Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени
Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Робототехника и комплексная автоматизация»
КАФЕДРА «Системы автоматизированного проектирования (РК-6)»

ОТЧЕТ О ВЫПОЛНЕНИИ ДОМАШНЕГО ЗАДАНИЯ

по дисциплине «Аналитические модели и имитационное моделирование»

Студент:	Осипов Арсений Константинович
Группа:	РК6-84б
Тип задания:	Домашнее задание №2
Название:	Дискретные цепи Маркова
Вариант:	10

Студент _____ (подпись, дата) Осипов А. К. (Фамилия, И.О.)

Преподаватель _____ (подпись, дата) Берчун Ю. В. (Фамилия, И.О.)

Оценка: _____

Москва, 2023

Содержание

Дискретные цепи Маркова **3**

1 Цель выполнения домашнего задания 3

2 Задание 3

3 Решение 4

Определение вектора предельных вероятностей для классов существенных состояний 5

Модификация стохастической матрицы 6

Определение вектора предельных вероятностей для всех состояний . . . 7

Имитационное моделирование 10

4 Вывод 11

Дискретные цепи Маркова

1 Цель выполнения домашнего задания

Цель выполнения домашнего задания – проанализировать случайные процессы по стохастической матрице переходов

2 Задание

Для цепи Маркова, заданной стохастической матрицей переходов:

1. нарисовать граф цепи;
2. выделить классы существенных и несущественных состояний (вручную – обязательно; программным путём – дополнительное задание для желающих);
3. рассчитать предельные вероятности для классов существенных состояний;
4. не выполняя матричных операций с полной стохастической матрицей переходов, записать предельные вероятности в следующих случаях:
   - 4.1 если достоверно известно, что система начинает работу в каком-то одном из состояний (для каждого исходного состояния – свой предельный вектор);
   - 4.2 если известно, что начальное распределение вероятностей – равновероятное среди всех несущественных состояний;
5. провести имитационное моделирование системы, соответствующей рассматриваемой цепи, для этого:
   - перебираем все состояния в качестве исходных;
   - случайно разыграть переход в новое состояние, учитывая распределение вероятностей перехода;
   - совершить 100 переходов;
   - подсчитать число вхождений в каждое из состояний системы;
   - повторить эксперимент 10 раз для каждого исходного состояния;
   - построить «графики» переключений состояний цепи (для наглядности соединяем дискретные точки);

3 Решение

Дана стохастическая матрица переходов $\mathbf{P}$:

$$\mathbf{P} = \begin{pmatrix}
0.27 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.21 & 0.0 & 0.22 & 0.0 & 0.3 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.21 & 0.0 & 0.0 & 0.26 & 0.0 & 0.0 & 0.0 & 0.28 & 0.0 & 0.0 & 0.0 & 0.25 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.24 & 0.0 & 0.0 & 0.23 & 0.27 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.26 \\
0.0 & 0.0 & 0.0 & 0.3 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.32 & 0.0 & 0.0 & 0.38 & 0.0 \\
0.0 & 0.21 & 0.0 & 0.0 & 0.19 & 0.0 & 0.0 & 0.23 & 0.19 & 0.0 & 0.0 & 0.0 & 0.18 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.52 & 0.0 & 0.0 & 0.12 & 0.36 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.4 & 0.0 & 0.0 & 0.24 & 0.24 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.12 \\
0.64 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.24 & 0.0 & 0.12 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.18 & 0.19 & 0.0 & 0.22 & 0.0 & 0.0 & 0.0 & 0.2 & 0.0 & 0.0 & 0.0 & 0.21 & 0.0 & 0.0 \\
0.4 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.12 & 0.0 & 0.12 & 0.0 & 0.36 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.52 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.12 & 0.0 & 0.0 & 0.36 & 0.0 \\
0.52 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.24 & 0.0 & 0.24 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.14 & 0.0 & 0.0 & 0.16 & 0.0 & 0.0 & 0.0 & 0.1 & 0.0 & 0.0 & 0.0 & 0.16 & 0.44 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.52 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.36 & 0.0 & 0.0 & 0.12 & 0.0 \\
0.0 & 0.0 & 0.76 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.24
\end{pmatrix}$$

На рисунке 2 представлен граф переходов, соответствующий матрице $\mathbf{P}$ (состояния из серого кластера в совокупности дают класс несущественных состояний).

