ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ФАКУЛЬТЕТ ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Расчётно-графическая работа № 1
по дисциплине «Теория вероятностей и продвинутая математическая статистика»

Выполнили студенты группы J3211:

Воробьев А.П.

ИСУ: 465440

Шакина А.С.

ИСУ: 465472

Преподаватель практики:

Кононов И.А.

Санкт-Петербург

2025

Задача 1

Постановка задачи

Требуется аналитически решить задачу. Помимо этого нужно написать программу, которая будет считать данные вероятности приблизительно с помощью метода Монте-Карло. Особо любознательные могут задаться поиском (аналитическим) оптимального количества итераций, которых будет достаточно для получения ответа с достаточно «разумной» точностью. При решении задач могут быть полезны принципы произведения вероятностей и включений-исключений.

Условие варианта 2

В городе живут $n+1$ людей. Человек, условный «прародитель», пишет два письма случайно выбранным адресатам, которые образуют «первое» поколение. Те делают то же самое, в результате чего образуется «второе поколение». В общем, на каждое полученное письмо горожанин готовит два ответа и отправляет двум случайным жителям. Найти вероятность того, что прародитель не входит ни в одно из поколений с номерами $1 \dots r$.

Аналитическое решение

Обозначения и допущения

Обозначения:

- $n+1$ - количество жителей города.
- r - количество поколений, которые мы рассматриваем.
- S_i - число входящих писем в поколении i.
- A - событие, что прародитель не является адресатом письма

Допущение: Отправитель письма не может быть его адресатом.

Решение

Рассмотрим i-е поколение. Каждое входящее письмо порождает два исходящих, которые отправляются двум случайным жителям города (кроме самого отправителя). Так как в первом поколении - $S_1 = 2$ письма, то во втором поколении будет $S_2 = 2 \cdot S_1 = 4$ и так далее. Таким образом, количество писем в i-м поколении можно выразить формулой:

$$S_i = 2 \cdot S_{i-1} = 2^i$$

Заметим, что в первом поколении не может быть прародителя, поэтому количество писем M за r поколений, адресатом для которого может быть прародитель, равно:

$$M = \sum_{i=2}^{r} S_i = \sum_{i=2}^{r} 2^i = 2^{r+1} - 4$$

Теперь найдем вероятность $P(A)$, что прародитель не является адресатом одного письма. Так как отправитель не может быть адресатом, то для каждого письма есть n возможных адресатов, из которых $n-1$ не являются прародителем. Значит:

$$P(A) = \frac{n-1}{n}$$

Так как все адресаты выбираются независимо, то вероятность $P(A_M)$, что прародитель не является адресатом ни одного из M писем:

$$P(A_M) = P(A)^M = \left(\frac{n-1}{n}\right)^{2^{r+1}-4}$$

Ответ

Таким образом, искомая вероятность, что прародитель не входит ни в одно из поколений с номерами $1 \dots r$, равна:

$$\boxed{P(A_M) = \left(\frac{n-1}{n}\right)^{2^{r+1}-4}}$$

Приближённое решение методом Монте-Карло

Решение методом Монте-Карло реализовано в блокноте 1.ipynb.

Задача 2

Постановка задачи

Тоже как и в первом упражнении нужно решить аналитически и написать скрипт, вычисляющий вероятность приближенно по методу Монте-Карло. Дополнительный вопрос о поиске числа итерация тоже имеет место быть.

Условие варианта 2

На отрезке длины l независимо одна от другой поставлены две точки, положение каждой из которых равновозможно на этом отрезке. Определить вероятность того, что из трех получившихся частей одного отрезка можно построить треугольник.

Аналитическое решение

Упрощения и обозначения

Обозначения:

- x, y - случайные величины, соответствующие координатам точек на отрезке.
- $u = min(x, y)$, $v = max(x, y)$ - упорядоченные координаты точек.
- $a = u$, $b = v - u$, $c = 1 - v$ - длины отрезков, на которые разбиваются исходный отрезок.
- A - событие, что из отрезков можно построить треугольник.

