Министерство науки и высшего образования
Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

Факультет *«Информатика и системы управления»*

ОТЧЕТ
по лабораторной работе №13

по дисциплине

«Информационный поиск и извлечение информации из текстов»

Студент группы ИУ9-21М _____ *С.С. Погосян*
(подпись, дата)

Руководитель _____ *Н.В. Лукашевич*
(подпись, дата)

Москва — 2021

1. Постановка задачи

Задание

- Есть три текста
  - w0, w1,w1
  - w0, w1,w2
  - w0, w2, w2
- Нужно сделать 5 итераций (проходов по трем текстам) EM-алгоритма (слайд 35)
  - Случайная инициализация матриц (ненулевые значения, нормализация по столбцам)
  - Подсчет p(t|w,d) для все текстов
  - Затем пересчет вероятностей в матрицах
  - Выдаем результат
  - Любым методом: программа или калькулятор

2. Решение

Все матрицы проинициализированы случайным образом функцией numpy randint (см. приложение)

i	$p(w,t,d)$, $dim = 3 \times 2 \times 3$	$\phi(w,t)$	$\theta(t,d)$
1	$p[0] = \begin{pmatrix} 0.7468 & 0.2559 & 0.0965 \\ 0.2531 & 0.7440 & 0.9034 \end{pmatrix}$ $p[1] = \begin{pmatrix} 0.8644 & 0.4266 & 0 \\ 0.1355 & 0.5733 & 0 \end{pmatrix}$ $p[2] = \begin{pmatrix} 0.8638 & 0.4252 & 0.1868 \\ 0.1361 & 0.5747 & 0.8131 \end{pmatrix}$	$\phi = \begin{pmatrix} 0.2843 & 0.4598 \\ 0.3339 & 0.1714 \\ 0.3817 & 0.3687 \end{pmatrix}$	$\theta = \begin{pmatrix} 0.8250 & 0.3692 & 0.1417 \\ 0.1749 & 0.6307 & 0.8582 \end{pmatrix}$
2	$p[0] = \begin{pmatrix} 0.7446 & 0.2658 & 0.0926 \\ 0.2553 & 0.7341 & 0.9073 \end{pmatrix}$ $p[1] = \begin{pmatrix} 0.9018 & 0.5327 & 0 \\ 0.0981 & 0.4672 & 0 \end{pmatrix}$ $p[2] = \begin{pmatrix} 0.8300 & 0.3774 & 0.1459 \\ 0.1699 & 0.6225 & 0.8540 \end{pmatrix}$	$\phi = \begin{pmatrix} 0.2835 & 0.4616 \\ 0.3686 & 0.1376 \\ 0.3478 & 0.4007 \end{pmatrix}$	$\theta = \begin{pmatrix} 0.8254 & 0.3919 & 0.1193 \\ 0.1745 & 0.6080 & 0.8806 \end{pmatrix}$
3	$p[0] = \begin{pmatrix} 0.7439 & 0.2836 & 0.0768 \\ 0.2560 & 0.7163 & 0.9231 \end{pmatrix}$ $p[1] = \begin{pmatrix} 0.9268 & 0.6333 & 0 \\ 0.0731 & 0.3666 & 0 \end{pmatrix}$ $p[2] = \begin{pmatrix} 0.8041 & 0.3587 & 0.1052 \\ 0.1958 & 0.6412 & 0.8947 \end{pmatrix}$	$\phi = \begin{pmatrix} 0.2808 & 0.4660 \\ 0.3967 & 0.1081 \\ 0.3224 & 0.4257 \end{pmatrix}$	$\theta = \begin{pmatrix} 0.8249 & 0.4252 & 0.0910 \\ 0.1750 & 0.5747 & 0.9089 \end{pmatrix}$
4	$p[0] = \begin{pmatrix} 0.7395 & 0.3083 & 0.0568 \\ 0.2604 & 0.6916 & 0.9431 \end{pmatrix}$ $p[1] = \begin{pmatrix} 0.9453 & 0.7307 & 0 \\ 0.0546 & 0.2692 & 0 \end{pmatrix}$ $p[2] = \begin{pmatrix} 0.7811 & 0.3591 & 0.0704 \\ 0.2188 & 0.6408 & 0.9295 \end{pmatrix}$	$\phi = \begin{pmatrix} 0.2767 & 0.4728 \\ 0.4199 & 0.0808 \\ 0.3033 & 0.4463 \end{pmatrix}$	$\theta = \begin{pmatrix} 0.8220 & 0.4660 & 0.0636 \\ 0.1779 & 0.5339 & 0.9363 \end{pmatrix}$
5	$p[0] = \begin{pmatrix} 0.7300 & 0.3381 & 0.0382 \\ 0.2699 & 0.6618 & 0.9617 \end{pmatrix}$ $p[1] = \begin{pmatrix} 0.9600 & 0.8193 & 0 \\ 0.0399 & 0.1806 & 0 \end{pmatrix}$ $p[2] = \begin{pmatrix} 0.7583 & 0.3723 & 0.0441 \\ 0.2416 & 0.6276 & 0.9558 \end{pmatrix}$	$\phi = \begin{pmatrix} 0.2724 & 0.4806 \\ 0.4381 & 0.0560 \\ 0.2893 & 0.4633 \end{pmatrix}$	$\theta = \begin{pmatrix} 0.8161 & 0.5099 & 0.0412 \\ 0.1838 & 0.4900 & 0.9587 \end{pmatrix}$

