подсчитать, что при использовании корректирующих кодов $\mathrm{P}(X \leqslant 1) \approx 0.8503$. При этом, без их использования $X \sim \mathrm{Bin}(4, 0.1)$, и $P(X = 0) \approx 0.6561$.

□

Задача 2. В США во время второй мировой войны всех призывников подвергали медицинскому обследованию. Реакция Вассермана позволяет обнаруживать в крови больных сифилисом определенные антитела. Для это смешиваются пробы крови k человек, если проба положительная, каждого человека из этой группы следует проверить и совершить $k+1$ измерений. Количество испытаний - n, вероятность успеха - p.

Решение.
Допустим, что n делится нацело на k. Тогда нужно проверить $n \div k$ групп обследуемых. Пусть X_j - количество проверок, потребовавшихся в j-й группе, $j = 1, ..., n/k$. Тогда

$$X_j = \begin{cases} 1, & \text{с вероятностью } (1-p)^k \text{ все } k \text{ человек здоровы,} \\ k+1, & \text{с вероятностью } 1-(1-p)^k \text{ есть больные,} \end{cases}$$

Обозначим общее число проверок $X_1 + \ldots + X_{n/k}$ через Z. Задача заключается в том, как для заданного значения p определить размер группы $k_0 = k_0(p)$, минимизирующий $\mathrm{E}\, Z$. Имеем

$$\mathrm{E}\, X_j = 1 \cdot (1-p)^k + (k+1)[1-(1-p)^k] = k + 1 - k(1-p)^k.$$

Отсюда по свойствам матожидания

$$\mathrm{E}\, Z = \mathrm{E}\, X_1 + \ldots + \mathrm{E}\, X_{n/k} = n[1 + 1/k - (1-p)^k].$$

Положим $H(x) = 1 + 1/x - (1-p)^x$ при $x > 0$.

Для близких к нулю значений p минимум $H(x)$ достигается в точке x_0, где x_0 – наименьший из корней уравнения $H'(x) = 0$, т.е. уравнения

$$\frac{1}{x^2} + (1-p)^x \log(1-p) = 0$$

Его нельзя разрешить явно относительно x. Поэтому, используя формулу $(1-p)^x \approx 1 - px$ при малых p, заменим $H(x)$ на функцию $G(x) = 1 + 1/x - 1 + px = 1/x + px$, имеющую точку минимума $\tilde{x}_0 = 1/\sqrt{p}$, причем $G(\tilde{x}_0) = 2\sqrt{p}$. Для $p = 0.01$ получаем $\tilde{x}_0 = 10$ и $G(\tilde{x}_0) = 1/5$, т.е. $\mathrm{E}\, Z \approx n/5$.

6.2 Распределение Пуассона

Задача 1. n ($\gg 1$) бит пересылаются по сети, причем вероятность для каждого бита инвертироваться при пересылке равна p ($\ll 1$). Подсчитать вероятность получения сообщения, не содержащего ошибок, используя пуассоновское приближение биномиального распределения.

Теория. См. задачу 6.1.

Напомним, что распределение Пуассона задается функцией

$$\mathrm{P}(X = n) = \frac{e^{-\lambda}\lambda^k}{k!},$$

где λ – параметр распределения (он же среднее, он же дисперсия).

Пуассоновское распределение разумно использовать вместо биномиальной $\mathrm{Bin}(n, p)$, когда число испытаний n очень велико, а вероятность успеха p крайне мала. Например, в сетях передачи данных, как правило, передаются строки большой длины ($n \sim 10^4$), а вероятность инверсии бита в них очень низка ($p \sim 10^{-6}$).

Решение. Пусть X - число инвертированных при пересылке бит. Т.к. биты инвертируются независимо, то $X \sim \mathrm{Pois}(\lambda)$, где $\lambda = np$. Вероятность безошибочной передачи сообщения при этом

$$\mathrm{P}(X = 0) = \frac{e^{-\lambda}\lambda^0}{0!} = e^{-\lambda}.$$

Если, например, $n = 10^4$ и каждый бит инвертируется с вероятностью $p = 10^{-6}$ $X \sim \mathrm{Pois}(0.01)$ и $\mathrm{P}(X = 0) = e^{-0.01} = 0.990049834$. Кстати, без пуассоновского приближения (когда $X \sim \mathrm{Bin}(10^4, 10^{-6})$ имеем $\mathrm{P}(X = 0) = 0.990049829$, т.е. погрешность приближения порядка 5×10^{-9}.