Упрощения:

- Без потери общности можно считать, что $l = 1$, так как вероятность не зависит от длины отрезка.
- Две точки на отрезке можно рассматривать, как координаты на единичном квадрате $[0, 1] \times [0, 1]$.
- Поскольку пары (x, y) и (y, x) приводят к одинаковым отрезкам, то можно рассматривать только половину квадрата, где $x \leq y$. Площадь этой половины равна $\frac{1}{2}$.

Решение

Из всего вышесказанного, мы имеем, что $x = u$, $y = v$, $0 \leq x \leq y \leq 1$, $a = x, b = y - x$, $c = 1 - y$. Событие A - соблюдение неравенств треугольника можно записать как:

$$\begin{cases} a + b > c \leftrightarrow x + (y - x) > 1 - y \leftrightarrow y > \frac{1}{2} \\ a + c > b \leftrightarrow x + (1 - y) > y - x \leftrightarrow 2x + 1 > 2y \\ b + c > a \leftrightarrow (y - x) + (1 - y) > x \leftrightarrow 1 - x > x \leftrightarrow x < \frac{1}{2} \end{cases}$$

С учётом ограничений $0 \leq x \leq y \leq 1$, получаем систему неравенств:

$$\begin{cases} 0 \leq x < \frac{1}{2} \\ \frac{1}{2} < y \leq 1 \\ x \leq y \\ y < x + \frac{1}{2} \end{cases}$$

Построим область, удовлетворяющую системе неравенств:

Найдём область S_A, удовлетворяющую системе неравенств.

$$S_A = \int_{x=0}^{x=\frac{1}{2}} \int_{y=\frac{1}{2}}^{y=x+\frac{1}{2}} dydx = \int_{x=0}^{x=\frac{1}{2}} ((x + \frac{1}{2}) - \frac{1}{2})dx = \int_{x=0}^{x=\frac{1}{2}} xdx = \frac{1}{8} - 0 = \frac{1}{8}$$

Таким образом, вероятность $P(A)$ равна отношению площади S_A к площади треугольной области, которая равна $\frac{1}{2}$:

$$P(A) = \frac{S_A}{\frac{1}{2}} = \frac{\frac{1}{8}}{\frac{1}{2}} = \frac{1}{4}$$

Ответ

Таким образом, искомая вероятность, что из трёх частей отрезка можно построить треугольник, равна:

$$\boxed{P(A) = \frac{1}{4}}$$

Приближённое решение методом Монте-Карло

Решение методом Монте-Карло реализовано в блокноте 2.ipynb.

Задача 3

Постановка задачи

Требуется аналитически решить задачу.

Условие варианта 2

Введём события $A_i = \{X = i\}, B_i = \{Y = i\}, i \geq 0$. Известно, что для любых $i \geq 0, j \geq 0$ события A_i и B_j – независимы, при этом

$$P(X = i) = e^{-\lambda}\frac{\lambda^i}{i!}, \lambda > 0, i \geq 0,$$

$$P(Y = j) = e^{-\mu}\frac{\mu^j}{j!}, \mu > 0, i \geq 0.$$

Требуется найти $P(X = i|X + Y = j)$.

Решение

Для нахождения вероятности $P(X = i|X + Y = j)$ используем формулу условной вероятности:

$$P(A|B) = \frac{P(A \cap B)}{P(B)},$$

причём $A = \{X = i\}, B = \{X + Y = j\}$.

Найдём $P(A \cap B)$. Пересечение событий означает, что что X должно быть равно i, и одновременно $X + Y$ должно быть равно j. Это возможно только если $Y = j - i$.

Таким образом, событие $\{X = i \cap X + Y = j\}$ эквивалентно событию $\{X = i \cap Y = j - i\}$. Подставим в формулу:

$$P(A \cap B) = e^{-\lambda}\frac{\lambda^i}{i!} \cdot e^{-\mu}\frac{\mu^{j-i}}{(j-i)!} = e^{-(\lambda+\mu)} \cdot \frac{\lambda^i \mu^{j-i}}{i!(j-i)!}$$