Исходный код доступен по ссылке: `https://github.com/legion15q/sem2/tree/master/num13/py`

3. Приложение

```
import numpy as np


def main():
    phi = np.random.randint(100, size=(3, 2)) / 100
    etta = np.random.randint(100, size=(2, 3)) / 100
    p = np.random.randint(100, size=(3, 2, 3)) / 100
    n_dw = np.random.randint(100, size=(3, 3)) / 100
    n_dwt = np.random.randint(100, size=(3, 3, 2)) / 100
    n_wt = np.random.randint(100, size=(3, 2)) / 100
    n_td = np.random.randint(100, size=(2, 3)) / 100
    n_t = np.random.randint(100, size=(2)) / 100
    n_d = np.random.randint(100, size=(3)) / 100

    for e_m in range(5):
        print("p:")
        for i in range(len(p)):
            for j in range(len(p[i])):
                for k in range(len(p[i][j])):
                    p[i][j][k] = phi[i][j] * etta[j][k] / (phi[i][0] * etta[0][k]
                    + phi[i][1] * etta[1][k])
                    if (i == 1) and (k == 2):
                        p[i][j][k] = 0
        print(p)

        for i in range(len(n_wt)):
            for j in range(len(n_wt[i])):
                n_wt[i][j] = 0
        for i in range(len(n_wt)):
            for j in range(len(n_wt[i])):
                for k in range(len(p[i][j])):
                    n_wt[i][j] += p[i][j][k]
        for i in range(len(n_t)):
            for j in range(len(n_wt[0])):
                n_t[i] = 0
        for i in range(len(n_wt[0])):
            for j in range(len(n_wt)):
                n_t[i] += n_wt[j][i]
        for i in range(len(phi)):
            for j in range(len(phi[i])):
                phi[i][j] = n_wt[i][j] / n_t[j]
        print('phi:')
        for i in range(len(phi)):
            str_ = ''
            for j in range(len(phi[i])):
```

```
                str_ += str(phi[i][j]) + ' '
            print(str_)
        for i in range(len(n_td)):
            for j in range(len(n_td[i])):
                n_td[i][j] = 0
        for i in range(len(n_td)):
            for j in range(len(n_td[i])):
                for k in range(len(p)):
                    n_td[i][j] += p[k][i][j]
        for i in range(len(n_d)):
            n_d[i] = 0
        for i in range(len(n_td)):
            for j in range(len(n_d)):
                n_d[j] += n_td[i][j]
        for i in range(len(etta)):
            for j in range(len(etta[i])):
                etta[i][j] = n_td[i][j] / n_d[j]

        print('etta:')
        for i in range(len(etta)):
            str_ = ''
            for j in range(len(etta[i])):
                str_ += str(etta[i][j]) + ' '
            print(str_)
        print('')
if __name__ == '__main__':
    main()
```