

Задание 1

И. Герасимов

1 Использование программы

```
python3.8 main.py -f FILE [-n N] [-s SEQUENCE]
```

Наличие параметра SEQUENCE определяет выбранный режим. Если его нет, то первый, если есть, то второй.

2 Обоснование интерпретации выходных данных

Моделирование источника происходит следующим образом:

1. Случайно выбираются два числа от 0 до 1. Одно определяет выбранную модель согласно распределению переключателя. Второе определяет выбранный символ согласно распределению выбранной модели.
2. Определение происходит следующим образом: программа поочередно проходит по всем моделям/символам и суммирует вероятность. Если на очередной итерации полученная сумма превысила случайно выбранное число, то берется модель/символ этой итерации.

пример. Пусть источник находится на переключателе с распределением 0.7 для монеты-1 и 0.3 для монеты-2. Было выбрано число 0.72453425. На первой итерации сумма равна 0.7 и она меньше выбранного числа. На второй итерации сумма стала больше и выбрана монета-2.

Теперь нужно определить символ по монете-2 с распределением 0.6 и 0.4. Было выбрано число 0.21353464 и на первой итерации получили большую сумму. Выбран первый символ.

Практическое (по полученной выборке длины N от источника) вычисление вероятности какого-либо сообщения длины l выполняется подсчетом количества появлений этого сообщения на общее количество сообщений в выборке длины N (то есть всего $N - l + 1$)

3 Пример стационарного источника

Описание источника можно найти в файле `station.json`. Имеем два переключателя и две монеты. В нечетный момент времени используется первый переключатель. В четный момент времени используется второй переключатель. Распишем вероятности каждого из символов в нечетный и четный моменты. Поскольку требуется построение стационарного источника, мы хотим, чтобы эти вероятности совпали (то есть нет зависимости от времени).

Пусть для первого переключателя вероятности моделей равны, а для второго - 0.7 и 0.3 соответственно. Пусть:

p - вероятность первого символа в первой модели. $1-p$ - вероятность второго символа.

q - вероятность первого символа во второй модели. $1-q$ - вероятность второго символа.

Получаем следующие уравнения для стационарного источника:

$$0.5p + 0.5q = 0.7p + 0.3q$$

$$0.5(1-p) + 0.5(1-q) = 0.7(1-p) + 0.3(1-q)$$

Решением будет $p = q$. То есть, если у нас в разные моменты времени разные переключатели, то модели должны быть одинаковыми. В файле указан пример для $p = q = 0.4$.

То есть, если мы в зависимости от времени выбираем две разных монеты, мы все равно можем получить независимость вероятностей от времени.

Отметим, что источник также является источником без памяти и, соответственно, эргодическим.

4 Пример нестационарного источника

Пусть имеется станок, который выпускает некоторую деталь. Деталь, созданная станком может оказаться дефектной. С каждой выпущенной деталью, качество станка уменьшается. Поэтому через некоторое число деталей (например после 4 итераций) его заменяют на новый.

Получаем следующий нестационарный источник, описанный в файле `nonstation.json`. Пусть символ «good» означает, что деталь в хорошем состоянии, а «bad» - деталь бракованная. Опишем 4 переключателя, при переходе по которым вероятность выпуска хорошей детали уменьшается, а бракованной увеличивается. Далее источник переходит к первому переключателю, что означает замену станка.

5 Пример неэргодического источника

Пусть мы хотим реализовать источник, который будет выдавать ответы «да», «нет», «не знаю» для дистанционного тестирования в зависимости от настроения сдающего. Настроение сдающего будет выбираться произвольно из множества «оптимист», «пессимист», «студент». Если настроение «оптимист», то предпочтение отдается ответу «да». «Пессимист» - предпочтение ответу «нет». «студент» - предпочтение ответу «не знаю».

Источник можно описывать следующим образом (файл nonergodic.json). имеется единственный переключатель, реализующий равновероятный выбор одной из трех моделей. В первой и второй моделях символы «да» и «нет» соответственно имеют большую вероятность, чем другие. В третьей всегда выбирается символ «не знаю».

Рассмотрим вероятности появления некоторого ответа:

$$Pr[\text{«да»}] = \frac{1}{3}0.8 + \frac{1}{3}0.1 = 0.9\frac{1}{3}$$

$$Pr[\text{«нет»}] = \frac{1}{3}0.1 + \frac{1}{3}0.8 = 0.9\frac{1}{3}$$

$$Pr[\text{«не знаю»}] = \frac{1}{3}0.1 + \frac{1}{3}0.1 + \frac{1}{3} = 1.2\frac{1}{3}$$

Что не совпадает с частотами по каждой реализации.

6 Предлагаемое развитие предложенного формата для описания источника с памятью

Предлагается следующее (реализуется в задании 2):

В source указывать не строку, а словарь, например:

```
{
  'switch': switch_1,
  'input': [
    {
      'code': ['0'],
      'next': 0
    }
  ]
}
```

- По ключу switch указывается имя переключателя;
- input - список возможных значений памяти. Состоит из словарей;
- По ключу code указывается предыдущее требуемое сообщение - чему должна быть равна память.
- По ключу next указывается номер элемента в списке source, который должен обрабатываться, если память равна содержимому по ключу code.

Для источников без памяти input равен пустому списку.