TG

HW1

Определить функцию полезности обучения в 4-ом семестре.

Сперва зададим множество альтернатив X. Для этого введем следующие 4 параметра, которые описывают качество обучения:

- m - результирующее количество баллов $\in [0,1]$, где 0 - возможный минимум, 1 - возможный максимум;
- k - качество полученных знаний $\in [0,1]$ - оценка по ощущению;
- t - затраченное время $\in [0,1]$, где 0 - возможный минимум, 1 - возможный максимум;
- h - качество ментального здоровья на выходе $\in [0,1]$ - оценка по ощущению;

Тогда пусть $X = [0,1]^4$.

Теперь введем отношение предпочтительности на X. Для этого зададим линейную форму $\omega = (2, 3, -1, 1)$ (что-то вроде набора весов для каждого из параметров), тогда $\forall\ A, B \in X:\ A \prec B \ \Leftrightarrow\ \omega A < \omega B,\ \ A = B \ \Leftrightarrow\ \omega A = \omega B$. На ΔX эти отношения индуцируются через соответствие $\alpha A + (1-\alpha)B \ \to\ \alpha\omega A + (1-\alpha)\omega B$. Очевидно, что отношения $(\prec)$ и $(=)$ корректно определены и удовлетворяют четырем аксиомам.

По теореме фон Неймана найдется функция $U : X \to \mathbb{R}$, сохраняющая порядок. Нетрудно ее определить. Пусть $\forall\ A \in X:\ U(A) = \omega A$. Для лотерей: $\forall\ C = (\alpha A + (1-\alpha)B) \in \Delta X:\ U(C) = \alpha U(A) + (1-\alpha)U(B)$.

HW2

Выбрать любую игру/ситуацию из жизни и сформулировать её в виде биматричной игры в нормальной форме.

Рассмотрим ситуацию: муж (1-ый игрок) и жена (2-ой игрок) вечером воскресенья решают когда они хотят лечь спать - рано (E) или поздно (L). Причем, если они лягут вместе - это +1 к ауре для каждого, если кто-то ляжет рано - +1 ему к ауре за здоровый сон, если ляжет поздно - удовлетворит потребность в сериале и получит +1 к ауре. В этой семье муж - фрилансер и ему не надо вставать в подельник рано утром на работу, а жена работает в офисе, поэтому если она ляжет рано и встанет выспавшейся в рабочий день, то получит +1 к ауре.

Построим матрицы ситуации:

$$A_1 = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix},\ \ A_2 = \begin{pmatrix} 3 & 1 \\ 2 & 2 \end{pmatrix}$$

Стратегия мужа: $\sigma_1 = (p, 1-p)$; стратегия жены: $\sigma_2 = (q, 1-q)$. Найдем устойчивое по Нэшу состояние используя лемму о равенстве платежей:

$$K_1(E, \sigma_2) = \begin{pmatrix} 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} q \\ 1-q \end{pmatrix} = 1 + q$$

$$K_1(L, \sigma_2) = \begin{pmatrix} 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} q \\ 1-q \end{pmatrix} = 2 - q$$

$$q = 0,5$$

$$K_2(\sigma_1, E) = \begin{pmatrix} p & 1-p \end{pmatrix} \cdot \begin{pmatrix} 3 & 1 \\ 2 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 2 + p$$

$$K_2(\sigma_1, L) = \begin{pmatrix} p & 1-p \end{pmatrix} \cdot \begin{pmatrix} 3 & 1 \\ 2 & 2 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 2 - p$$

$$p = 0$$

Значит устойчивые стратегии мужа и жены соответственно равны: $\sigma_1 = (0,\ 1), \sigma_2 = (0,5,\ 0,5)$

HW3

Решить составленную Вами игру методами линейного программирования. Решить составленную Вами игру итеративным матричным алгоритмом (книгра Петросян, Зинкевич, страница 52). Проанализировать результаты и подготовить отчет.

Как уже выяснилось, в биматричной игре мы имеем следующие матрицы для мужа и жены соответственно:

$$A_1 = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix},\ \ A_2 = \begin{pmatrix} 3 & 1 \\ 2 & 2 \end{pmatrix}$$

Теперь найдем цены игры для мужа и жены методом **линейного программирования**. Благо нам не понадобится помощь вычислительной машины в этом вопросе, все отлично решается руками (и очевидно, что машина получит такой же ответ). Сначала решим для мужа. Пусть $x = (x_1, x_2)^T$, тогда

$$\begin{cases} x_i \geqslant 0, \\ x^T A_1 \geqslant (1,1)^T, \\ x_1 + x_2 \to min \end{cases}$$

$$\begin{cases} 2x_1 + x_2 \geqslant 1, \\ x_1 + 2x_2 \geqslant 1, \end{cases}$$

$$x_1 + x_2 \geqslant \frac{2}{3}$$

Мы хотим минимизировать эту сумму, тогда $d_x = min(x_1 + x_2) = \frac{2}{3}$, а отсюда получаем, что цена игры составит $\frac{1}{d_x} = 1,5$, что мы как раз-таки получаем при $q = 0,5$ (см. HW2).

