**Министерство образования Российской Федерации**

# МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Н.Э. БАУМАНА

Факультет: Информатика и системы управления Кафедра: Информационная безопасность (ИУ8)

# ТЕОРИЯ ИГР И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

**Лабораторная работа №6 на тему:**

«Информационное противоборство» Вариант 9

|  |  |
| --- | --- |
| Преподаватель: | Коннова Н.С. |
| Студент: | Киселев В.А. |
| Группа: | ИУ8-104 |

Москва, 2025

# Цель работы

Изучить теоретико-игровую модель информационного противоборства в социальных сетях. Промоделировать информационное управление в рамках игры и определить итоговое мнение агентов.

# Задание

1. Для 10 агентов случайным образом сгенерировать стохастическую матрицу доверия.
2. Назначить всем агентам случайное начальное мнение из заданного отрезка числового отрезка. Найти итоговое мнение агентов.
3. Случайным образом выбрать количество и номера непересекающихся агентов влияния из общего числа агентов для первого и второго игроков. Назначить им начальные мнения первого и второго игроков. Остальным агентам (нейтральным) назначить случайные начальные мнения. Смоделировать информационное управление в рамках игры и определить итоговое мнение агентов.

# Ход работы

1. ГЕНЕРАЦИЯ СТОХАСТИЧЕСКОЙ МАТРИЦЫ ДОВЕРИЯ

--------------------------------------------------

Сгенерированная матрица доверия:

[[0.197 0.117 0.015 0.252 0.101 0.007 0.126 0.079 0.106]

[0.148 0.181 0.119 0.052 0.043 0.196 0.023 0.092 0.146]

[0.078 0.038 0.041 0.069 0.17 0.153 0.079 0.176 0.197]

[0.208 0.063 0.191 0.056 0.107 0.072 0.051 0.141 0.112]

[0.088 0.182 0.144 0.027 0.061 0.043 0.13 0.164 0.16 ]

[0.092 0.192 0.1 0.095 0.116 0.09 0.048 0.166 0.101]

[0.002 0.147 0.159 0.109 0.152 0.049 0.075 0.164 0.142]

[0.166 0.056 0.109 0.048 0.059 0.116 0.182 0.152 0.111]

[0.159 0.042 0.129 0.104 0.093 0.107 0.079 0.14 0.147]]

Суммы строк матрицы (должны быть равны 1):

[1. 1. 1. 1. 1. 1. 1. 1. 1.]

2. МОДЕЛИРОВАНИЕ БЕЗ УПРАВЛЕНИЯ

--------------------------------------------------

Начальные мнения агентов: [ 1.116 2.534 8.272 9.723 13.311 5.995 5.347 10.331 4.813]

Процесс итераций:

Итерация 0: [ 1.116 2.534 8.272 9.723 13.311 5.995 5.347 10.331 4.813]

Итерация 1: [6.479 5.641 7.559 6.63 6.255 6.886 7.846 6.269 6.825]

Итерация 2: [6.608 6.596 6.665 6.725 6.689 6.517 6.648 6.848 6.751]

Итерация 3: [6.684 6.648 6.691 6.679 6.686 6.68 6.699 6.673 6.683]

Итерация 4: [6.68 6.677 6.681 6.682 6.678 6.676 6.679 6.683 6.682]

Итерация 5: [6.68 6.679 6.68 6.68 6.68 6.68 6.68 6.68 6.68 ]

...

Сходимость достигнута на итерации 10

Итоговые мнения агентов: [6.68 6.68 6.68 6.68 6.68 6.68 6.68 6.68 6.68]

3. МОДЕЛИРОВАНИЕ С ИНФОРМАЦИОННЫМ УПРАВЛЕНИЕМ

--------------------------------------------------

Агенты первого игрока: [8 2]

Агенты второго игрока: [0 1]

Нейтральные агенты: [5 7 6 3 4]

Начальные мнения с учетом управления:

Агент 0 (игрок 2): -32.344

Агент 1 (игрок 2): -83.608

Агент 2 (игрок 1): 34.396

Агент 3 (нейтральный): 13.698

Агент 4 (нейтральный): 12.985

Агент 5 (нейтральный): 6.120

Агент 6 (нейтральный): 15.554

Агент 7 (нейтральный): 14.839

Агент 8 (игрок 1): 42.155

Процесс итераций:

Итерация 0: [-32.344 -83.608 34.396 13.698 12.985 6.12 15.554 14.839 42.155]

Итерация 1: [-3.221 -5.467 11.992 4.783 -0.439 -4.735 6.49 5.672 8.629]

Итерация 2: [2.216 1.193 2.779 3.256 3.504 1.952 3.964 3.133 3.335]

Итерация 3: [2.907 2.444 2.971 2.78 2.764 2.65 2.892 2.899 2.877]

Итерация 4: [2.8 2.763 2.812 2.844 2.801 2.768 2.794 2.837 2.835]

Итерация 5: [2.813 2.799 2.81 2.809 2.806 2.805 2.81 2.806 2.811]

...