□

Задача 2. Подсчитать вероятность того, что из 10 выпущенных компьютерных чипов будет не более одного бракованного, если вероятность выпустить бракованный чип равна 0.1, а чипы производятся независимо.

Решение. Пусть X – число выпущенных бракованных чипов. Согласно условию приближенно $X \sim \mathrm{Pois}(1)$ и

$$\mathrm{P}(X = 0) + \mathrm{P}(X = 1) = \frac{e^{-1}1^0}{0!} + \frac{e^{-1}1^1}{1!} = 2e^{-1} \approx 0.7358.$$

□

Задача 3. Пусть число запросов к веб-серверу в день имеет распределение Пуассона с параметром λ. Каждый из запросов задается либо человеком (с вероятностью p), либо ботом (с вероятностью $1-p$). Показать, что числа запросов в день от людей и от ботов суть независимые пуассоновские случайные величины (и подсчитать их распределения).

Решение. Пусть $N \sim \mathrm{Pois}(\lambda)$ – число запросов к веб-серверу в день, а X и Y – суть количества запросов от людей и ботов, соответственно. Условное распределение

$\mathrm{Law}(X|N) \sim \mathrm{Pois}(N, p)$ (сделано N независимых запросов, вероятность «успеха» (запрос сделал человек) равна p). Аналогично $\mathrm{Law}(Y|N) \sim \mathrm{Pois}(N, 1-p)$. Рассмотрим совместное распределение (условие выпишем формально)

$$\begin{aligned} \mathrm{P}(X = i, Y = j) = \mathrm{P}(X = i, Y = j | X + Y = i + j)\mathrm{P}(X + Y = i + j) + \\ \mathrm{P}(X = i, Y = j | X + Y \neq i + j)\mathrm{P}(X + Y \neq i + j) = \\ \mathrm{P}(X = i, Y = j | X + Y = i + j)\mathrm{P}(X + Y = i + j). \end{aligned}$$

Тогда $\mathrm{P}(X = i, Y = j | X + Y = i + j) = C_{i+j}^i p^i (1-p)^j$, т.к. это мультиномиальное распределение, и $\mathrm{P}(X + Y = i + j) = e^{-\lambda}\lambda^{i+j}/(i+j)!$, т.к. это по условию пуассоновское распределение, поэтому в итоге

$$\begin{aligned} \mathrm{P}(X = i, Y = j) &= C_{i+j}^i p^i (1-p)^j e^{-\lambda} \frac{\lambda^{i+j}}{(i+j)!} = \\ &= e^{-\lambda} \frac{(\lambda p)^i}{i!} \frac{(\lambda(1-p))^j}{j!} = \\ &= e^{-\lambda p} \frac{(\lambda p)^i}{i!} e^{-\lambda(1-p)} \frac{(\lambda(1-p))^j}{j!} = \\ &= \mathrm{P}(X = i)\mathrm{P}(Y = j). \end{aligned}$$

Как и ожидалось, $X \sim \mathrm{Pois}(\lambda p)$, $Y \sim \mathrm{Pois}(\lambda(1-p))$.

□

7 Непрерывные распределения

7.1 Экспоненциальное распределение

Задача 1. Пусть время до поломки жесткого диска распределено экспоненциально с параметром $\lambda > 0$. Подсчитать вероятность того, что жесткий диск сломается в течение 10 дней после начала эксплуатации.