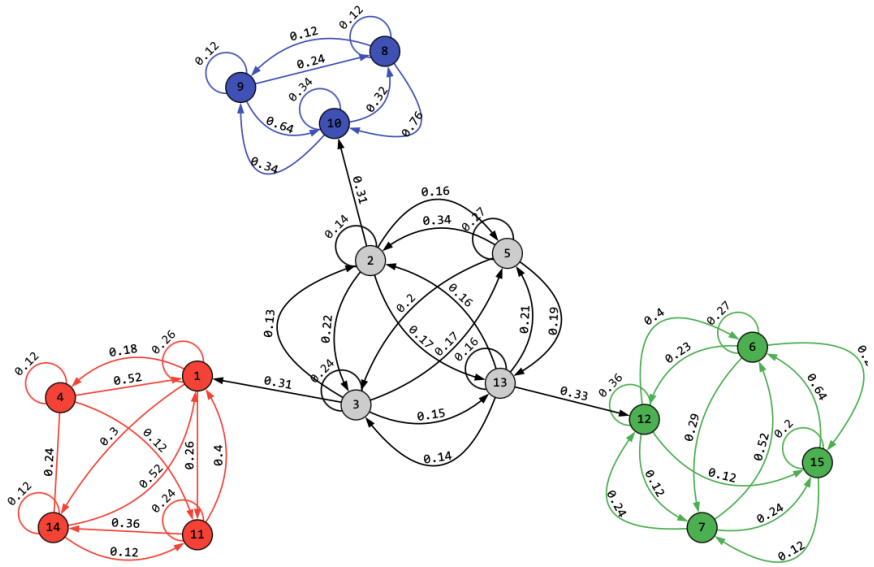

Рис. 1. Граф переходов

Определение вектора предельных вероятностей для классов существенных состояний

Рассмотрим каждый класс существенных состояний как отдельный марковский процесс.

Была разработана функция, генерирующая матрицу для кластера, заданного списком состояний.

Листинг 1. *Генерация стахастической матрицы переходов внутри кластера*

```
def make_class_matrix(class_states, P):
    n = len(class_states)
    class_matrix = np.zeros((n, n))
    k = 0

    for i in np.sort(class_states):
        row_prob = []
        for indx in range(len(P[i − 1])):
            if indx+1 in class_states:
                row_prob.append(P[i − 1][indx])
        class_matrix[k] = row_prob
        k += 1

    return class_matrix
```

Таким образом имеем 3 матрицы ($\mathbf{P_1}$, $\mathbf{P_2}$, $\mathbf{P_3}$) определяющие переходы внутри классов.

$$\mathbf{P_1} = \begin{pmatrix} 0.27 & 0.21 & 0.22 & 0.3 \\ 0.64 & 0.0 & 0.24 & 0.12 \\ 0.4 & 0.12 & 0.12 & 0.36 \\ 0.52 & 0.0 & 0.24 & 0.24 \end{pmatrix}$$

$$\mathbf{P_2} = \begin{pmatrix} 0.3 & 0.32 & 0.38 \\ 0.52 & 0.12 & 0.36 \\ 0.52 & 0.36 & 0.12 \end{pmatrix}$$

$$\mathbf{P_3} = \begin{pmatrix} 0.24 & 0.23 & 0.27 & 0.26 \\ 0.52 & 0.12 & 0.36 & 0.0 \\ 0.4 & 0.24 & 0.24 & 0.12 \\ 0.76 & 0.0 & 0.0 & 0.24 \end{pmatrix}$$

Далее по известной формуле находим вектор предельных вероятностей.

$$(\mathbf{P}^T - \mathbf{E}) \times \overline{p} = \overline{0},$$

где $\overline{p}$–вектор предельных вероятностей

Однако, поскольку система является линейно зависимой в качестве последнего уравнения внесем условие $\sum_{i=1}^{n_{vertex}} p_i = 1$.

Была реализована функция, вычисляющая вектор предельных вероятностей $\overline{p}$.

