

Отчёт по лабораторной работе №7

Графики

Сырцов Александр Юрьевич

Содержание

1 Цель работы	5
2 Задание	6
3 Выполнение лабораторной работы	7
3.1 Собственные значения и векторы	7
3.2 Марковские цепи	9
4 Выводы	14

List of Tables

List of Figures

3.1	Матрица A	7
3.2	Функция для нахождения собственных значений и векторов	8
3.3	Результат в действительных числах	9
3.4	Матрица переходов	10
3.5	Векторы начальных состояний	10
3.6	Векторы вероятности через 5 шагов блуждания	11
3.7	Равновесный вектор x	12
3.8	Проверка равновесности вектора x	13

1 Цель работы

Научится вычислять собственные значения и векторы и научится оперировать ими для решения смежных задач.

2 Задание

- Сделать отчёт по лабораторной работе в формате Markdown.
- В качестве ответа предоставить отчёты в 3 форматах: pdf, docx и md (в архиве, поскольку он должен содержать скриншоты, Makefile и т.д.).

3 Выполнение лабораторной работы

3.1 Собственные значения и векторы

1. Необходимо найти собственные векторы и значения матрицы A. Для этого после включения журналирования задаю матрицу (рис. -fig. 3.1).

```
octave:3> A = [1 2 -3; 2 4 0; 1 1 1]
A =
1   2   -3
2   4    0
1   1    1
```

Figure 3.1: Матрица A

2. Теперь использую встроенную функцию `eig()`, которая возвращает матрицу из собственных векторов и матрицу с собственными значениями на главной диагонали, при условии, что определён список из двух объектов. (рис. -fig. 3.2).

```

octave:5> [v lambda] = eig(A)
v =

```

-0.2400 + 0i	-0.7920 + 0i	-0.7920 - 0i
-0.9139 + 0i	0.4523 + 0.1226i	0.4523 - 0.1226i
-0.3273 + 0i	0.2322 + 0.3152i	0.2322 - 0.3152i

```

lambda =

```

Diagonal Matrix

4.5251 + 0i	0	0
0	0.7374 + 0.8844i	0
0	0	0.7374 - 0.8844i

Figure 3.2: Функция для нахождения собственных значений и векторов

3. По результатам видно, что были получены комплексные числа, поэтому для получения действительных чисел задаю симметричную матрицу вида

$$C = A^T A$$

и нахожу собственные векторы и значения этой матрицы функцией `eig()` (рис. -fig. 3.3).

```

octave:6> C = A' * A
C =

```

6	11	-2
11	21	-5
-2	-5	10

```

octave:7> [v lambda] = eig(C)
v =

```

0.876137	0.188733	-0.443581
-0.477715	0.216620	-0.851390
-0.064597	0.957839	0.279949

```

lambda =

```

Diagonal Matrix

0.1497	0	0
0	8.4751	0
0	0	28.3752

Figure 3.3: Результат в действительных числах

3.2 Марковские цепи

4. Цепь Маркова – последовательность событий, образующая дискретное распределение, где все состояния имеют зависимость. Рассматривается следующий пример:

Пусть у нас есть 5 состояний для описания перемещения по дороге. В состояни-

ях 2, 3 и 4 совершаются поворот влево или вправо, а на концах дороги происходит остановка.

Задача – предсказать, где мы окажемся через k-е количество шагов блуждания. В первую очередь я задал транспонированную матрицу переходов T (рис. -fig. 3.4).

```
octave:8> T = [1 0.5 0 0 0; 0 0 0.5 0 0; 0 0.5 0 0.5 0; 0 0 0.5 0 0; 0 0 0 0.5 1]
T =
1.0000  0.5000      0      0      0
0          0  0.5000      0      0
0  0.5000      0  0.5000      0
0          0  0.5000      0      0
0          0      0  0.5000  1.0000
```

Figure 3.4: Матрица переходов

5. После задал 4 вектора вероятностей начальных состояний: a - начальное состояние неизвестно, b - начальное состояние может быть на концах дороги, c - почти наверно мы находимся в состоянии 2, d - почти наверно находимся в состоянии 3 (рис. -fig. 3.5).

```
octave:10> a = [0.2; 0.2; 0.2; 0.2; 0.2];
octave:11> b = [0.5; 0; 0; 0; 0.5];
octave:12> c = [0; 1; 0; 0; 0];
octave:13> d = [0; 0; 1; 0; 0];
```

Figure 3.5: Векторы начальных состояний

6. Чтобы выяснить состояния через несколько шагов необходимо умножить матрицу переходов в степени, соответствующей количеству шагов, на вектор начальных состояний. В нашем случае всего 5 шагов. Выясним состояния после пяти шагов в зависимости от начальных условий. Я получил следующий результат (рис. -fig. 3.6).

```

octave:14> T^5 * a
ans =
0.450000
0.025000
0.050000
0.025000
0.450000

octave:15> T^5 * b
ans =
0.5000
0
0
0
0.5000

octave:16> T^5 * c
ans =
0.6875
0
0.1250
0
0.1875

octave:17> T^5 * d
ans =
0.3750
0.1250
0
0.1250
0.3750

```

Figure 3.6: Векторы вероятности через 5 шагов блуждания

7. Теперь нужно найти равновесное состояние, то есть состояние, которое не приведёт ни к каким изменениям системы. Вектор такого равновесного состояния представляет из себя собственный вектор сумма элементов которого равна единице, чтобы собственное значение было равно единице. Сначала задаю новую матрицу переходов T и нахожу его собственные векторы. Далее беру первый собственный вектор и записываю элементы в виде частного значения элемента и суммы всех элементов вектора, чтобы

сумма элементов вектора x (вектор равновесного состояния) была равна единице (рис. -fig. 3.7).

```
octave:18> T = [0.48 0.51 0.14; 0.29 0.04 0.52; 0.23 0.45 0.34]
T =
0.480000 0.510000 0.140000
0.290000 0.040000 0.520000
0.230000 0.450000 0.340000

octave:19> [v lambda] = eig(T)
v =
-0.6484 -0.8011 0.4325
-0.5046 0.2639 -0.8160
-0.5700 0.5372 0.3835

lambda =
Diagonal Matrix
1.0000 0 0
0 0.2181 0
0 0 -0.3581

octave:20>
octave:20> x = v(:, 1)/sum(v(:, 1))
x =
0.3763
0.2929
0.3308
```

Figure 3.7: Равновесный вектор x

8. Проверяю вектор, проводя с ним разное количество шагов (10 и 50), в обоих случаях значения одинаковы. Кроме того, вычитаю результаты от разного количества шагов, результат отличается лишь через десяток знаков после запятой вследствие погрешности алгоритмов нахождения собственных векторов, так что можно говорить о том, что разница при разном количестве шагов стремится к нулю, что говорит о том, что значения и вправду одинаковые, а найденный вектор является равновесным состоянием (рис. -fig. 3.8).

```
octave:21> T^10 * x
ans =
0.3763
0.2929
0.3308

octave:22> T^50 * x
ans =
0.3763
0.2929
0.3308

octave:23> T^50 * x - T^10 * x
ans =
2.2204e-16
1.6653e-16
1.1102e-16
```

Figure 3.8: Проверка равновесности вектора x

4 Выводы

Мне удалось научится находить собственные векторы и значения матриц, а также удалось применить их для решения задач на цепи Маркова.