Теория. Экспоненциальное распределение показывает, через какое время произойдет то или иное событие (землетрясение, запрос на веб-сервер, поломка жесткого диска и т.д.). Если $X \sim \mathrm{Exp}(\lambda)$, то

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geqslant 0, \\ 0, & x < 0. \end{cases}$$

Довольно важен часто вопрос вида: чему равна вероятность $\mathrm{P}(X > t + s | X + s)$? Иными словами, если жесткий диск уже прослужил s лет, какие шансы, что он еще прослужит t лет?

$$\begin{aligned} \mathrm{P}(X > t + s | X > s) &= \frac{\mathrm{P}(X > t + s, X > s)}{\mathrm{P}(X > s)} = \frac{\mathrm{P}(X > t + s)}{\mathrm{P}(X > s)} = \\ &= \frac{\lambda e^{-\lambda(t+s)}}{e^{-\lambda s}} = e^{-\lambda t} = \mathrm{P}(X > t). \end{aligned}$$

Таким образом, процесс «памяти» не имеет.

Решение. Пусть $X \sim \mathrm{Exp}(\lambda)$ – время до поломки жесткого диска. Тогда

$$\mathrm{P}(X < 10) = 1 - e^{-10\lambda}.$$

Например, если $\lambda = 1$ год ($X \sim \text{Exp}(1/365)$, то $\text{P}(X < 10) = 1 - e^{-10/365} = 0.027$, а если $\lambda = 1$ месяц ($X \sim \text{Exp}(1/30)$, то $\text{P}(X < 10) = 1 - e^{-10/30} = 0.283$.

□

Задача 2. Подсчитать вероятность того, что посетитель некоторого сайта проведет на нем более 10 минут, если время, проводимое посетителями на сайте, является экспоненциально распределенной случайной величиной со средним, равным 5 минутам.

Решение. Пусть среднее время, проведенное посетителем на сайте – $X \sim \text{Exp}(\lambda)$, $\lambda = 1/5$. Тогда

$$\text{P}(X > 10) = 1 - (1 - e^{-10\lambda}) = e^{-10/5} = 0.865.$$

□

7.2 Нормальное распределение

Задача 1. По проводу передается сигнал X со значением -2 либо $+2$ вольта, причем -2 В означает логический ноль, а $+2$ В — логическую единицу. На другом конце провода принятый сигнал имеет вид $R = X+Y$, где Y — стандартно нормально распределенная случайная составляющая. Подсчитать вероятность ошибочного декодирования сигнала, если правило декодирования предписывает считать R логической единицей при $R > 0.5$, и логическим нулем при $R < 0.5$.

Решение.

$$\begin{aligned}
\text{P}(\text{ошибочное декодирование}) &= \text{P}(R > 0.5|X = -2) + \text{P}(R < 0.5|X = +2) = \\
&= \text{P}(Y - 2 > 0.5) + \text{P}(Y + 2 < 0.5) = \\
&= 1 - \Phi(2.5) + \Phi(< 1.5) \approx 0.0062 + 0.0668.
\end{aligned}$$

□

8 Алгоритмы

Теория. *QuickSort* – быстрый рекурсивный алгоритм сортировки. Выбор индекса основан на функции разбиения:

```
int Partition (int [] arr, int n) {
    int lh = 1, rh = n - 1;
    int pivot = arr [0];
    while (true) {
        while (lh < rh && arr [rh] >= pivot) rh --;
        while (lh < rh && arr [lh] < pivot) lh ++;
        if (lh == rh) break;
        Swap(arr [lh], arr [rh]);
  }
  if (arr [lh] >= pivot) return 0;
    Swap(arr[0], arr[lh]);
    return lh;
}
```

QuickSort производит сортировку со средней скоростью $O(n\ log\ n)$, но, в худшем случае, время работы увеличивается до $O(n^2)$, если на кажной итерации выбираются максимальный или минимальный элементы. Тогда вероятность того, что Quicksort будет работать максимально долго равна вероятности составления дегенеративных бинарных деревьев поиска. Пусть X это количество произведенных сравнений при сортировке n элементов, тогда $E[X]$ это - математическое ожидание продолжительности алгоритма. Примем за X_1,X_2,...,X_n входную последовательность для сортировки, а Y_1,Y_2,...,Y_n выходная отсортированная последовательность. Пусть I_a,$b = 1$ если Y_aY_b сравнены и 0, если нет. Тогда, согласно порядку, каждая пара сравнивается только один раз

$$X = \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} I_a, b$$

Тогда, для элементов Y_a и Y_b, если выбранный опорный элемент массива не между ними, элементы не будут сравнены между собой напрямую. Значит, мы рассматриваем только случаи, где опорный элемент лежит между ними Y_a,...,Y_b. Тогда, вероятность события сравнения двух элементов $2/(b-a+1)$ (с аппроксимацией).