Найдём $P(B) = P(X + Y = j)$. Запишем событие $\{X + Y = j\}$ как $\{X = k \cap Y = j - k\} \quad \forall k \in 1, \ldots, j$. Так как события несовместны и независимы:

$$P(X+Y=j)=\sum_{k=0}^{j}P(X=k\cap Y=j-k)=\sum_{k=0}^{j}P(X=k)P(Y=j-k)$$

По условию:

$$P(X=k)=e^{-\lambda}\frac{\lambda^k}{k!},\ P(Y=j-k)=e^{-\mu}\frac{\mu^{j-k}}{(j-k)!},$$

тогда

$$P(X+Y=j)=e^{-(\lambda+\mu)}\sum_{k=0}^{j}\frac{\lambda^k\mu^{j-k}}{k!(j-k)!}$$

Воспользуемся биномом Ньютона:

$$(\lambda+\mu)^j=\sum_{k=0}^{j}\frac{j!}{k!(j-k)!}\lambda^k\mu^{j-k}$$

Следовательно:

$$P(X+Y=j)=e^{-(\lambda+\mu)}\frac{(\lambda+\mu)^j}{j!}.$$

Подставим полученные выражения в формулу условной вероятности:

$$P(A|B)=\frac{P(A\cap B)}{P(B)}=\frac{e^{-(\lambda+\mu)}\frac{\lambda^i\mu^{j-i}}{i!(j-i)!}}{e^{-(\lambda+\mu)}\frac{(\lambda+\mu)^j}{j!}}=\frac{j!}{i!(j-i)!}\cdot\frac{\lambda^i\mu^{j-i}}{(\lambda+\mu)^j}=$$

$$=C_j^i\cdot(\frac{\lambda}{\lambda+\mu})^i\cdot(\frac{\mu}{\lambda+\mu})^{j-i}$$

Ответ

Таким образом, искомая вероятность равна:

$$\boxed{C_j^i\cdot(\frac{\lambda}{\lambda+\mu})^i\cdot(\frac{\mu}{\lambda+\mu})^{j-i}}$$

Задача 4

Постановка задачи

Представьте, что вам нужно быстро создать некий программный продукт, в котором будут использоваться вероятностные модели. Возможно, вы будете использовать Python и модуль `SciPy`, в котором реализовано достаточно много математических алгоритмов и моделей, в том числе и вероятностно-статистических (подмодуль `stats`). Допустим, в вашей модели есть “нестандартное” распределение (например, не реализованное в `SciPy`), для которого известна плотность, и вам нужно быстро реализовать распределение, то есть определить функцию распределения, квантили и прочие методы. Тогда идея унаследоваться от класса `rv_continuous` кажется разумной: ведь достаточно переопределить метод, соответствующий плотности. Казалось бы, отличная идея, плюс применили объектно-ориентированный подход. Но допустим, что вам понадобилось сгенерировать случайные числа (на самом деле псевдослучайные) из данного распределения, но вдруг что-то пойдёт не так, например, будет слишком долго работать... И вдруг нужно будет сообразить другие методы.

Само задание:

1. Для заданной плотности p наследуйтесь от класса `rv_continuous` и сгенерируйте с помощью реализованного подкласса для вашего распределения n случайных чисел из данного распределения. Посмотрите, как будет работать алгоритм при росте n (сразу большим n не делайте). Убедитесь эмпирически, что распределение сгенерированных чисел согласуется с данным вероятностным законом.
2. Реализуйте генерацию для вашего распределения с помощью обратной к функции распределения (при написании функции для F^{-1} вызов специальных функций, например, квантили стандартного нормального закона разрешается; не забыть аналитический вывод F^{-1}). Проведите тот же эксперимент.
3. Выберите ещё один метод для генерации случайных чисел (можно, например, rejecting sampling, ratio of uniforms или другой метод). Опишите его математическое обоснование. Воспользуйтесь реализацией

или сами реализуйте. Проведите тот же эксперимент.