Листинг 2. *расчет вектора предельных вероятностей*

```
def marginal_probabilities(P, sum_p=1):
    A = P.T - np.eye(len(P), dtype=float)
    A[-1] = np.full(len(P), 1)
    b = np.zeros(len(P))
    b[-1] = sum_p
    p = np.linalg.solve(A, b)

    return p
```

Таким образом, векторы предельных вероятностей:

$$\overline{p}_1 = (0.4067 \quad 0.1103 \quad 0.207 \quad 0.276)^T$$

$$\overline{p}_2 = (0.4262 \quad 0.2766 \quad 0.2972)^T$$

$$\overline{p}_3 = (0.4183 \quad 0.1721 \quad 0.2301 \quad 0.1794)^T$$

Модификация стохастической матрицы

Для дальнейших вычислений, а также наглядной визуализации результатов моделирования необходимо отсортировать матрицу.

Была реализована функция изменения матрицы по списку состояний.

Листинг 3. *Модификация матрицы*

```
def change_matrix(pos, P):
    n = len(P)
    sort_matrix = np.zeros((n, n))
    for i in range(len(P)):
        sort_matrix[i]=P[pos[i]-1]

    sort_matrix = sort_matrix.T
    for i in range(len(sort_matrix)):
        P[i] = sort_matrix[i]
    for i in range(len(P)):
        sort_matrix[i]=P[pos[i]-1]
    return sort_matrix.T
```

Граф соответствующий измененной стохастической матрице (для удобства кластеры выделены тем же цветом).

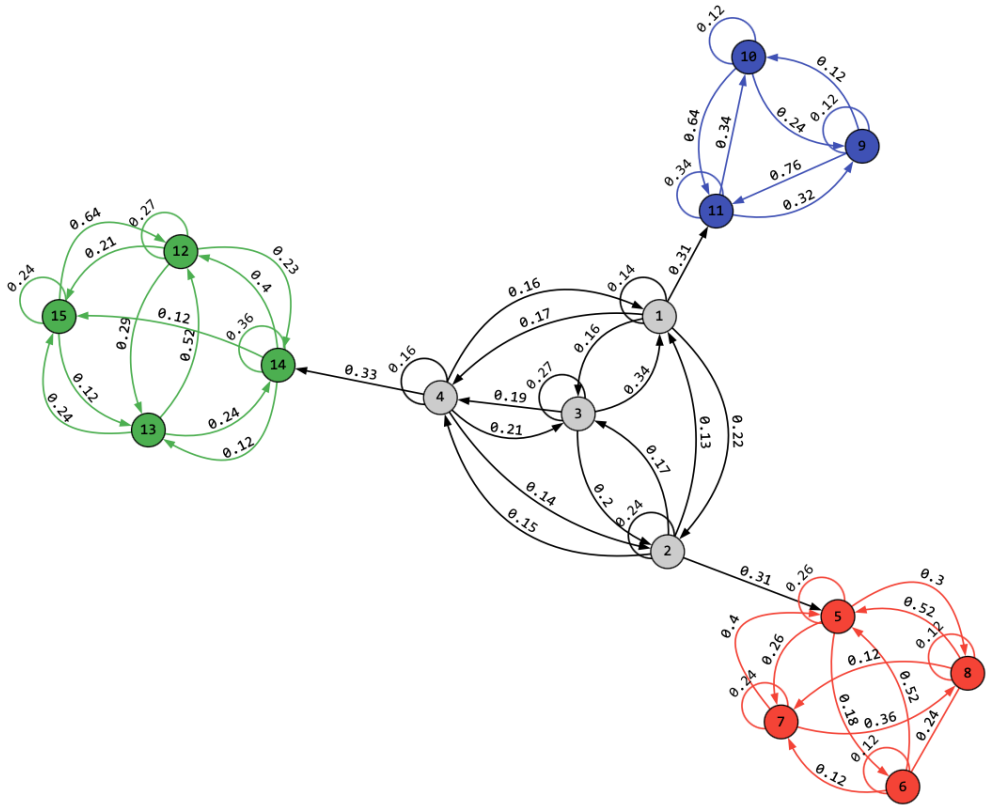

Рис. 2. Модифицированный граф

Определение вектора предельных вероятностей для всех состояний

Для эффективности вычислений выполним свертку кластеров существенных состояний, представив каждый кластер существенных состояний, как одно состояние с единичным переходом в самого себя (рисунок 3).