Сходимость достигнута на итерации 12

Итоговые мнения агентов: [2.808 2.808 2.808 2.808 2.808 2.808 2.808 2.808 2.808]

ВЫВОД: Выиграл первый игрок (итоговое мнение положительное)

4. АНАЛИЗ МАТРИЦЫ ВЛИЯНИЯ

--------------------------------------------------

Предельная матрица влияния (после сходимости):

[[0.132 0.108 0.108 0.094 0.098 0.093 0.093 0.14 0.135]

[0.132 0.108 0.108 0.094 0.098 0.093 0.093 0.14 0.135]

[0.132 0.108 0.108 0.094 0.098 0.093 0.093 0.14 0.135]

[0.132 0.108 0.108 0.094 0.098 0.093 0.093 0.14 0.135]

[0.132 0.108 0.108 0.094 0.098 0.093 0.093 0.14 0.135]

[0.132 0.108 0.108 0.094 0.098 0.093 0.093 0.14 0.135]

[0.132 0.108 0.108 0.094 0.098 0.093 0.093 0.14 0.135]

[0.132 0.108 0.108 0.094 0.098 0.093 0.093 0.14 0.135]

[0.132 0.108 0.108 0.094 0.098 0.093 0.093 0.14 0.135]]

Влияние агентов (стационарное распределение):

Агент 0: 0.1317

Агент 1: 0.1076

Агент 2: 0.1079

Агент 3: 0.0939

Агент 4: 0.0982

Агент 5: 0.0931

Агент 6: 0.0928

Агент 7: 0.1396

Агент 8: 0.1353

Рейтинг агентов по влиянию:

1. Агент 7 (влияние: 0.1396)

2. Агент 8 (влияние: 0.1353)

3. Агент 0 (влияние: 0.1317)

4. Агент 2 (влияние: 0.1079)

5. Агент 1 (влияние: 0.1076)

6. Агент 4 (влияние: 0.0982)

7. Агент 3 (влияние: 0.0939)

8. Агент 5 (влияние: 0.0931)

9. Агент 6 (влияние: 0.0928)

# Выводы

В ходе выполнения лабораторной работы была изучена теоретико- игровая модель информационного противоборства в социальных сетях, а также проведено моделирование процесса формирования мнений агентов под влиянием информационного управления.

На основе случайно сгенерированной стохастической матрицы доверия (отражающей взаимное влияние агентов) и вектора начальных мнений рассчитаны итоговые мнения участников системы.

Смоделирован сценарий информационной конкуренции между двумя игроками: им случайным образом были назначены агенты, которые никак не пересекались между собой, также были назначены мнения. Если один игрок имеет больше агентов влияния или более высокий уровень доверия, его позиция доминирует в конечном состоянии системы.

import numpy as np

def simulate\_opinion\_dynamics(n\_agents=9, epsilon=1e-6, max\_iter=1000):

    """

    Моделирование динамики мнений в социальной сети с подробным выводом информации

    """

    print("=" \* 60)

    print("ЛАБОРАТОРНАЯ РАБОТА №8: МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННОГО ПРОТИВОБОРСТВА")

    print("=" \* 60)

    # 1. Генерация стохастической матрицы доверия

    print("\n1. ГЕНЕРАЦИЯ СТОХАСТИЧЕСКОЙ МАТРИЦЫ ДОВЕРИЯ")

    print("-" \* 50)

    np.random.seed(42)  # Для воспроизводимости результатов

    A = np.random.rand(n\_agents, n\_agents)

    A = A / A.sum(axis=1, keepdims=True)

    np.set\_printoptions(precision=3, suppress=True)

    print("Сгенерированная матрица доверия:")

    for i in range(n\_agents):

        print(f"Строка {i}: {A[i]}")

    row\_sums = A.sum(axis=1)

    print("\nСуммы строк матрицы (должны быть равны 1):")

    for i, sum\_val in enumerate(row\_sums):

        print(f"Строка {i}: {sum\_val:.6f}")

    # 2. Моделирование без управления

    print("\n\n2. МОДЕЛИРОВАНИЕ БЕЗ УПРАВЛЕНИЯ")

    print("-" \* 50)

    x0 = np.random.uniform(1, 20, n\_agents)

    print("Начальные мнения агентов:")

    for i, opinion in enumerate(x0):

        print(f"Агент {i}: {opinion:.3f}")