Аналогичным образом найдем и для жены. Пусть $y = (y_1, y_2)^T$, тогда

$$\begin{cases} y_i \geqslant 0, \\ Ay \leqslant (1,1)^T, \\ y_1 + y_2 \to max \end{cases}$$

$$\begin{cases} 3y_1 + y_2 \leqslant 1, \\ 2y_1 + 2y_2 \leqslant 1 \end{cases}$$

Здесь тоже все достаточно прозрачно: $d_y = max(y_1 + y_2) = \frac{1}{2}$, то есть цена игры составит $\frac{1}{d_y} = 2$, что мы получаем при $p = 0$.

Теперь перейдем к **итеративному методу**. Реализуем подход, когда на каждой итерации вычисляется самая выгодная стратегия для обоих игроков. Убедимся, что следующая программа на python выдаст нам ожидаемые значения:

```
# python
def iterate_algo(A):
    n = len(A)
    m = len(A[0])

    x_strategy = [0] * n
    y_strategy = [0] * m

    v_high_values = []
    v_low_values = []

    its = 1000

    for _ in (range(its)):
        mx = - 10 ** 9
        index_x = 0
        for i in range(n):
            v_high = sum([A[i][j] * y_strategy[j] for j in range(m)])
            if v_high >= mx:
                index_x = i
                mx = v_high
        mn = 10 ** 9
        index_y = 0
        for j in range(m):
            v_low = sum([A[i][j] * x_strategy[i] for i in range(n)])
            if v_low <= mn:
                index_y = j
                mn = v_low
        x_strategy[index_x] += 1
        y_strategy[index_y] += 1
        v_high_values.append(mx)
        v_low_values.append(mn)

    print("Первый игрок:", [t / its for t in x_strategy])
    print("Второй игрок:", [t / its for t in y_strategy])

    print(
        max([v_low_values[k] / k for k in range(1, its)]),
        " <= v <= ",
        min([v_high_values[k] / k for k in range(1, its)])
    )

A1 = [[2, 1], [1, 2]]
A2 = [[3, 1], [2, 2]]

iterate_algo(A1)
iterate_algo(A2)
```

Это код выведет следующее:

```
Первый игрок: [0.517, 0.483]
Второй игрок: [0.495, 0.505]
1.5  <= v <=  1.5
Первый игрок: [0.0, 1.0]
Второй игрок: [0.0, 1.0]
2.0  <= v <=  2.0
```

Что же мы видим? Мы снова получили те же самые устойчивые стратегии из предыдущего домашнего задания и те же значения цен игры.

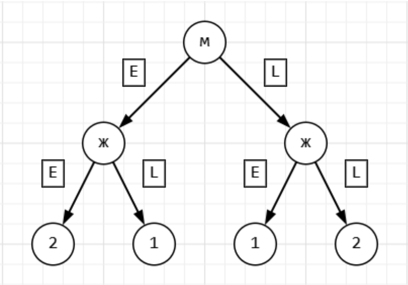
Вывод: Линейное программирование и итеративный метод позволяют решать задачи поиска равновесия по Нэшу с достаточно высокой точностью. Они могут быть весьма полезны при достаточно больших объемах входных данных, когда неудобно вычислять ответ вручную.

HW4

Построить дерево игры, предложенной Вами и проанализировать выигрышное поведение на первом ходу. Если для Вашей игры построение дерева невозможно, взять любую игры (крестики-нолики и тд) и с помощью компьютера построить дерево и проанализировать первые ходы.

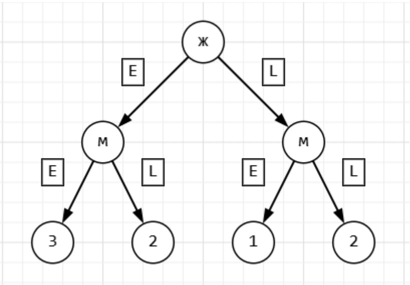
Немного переосмыслим игру, т.е. будем считать, что муж и жена делают свой выбор по очереди, в листьях будем писать выигрыш первого сделавшего выбор.

Для случая когда муж выбирает первым:



Заметим, что отсюда сразу видно, что у мужа нет более предпочтительных ходов, а подобный результат мы уже получали ранее: $(0,5\ 0,5)$.

Для случая когда жена выбирает первой:



Здесь сразу видно, что наиболее выгодной стратегией для жены является E. Это отличается от равновесной по Нэшу стратегии, которую мы получили ранее, потому что здесь мы ищем более выгодную ситуацию для конкретного человека, а не ситуацию, где никто не захочет менять свой выбор.