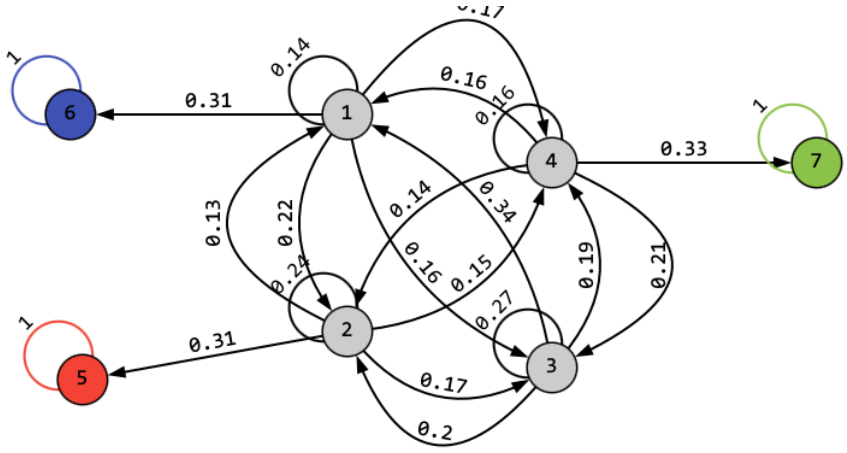

Рис. 3. Граф после свертки

Данному графу соответсвует следующая стохастическая матрица переходов:

$$\mathbf{P}(n) = \begin{pmatrix} 0.21 & 0.26 & 0.28 & 0.25 & 0.0 & 0.0 & 0.0 \\ 0.21 & 0.19 & 0.19 & 0.18 & 0.23 & 0.0 & 0.0 \\ 0.18 & 0.22 & 0.2 & 0.21 & 0.0 & 0.0 & 0.19 \\ 0.14 & 0.16 & 0.1 & 0.16 & 0.0 & 0.44 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{pmatrix}$$

На основе полученной стохастической матрицы получим предельную матрицу, которая может быть вычеслена следующим образом:

$$\mathbf{P}(n) = \mathbf{P}^n$$

В пределе вероятность оказаться в любом состоянии, пренадлежащем несущественному кластеру, равна 0. Таким образом можно сформулировать критерий остановки последовательности $\mathbf{P}(n) = \mathbf{P} \cdot \mathbf{P} \cdot \mathbf{P} \cdot \ldots$. Будем считать, что матрица сошлась к предельной, когда сумма вероятностей переходов внутри несущественного кластера будет меньше заданного ϵ.

Листинг 4. *Нахождение предельной матрицы переходов*

```
1    def P_n(P_svert, n):
2        eps = 0.0001
3        s = 1
4        while s > eps:
5            P_svert = P_svert.dot(P_svert)
6            s = 0
7            for j in range(n):
8                s += np.sum(P_svert[j][0:n])
9        return P_svert
```

В результате выполнения функции получена следующая матрица:

$$\mathbf{P}(n) = \begin{pmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.275 & 0.507 & 0.218 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.443 & 0.391 & 0.166 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.225 & 0.414 & 0.361 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.157 & 0.732 & 0.111 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{pmatrix}$$

Далее рассматриваем возможные старты только из состояний, принадлежащих несущественному кластеру (при старте из любого состояния существенного кластера вектор предельных вероятностей будет состоять из нулей и значений при рассчете вероятностей для каждого конкретного кластера).

Пусть имеется один кластер несущественных состояний и три кластера существенных состояний, тогда

$$\mathbf{P}(n)^T \cdot \pi = (0,\, 0,\, 0 \ldots x,\, y,\, z)^T,$$

где π - вектор, состоящий из всех нулей за исключением состояния, из которого осуществляется старт (там стоит 1), x, y, z - предельные вероятности оказаться в одном из существенных кластеров. Количество лидирующих нулей равно количеству состояний в несущественном кластере.

Таким образом искомые вектора могут быть найдены в виде $p = (0,\, 0,\, 0 \ldots x \cdot p_1, y \cdot p_2, z \cdot p_3)^T$.