$$\sum_{b=a+1}^{n} \frac{2}{b-a+1} \approx \int_{a+1}^{n} \frac{2db}{b-a+1} = 2\ln(n-a+1)|_{a+1}^{n} \approx 2\ln(n-a+1)$$

Задача 1. Известно, что в худшем случае скорость работы алгоритма QuickSort равна $O(n^2)$. Подсчитать вероятность того, что QuickSort отработает за $O(n^2)$, если входной массив случайно отсортирован.

Решение. `QuickSort` работает со средней скоростью $O(n\ log\ n)$, но, в худшем случае, время работы увеличивается до $O(n^2)$, если на каждой итерации выбираются максимальный или минимальный элементы. На каждом рекурсивном вызове элемент `pivot = [max, min]`, поэтому мы остаются с $n-1$ элементами на следующей итерации. Таким образом, возможны 2 «плохих» выбора за каждую итерацию:

$$\mathrm{P(worst\ case)} = \frac{2}{n} \cdot \frac{2}{n-1} \cdot ... \cdot \frac{2}{2} = \frac{2^{n-1}}{n!}$$

□

9 Математическое ожидание

Задача 1. В хеш-таблицу с n корзинками хешируются строки, причем вероятность при хешировании выбрать любую из корзинок одинакова. Подсчитать математическое ожидание числа строк, которые необходимо поместить в таблицу, чтобы каждая из корзинок содержала хотя бы одну строку.

Теория. Для этой задачи нам потребуется понятие *геометрического распределения.* Геометрическое распределение $\mathrm{Geo}(p)$ с вероятностью успеха p – это распределение, описывающее количество независимых испытаний, требуемых для достижения первого успеха, причем вероятность успеха в каждом испытании равна p. Случайная величина $X \sim \mathrm{Geo}(p)$ принимает значения $1, 2, \ldots$ с вероятностями $\mathrm{P}(X = 1) = (1-p)^{n-1}p$, $n = 1, 2, \ldots$, соответственно. При этом

$\mathrm{E}\,X = \frac{1}{p}, \mathrm{V}\,X = \frac{1-p}{p^2}$. Приложения: подбрасывание монетки до первого «орла», генерирование бит до первой единицы и т.п.

Решение. Обозначим X случайную величину, равную количеству строк, которые должны попасть в таблицу, чтобы вкаждая из корзинок содержала хотя бы одну строку. Рассмотрим схему испытаний, в которой «успехом» назовем заполнение корзинки, которая до этого была пустой. Тогда, если X_i – количество испытаний, которое требуется, чтобы получить i-тый «успех» после $(i-1)$-го. Так как после i-того «успеха» i корзинок имеют хотя бы одну строку, то то вероятность захешировать следующую строку в пустую корзинку равна $p = \frac{n-i}{n}$. Тогда

$$\mathrm{P}(X_i = k) = C_n^{n-i}\Big(\frac{i}{n}\Big)^{k-1} \Longleftrightarrow X_i \sim \mathrm{Geo}\Big(\frac{n-i}{n}\Big).$$

Отсюда $\mathrm{E}\,X_i = \frac{1}{p} = \frac{n}{n-i}$. Поскольку, естественно, $X = X_0 + X_1 + \dots + X_{n-1}$, то и $\mathrm{E}\,X = \mathrm{E}\,X_0 + \mathrm{E}\,X_1 + \dots + \mathrm{E}\,X_{n-1}$, то

$$\mathrm{E}\,X = \frac{n}{n} + \frac{n}{n-1} + \dots + \frac{n}{1} = n\Big(\frac{1}{n} + \frac{1}{n-1} + \dots + \frac{1}{1}\Big) = O(n \log n).$$

□

Задача 2. На кластер из k веб-серверов поступают http-запросы, причем вероятность того, что запрос будет обработан i-тым сервером, равна p_i, а запросы обрабатываются независимо. Подсчитать математическое ожидание и дисперсию числа серверов, обработавших хотя бы один запрос, после обработки n запросов.