    # Итерационный процесс

    x = x0.copy()

    history = [x.copy()]

    print("\nПроцесс итераций (первые 5 итераций):")

    print(f"Итерация 0: {x}")

    for i in range(max\_iter):

        x\_new = A @ x

        history.append(x\_new.copy())

        if i < 5:

            print(f"Итерация {i+1}: {x\_new}")

        if np.linalg.norm(x\_new - x) < epsilon:

            print(f"...\nСходимость достигнута на итерации {i+1}")

            break

        x = x\_new

    print("\nИтоговые мнения агентов:")

    for i, opinion in enumerate(x):

        print(f"Агент {i}: {opinion:.3f}")

    # 3. Моделирование с информационным управлением

    print("\n\n3. МОДЕЛИРОВАНИЕ С ИНФОРМАЦИОННЫМ УПРАВЛЕНИЕМ")

    print("-" \* 50)

    # Выбор агентов влияния

    agents = np.arange(n\_agents)

    np.random.shuffle(agents)

    player1\_agents = agents[:2]

    player2\_agents = agents[2:4]

    neutral\_agents = agents[4:]

    print("Агенты первого игрока:", player1\_agents)

    print("Агенты второго игрока:", player2\_agents)

    print("Нейтральные агенты:", neutral\_agents)

    # Формирование начальных мнений

    x0\_controlled = np.random.uniform(1, 20, n\_agents)

    x0\_controlled[player1\_agents] = np.random.uniform(0, 100, len(player1\_agents))

    x0\_controlled[player2\_agents] = np.random.uniform(-100, 0, len(player2\_agents))

    print("\nНачальные мнения с учетом управления:")

    for i, opinion in enumerate(x0\_controlled):

        agent\_type = ""

        if i in player1\_agents:

            agent\_type = " (игрок 1)"

        elif i in player2\_agents:

            agent\_type = " (игрок 2)"

        else:

            agent\_type = " (нейтральный)"

        print(f"Агент {i}{agent\_type}: {opinion:.3f}")

    # Итерационный процесс

    x = x0\_controlled.copy()

    history\_controlled = [x.copy()]

    print("\nПроцесс итераций (первые 5 итераций):")

    print(f"Итерация 0: {x}")

    for i in range(max\_iter):

        x\_new = A @ x

        history\_controlled.append(x\_new.copy())

        if i < 5:

            print(f"Итерация {i+1}: {x\_new}")

        if np.linalg.norm(x\_new - x) < epsilon:

            print(f"...\nСходимость достигнута на итерации {i+1}")

            break

        x = x\_new

    print("\nИтоговые мнения агентов:")

    for i, opinion in enumerate(x):

        print(f"Агент {i}: {opinion:.3f}")

    # Определение победителя

    final\_opinion = x[0]

    print(f"\nИТОГОВОЕ МНЕНИЕ: {final\_opinion:.3f}")

    if final\_opinion > 0:

        print("ВЫВОД: Выиграл первый игрок (итоговое мнение положительное)")

    else:

        print("ВЫВОД: Выиграл второй игрок (итоговое мнение отрицательное)")

    # 4. Анализ матрицы влияния

    print("\n\n4. АНАЛИЗ МАТРИЦЫ ВЛИЯНИЯ")

    print("-" \* 50)

    # Вычисление стационарного распределения

    eigenvalues, eigenvectors = np.linalg.eig(A.T)

    stationary\_idx = np.argmin(np.abs(eigenvalues - 1.0))

    stationary\_distribution = np.real(eigenvectors[:, stationary\_idx])

    stationary\_distribution = np.abs(stationary\_distribution)  # Берем модуль

    stationary\_distribution /= stationary\_distribution.sum()  # Нормализуем

    # Предельная матрица влияния

    A\_infinity = np.tile(stationary\_distribution, (n\_agents, 1))

    print("Предельная матрица влияния (первые 3 строки):")

    for i in range(min(3, n\_agents)):

        print(f"Строка {i}: {A\_infinity[i]}")

    print("\nВлияние агентов (стационарное распределение):")

    for i, score in enumerate(stationary\_distribution):

        print(f"Агент {i}: {score:.4f}")

    # Рейтинг агентов по влиянию

    most\_influential = np.argsort(stationary\_distribution)[::-1]

    print("\nРейтинг агентов по влиянию:")

    for rank, agent in enumerate(most\_influential):

        print(f"{rank+1}. Агент {agent} (влияние: {stationary\_distribution[agent]:.4f})")

# Запуск моделирования

if \_\_name\_\_ == "\_\_main\_\_":

    simulate\_opinion\_dynamics()