$\overline{p_1} = (0, 0, 0, 0, 0.1119, 0.0303, 0.057, 0.076, 0.2162, 0.1403, 0.1507, 0.091, 0.0375, 0.0501, 0.0391)^T$

$\overline{p_2} = (0, 0, 0, 0, 0.1802, 0.0488, 0.0917, 0.1223, 0.1668, 0.1082, 0.1163, 0.0693, 0.0285, 0.0381, 0.0297)^T$

$\overline{p_3} = (0, 0, 0, 0, 0.0915, 0.0248, 0.0466, 0.0621, 0.1764, 0.1145, 0.123, 0.1511, 0.0622, 0.0831, 0.0648)^T$

$\overline{p_4} = (0, 0, 0, 0, 0.0639, 0.0173, 0.0325, 0.0433, 0.3121, 0.2025, 0.2176, 0.0464, 0.0191, 0.0255, 0.0199)^T$

Найдем вектор предельныех вероятностей если известно, что начальное распределение вероятностей – равновероятное среди всех несущественных состояний. Для рассматриваемого случая вектор π будет иметь вид $(1/n, 1/n ... 0, 0, ... 0)$, где количество ненулевых элементов равно n - количеству состояний в несущественном кластере, а количество нулевых позиций соотвествует количеству существенных кластеров.

$\overline{p} = (0, 0, 0, 0, 0.1119, 0.0303, 0.0569, 0.0759, 0.2179, 0.1414, 0.1519, 0.0895, 0.0368, 0.0492, 0.0384)^T$

Имитационное моделирование

Была разработана функция, реализующая переходы, заданное число раз, возвращающая траекторию и список с количеством посещений для каждого состояния (индексы – состояния).

Листинг 5. *реализация марковского процесса*

```
def mark_iter(n, m, states, s_start):
    current_s = s_start
    states_tr = [current_s]
    n_entry = [0 for _ in range(len(states))]
    for _ in range(n - 1):
        per_ver = m[current_s - 1]
        n_entry[current_s - 1] += 1
        next_s = np.random.choice(states, p=per_ver)
        current_s = next_s
        states_tr.append(current_s)
    return n_entry, states_tr
```

На рисунке 4 представлены «графики» переключений состояний цепи.

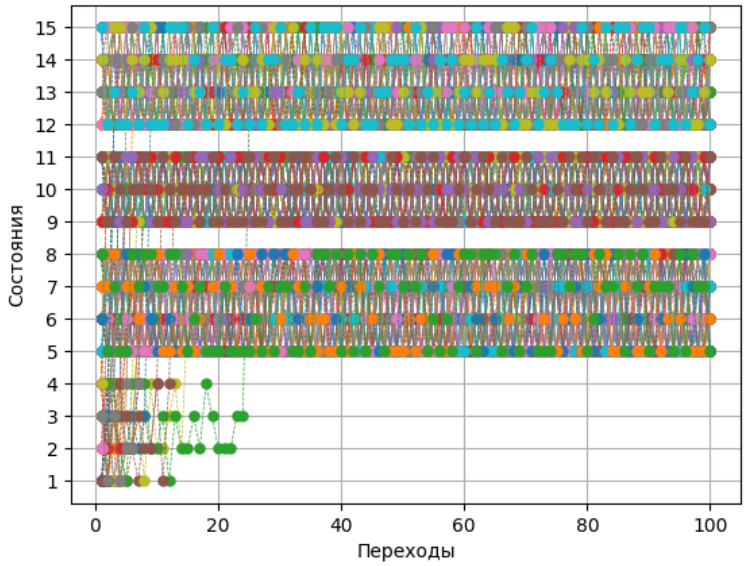

Рис. 4. Переключения состояния цепи

Статистические частоты заключительных состояний моделирования:

$$(0.0, 0.0, 0.0, 0.0, 0.16, 0.04, 0.093, 0.06, 0.187, 0.06, 0.087, 0.113, 0.073, 0.08, 0.047)$$

Листинг 6. *реализация подсчета частот*

```
1 def stats(trajectories):
2     s = ""
3     for tr in trajectories:
4         s = s + f"{ round(tr / 150, 3) } ,"
5     return s[:-2]
```

4 Вывод

В ходе выполнения домашнего задания №2 была проанализирована работа марковского процесса, заданного стохастической матрицей переходов.