Решение. Пусть событие A_i означает, что i-тый сервер не получил ни одного запроса из n обработанных, X – число событий вида A_i, а $Y = k - X$ – количество машин, которые на самом деле выполняли какую-то работу. Здесь используем, что т.к. формально $X = \sum\limits_{i=1}^{k} 1_{A_i}$ – сумма индикаторов, то $\mathrm{E}\,X = \sum\limits_{i=1}^{k} \mathrm{P}(A_i)$. Поскольку запросы независимы, то $\mathrm{P}(A_i) = (1-p_i)^n$, и

$$\mathrm{E}\,Y = k - \mathrm{E}\,X = k - \sum_{i=1}^{k} \mathrm{P}(A_i) = k - \sum_{i=1}^{k} (1-p_i)^n.$$

Что касается дисперсии $\mathrm{V}\,Y$, то $\mathrm{V}\,Y = \mathrm{V}\,X$. Т.к. события A_i, A_j независимы при $i \neq j$, то $\mathrm{P}(A_i \cap A_j) = (1-p_i-p_j)^n$, поэтому

$$\mathrm{E}[X(X-1)] = \mathrm{E}[X^2] - \mathrm{E}[X] = 2\sum_{i<j} \mathrm{P}(A_i \cap A_j) = 2\sum_{i<j} (1-p_i-p_j)^n.$$

Тогда дисперсия (здесь $\mathrm{V}\,X = \mathrm{E}[X^2] - \big(\mathrm{E}[X]\big)^2$)

$$\begin{aligned} \mathrm{V}\,X &= 2\sum_{i<j}(1-p_i-p_j)^n + \mathrm{E}[X] - \big(\mathrm{E}[X]\big)^2 = \\ &= 2\sum_{i<j}(1-p_i-p_j)^n + \sum_{i=1}^{k}(1-p_i)^n - \Big(\sum_{i=1}^{k}(1-p_i)^n\Big)^2. \end{aligned}$$

□

10 Центральная предельная теорема

Задача 1. Подсчитать число запусков некоторого алгоритма, необходимое для того, чтобы оценка среднего времени его работы принадлежала интервалу $[\mu - 0.5, \mu + 0.5]$ с 95% вероятностью, если среднее время его работы равняется μ секундам, а дисперсия времени его работы — 4 сек^2.

Теория. Нестрогое утверждение, связанное с ЦПТ, заключается в том, что если у вас есть n н.о.р. случайных величин $X_1, X_2, \dots, X_n$, причем $\mathrm{Law}(X_i) = F$, $\mathrm{E}_F X_i = \mu$, $\mathrm{E}_F X_i^2 = \sigma^2$, то тогда

$$\frac{X_1 + \dots + X_n - n\mu}{\sigma\sqrt{n}} \to \mathcal{N}(0,1) \qquad \text{при} n \to \infty.$$

Речь идет о сходимости по вероятности, т.е. форма эмпирического распределения выборочного среднего все больше и больше напоминает форму стандартного нормального распределения при увеличении размера выборки.

Решение. Пусть X_i – время работы алгоритма в ходе i-того запуска, а $\overline{X}_n = \frac{1}{n}\sum\limits_{i=1}^{n} X_i$ – выборочное среднее времен запуска. Тогда рассмотрим величину

$$Z_n = \frac{\sum_{i=1}^n X_i - n\mu}{\sigma\sqrt{n}} = \left|\sigma^2 = 4 \text{ сек.}, \mu\right| = \frac{\sum_{i=1}^n X_i - n\mu}{2\sqrt{n}} = \frac{\overline{X}_n - \mu}{2/\sqrt{n}}.$$