Условие варианта 2

$$p(x) = 3x^2 e^{-x^3} \cdot 1(x \geq 0)$$

Аналитические вычисления

Плотность распределения

$$p(x) = 3x^2 e^{-x^3} \cdot 1(x \geq 0)$$

Функция распределения

$$F(x) = \int_{-\infty}^{x} p(t)dt = \int_{0}^{x} 3t^2 e^{-t^3} dt = -e^{-t^3}\Big|_0^x = 1 - e^{-x^3}$$

То есть:

$$F(x) = \begin{cases} 0, & x < 0 \\ 1 - e^{-x^3}, & x \geq 0 \end{cases}$$

Обратная функция распределения

$$F(x) = 1 - e^{-x^3} \leftrightarrow 1 - F(x) = e^{-x^3} \leftrightarrow \ln(1 - F(x)) = -x^3 \leftrightarrow x = (-\ln(1 - F(x)))^{1/3}$$

Тогда обратная функция распределения:

$$F^{-1}(y) = (-\ln(1-y))^{1/3}, \quad y \in [0, 1)$$

Метод rejection sampling

Идея метода Генерация семплов по $f(x)$ происходит с помощью более простого вспомогательного распределения $g(x)$, которое мы можем просемплировать, и которое удовлетворяет следующему условию:

$$\forall x \; f(x) < cg(x), \quad c > 1$$

Алгоритм

1. Взять семпл x по распределению $g(x)$;
2. Выбрать случайное число u равномерно из отрезка $[0, cg(x)]$;
3. Вычислить $f(x)$;
4. Если $u \leq f(x)$, то x добавляется к семплам;
5. Если $u > f(x)$, то x отклоняется (отсюда и название метода).

Обоснование Пусть Y — случайная величина с плотностью распределения $g(y)$, а U — случайная величина, равномерно распределённая на отрезке $[0, cg(x)]$.

Мы хотим показать, что условная плотность Y при условии события $U \leq f(Y)$ равна искомой плотности $f(x)$.

$$P(Y = y \mid U \leq f(Y)) = \frac{P(Y = y \cap U \leq f(Y))}{P(U \leq f(Y))}$$

Рассмотрим знаменатель. Заметим, что U и Y независимы, так как U выбирается независимо от Y.

$$P(A) = P(U \leq f(Y)) = \int_{-\infty}^{\infty} P(U \leq f(y)) g(y) dy =$$
$$= \int_{-\infty}^{\infty} \frac{f(y)}{cg(y)} g(y) dy = \frac{1}{c} \int_{-\infty}^{\infty} f(y) dy = \frac{1}{c}$$

Рассмотрим числитель.

$$P(Y = y \cap U \leq f(Y)) = P(U \leq f(Y) | Y = y) \cdot P(Y = y) =$$
$$= P(U \leq f(y)) \cdot g(y) == \frac{f(y)}{cg(y)} g(y) = \frac{f(y)}{c}$$

Теперь объединим и получим искомый результат:

$$P(Y = y | U \leq f(Y)) = \frac{f(y)}{c} \cdot c = f(y)$$

Выбор вспомогательного распределения В качестве вспомогательного распределения выберем экспоненциальное распределение $g(x) = e^{-x}$. Для метода необходимо найти константу c такую, что $p(x) \leq c \cdot g(x)$ для всех $x \geq 0$. Это эквивалентно поиску максимума функции $h(x) = \frac{p(x)}{g(x)}$:

$$h(x) = \frac{3x^2 e^{-x^3}}{e^{-x}} = 3x^2 e^{x-x^3}$$

Найдём производную $h(x)$, чтобы найти её максимум:

$$h'(x) = 6xe^{x-x^3} + 3x^2 e^{x-x^3}(1 - 3x^2) =$$
$$= 3xe^{x-x^3}(2 + x(1 - 3x^2)) = 3xe^{x-x^3}(2 + x - 3x^3)$$

Приравняем производную к нулю, чтобы найти критические точки, $x \geq 0$:

$$3xe^{x-x^3}(2 + x - 3x^3) = 0 \leftrightarrow x = 0 \vee (x - 1)(3x^2 + 3x + 2) = 0 \leftrightarrow$$
$$\leftrightarrow x = 0 \vee x = 1$$

- При $x = 0$, $h(0) = 0$.
- При $x = 1$, $h(1) = 3(1)^2 e^{1-1} = 3$.

Поскольку $\lim_{x\to\infty} h(x) = 0$, максимум функции достигается в точке $x = 1$. Таким образом, можно выбрать $c = 3$.

Эмпирическое решение и выводы

Эксперименты, их результаты и выводы написаны блокноте 4.ipynb.