Согласно ЦПТ, Z_n – случайная величина, распределение которой приближенно *стандартное нормальное*. Нас интересует событие $A = \{\overline{X}_n \in [\mu - 0.5, \mu + 0.5\}$, причем $\mathrm{P}(A) \geqslant 0.95$. Учитывая связь двух величин Z_n и $\overline{X}_n$, выражаемую равенствами $Z_n = \frac{\sqrt{n}}{2}(\overline{X}_n - \mu)$ и $\overline{X}_n = \frac{2}{\sqrt{n}} Z_n + \mu$, запишем это неравенство в виде

$$\begin{aligned}
\mathrm{P}(\overline{X}_n \in [\mu - 0.5, \mu + 0.5) &= \mathrm{P}(-0.5 \leqslant \overline{X}_n - \mu \leqslant 0.5) = \\
&= \mathrm{P}\Big(-0.5 \leqslant \frac{2}{\sqrt{n}} Z_n \leqslant 0.5\Big) = \\
&= \mathrm{P}\Big(-0.5\frac{\sqrt{n}}{2} \leqslant Z_n \leqslant 0.5\frac{\sqrt{n}}{2}\Big) = \\
&= \Phi(\frac{\sqrt{n}}{4}) - \Phi(-\frac{\sqrt{n}}{4}) = \\
&= 2\Phi(\frac{\sqrt{n}}{4}) - 1 \geqslant 0.95.
\end{aligned}$$

Отсюда получаем, что мы должны иметь такое n, чтобы $\Phi(\frac{\sqrt{n}}{4}) \geqslant 0.975$ (это эффективно означает, что $\frac{\sqrt{n}}{4} \geqslant 2$), и можно подсчитать, что $n \geqslant 64$.

□

Задача 2. Используя центральную предельную теорему, подсчитайте вероятность того, что некоторый веб-сервер не справится с нагрузкой в следующую минуту, если сервер отказывает при обработке более 120 запросов в минуту, а число посетителей веб-сайта в минуту имеет распределение Пуассона с параметром 100.

Решение. Если бы нам надо было подсчитать точное решение этой задачи (напомним, что ЦПТ – аппроксимация!), то мы бы рассмотрели $X \sim \mathrm{Pois}(100)$ и нам необходимо было бы вычислить величину

$$\mathrm{P}(X \geqslant 120) = \sum_{i=120}^{\infty} \frac{e^{-100} 100^i}{i!} \approx 0.0282.$$

Но если мы хотим пользоваться ЦПТ, то нам надо понять, что случайная величина $X \sim \mathrm{Pois}(\lambda)$ – это *как будто* сумма большого числа (n) независимых случайных величин $X_1, \ldots, X_n$ с меньшей интенсивностью λ/n. Тогда $\mathrm{Pois}(100) \approx \sum\limits_{i=1}^{n} \mathrm{Pois}(100/n)$ и искомая вероятность может быть выражена как вероятность выброса для (приближенно) нормальной случайной величины X

$$\begin{aligned} \mathrm{P}(X \geqslant 120) = &= \mathrm{P}\Big(\frac{X-100}{\sqrt{100}} \geqslant \frac{119.5-100}{\sqrt{100}}\Big) = \\ &= 1 - \Phi(1.95) \approx 0.0256. \end{aligned}$$

В последнем равенстве принято $120 \approx 119.5$, чтобы получить 1.95 в аргументе функции ошибок.

□

11 Рекомендованная литература

1. Комбинаторика для начинающих. Автор: Московский физико-технический институт https://www.coursera.org/learn/kombinatorika-dlya-nachinayushchikh

2. Н.Я. Виленкин. Комбинаторика. – М.: Наука, 1969.

3. Н.Б. Алфутова, А.В. Устинов. Алгебра и теория чисел (сборник задач). – М.: МЦНМО, 2002.

4. А.М. Райгородский Комбинаторика и теория вероятностей. - МФТИ, 2012 - 109 с.

5. Д. Кнут, Р. Грэхем, О. Паташник. Конкретная математика. Математические основы информатики. М.: Мир, 1998