layout

February 27, 2025

1 The Layout Problem

[1]:
```
import numpy as np
import random
from utils import *
from human_ai import MultiHumanAI
from model.mallows import Mallows
```

1.1 Known Ground-Truth

Suppose there are k types of humans, each with **heterogeneous** ground-truth rankings. Denote the ground-truth ranking of human i as π_h^i.

If the algorithm has full knowledge of $\{\pi_h^i\}$, it can ensure that each type of human benefits from the collaboration by adopting the following strategy: always presenting a fixed set of items to the humans. In particular, it presents the first k items as the k top items of humans' ground-truth rankings.

In the following experiment,

- we consider `num_of_humans` types of humans, each of whom has a random ground-truth ranking.
- These ground-truth rankings are **known** to the algorithm.
- It takes the strategy by setting the k top items as its first k items of its ground-truth.

We can see **every human is beneficial from the collaboration**.

[2]:
```
m = 10
phi = 1
for num_of_humans in range(2, 6):
    D_hs = []
    for _ in range(num_of_humans):
        pi_h_star = list(range(1, m + 1))
        random.shuffle(pi_h_star)
        D_hs.append(Mallows(m, phi, pi_h_star))

    joint_system = MultiHumanAI(m, num_of_humans, D_hs, None)
    joint_system.find_layout()
    benefits = joint_system.benefit_of_human_single_best(num_of_humans)
```

```
    print(benefits)
```

```
[0.3671939362260944, 0.34884193704905997]
[0.21185074163951323, 0.36485074163951325, 0.3228507416395132]
[0.07885074163951322, 0.03185074163951329, 0.21185074163951323,
0.1828507416395132]
[0.1698507416395133, 0.35285074163951324, 0.25285074163951327,
0.05485074163951331, 0.20085074163951322]
```

1.2 Unknown Ground-Truth

However, the ground-truth rankings may not always known in advance to the algorith, especially in scenarios that protect user privacy.

To learn about humans' preference, algorithm usually adopt query-based learning to learn humans' preference. We suppose the humans are interacting with the algorithm in the following way:

- At time t, a human comes and a type-i human comes with a probability of p_i.
- The algorithm presents a set of items S_t to that human. She selects her favourite one from the items (but she sometimes would make mistakes). The human will get a **postive** review if the item is perfect to her and a **negative** review otherwise.
- The algorithm updates S_t by always picking the items that human like the most

[3]:
```
m = 10
phi = 1
for num_of_humans in range(2, 6):
    info(f"Number of humans {num_of_humans}")
    D_hs = []

    ## The probability of every type person arriving.
    ps = np.array([random.random() for _ in range(num_of_humans)])
    ps /= np.sum(ps)

    ## Generating ground-truth
    for _ in range(num_of_humans):
        pi_h_star = list(range(1, m + 1))
        random.shuffle(pi_h_star)
        D_hs.append(Mallows(m, phi, pi_h_star))

    ## 1000 interactions between the algorithm and these humans
    joint_system = MultiHumanAI(m, num_of_humans, D_hs, ps)
    joint_system.interaction(1000, 200)
```

```
[INFO] Number of humans 2
[INFO] t: 0, benefits: [-0.6321492583604867, -0.6321492583604867]
[INFO] t: 200, benefits: [0.3593302714709977, 0.2120025456139497]
[INFO] t: 400, benefits: [0.3593302714709977, 0.2120025456139497]
[INFO] t: 600, benefits: [0.3593302714709977, 0.2120025456139497]
[INFO] t: 800, benefits: [0.3593302714709977, 0.2120025456139497]
[INFO] Number of humans 3
```

```
[INFO] t: 0, benefits: [-0.6321492583604867, -0.6321492583604867,
0.07185074163951322]
[INFO] t: 200, benefits: [-0.6321492583604867, 0.36685074163951326,
0.3108507416395132]
[INFO] t: 400, benefits: [0.2908507416395133, 0.3328507416395132,
0.3218507416395132]
[INFO] t: 600, benefits: [0.2798507416395133, 0.3348507416395132,
0.3218507416395132]
[INFO] t: 800, benefits: [0.2908507416395133, 0.3278507416395132,
0.3258507416395132]
[INFO] Number of humans 4
[INFO] t: 0, benefits: [-0.6321492583604867, 0.1628507416395133,
0.2938507416395133, -0.6321492583604867]
[INFO] t: 200, benefits: [0.36085074163951325, 0.2758507416395133,
-0.6321492583604867, -0.6321492583604867]
[INFO] t: 400, benefits: [0.2628507416395133, 0.3158507416395132,
0.1518507416395133, -0.6321492583604867]
[INFO] t: 600, benefits: [0.2848507416395133, 0.3068507416395132,
0.08385074163951323, 0.1928507416395132]
[INFO] t: 800, benefits: [0.2748507416395133, 0.3078507416395132,
0.09785074163951324, 0.1878507416395132]
[INFO] Number of humans 5
[INFO] t: 0, benefits: [-0.6321492583604867, 0.24585074163951326,
0.09085074163951323, 0.1758507416395133, -0.6321492583604867]
[INFO] t: 200, benefits: [0.3218507416395132, -0.6321492583604867,
0.09285074163951323, 0.3158507416395132, 0.05385074163951331]
[INFO] t: 400, benefits: [0.3148507416395132, -0.6321492583604867,
0.08485074163951323, 0.3248507416395132, 0.07485074163951322]
[INFO] t: 600, benefits: [0.3058507416395132, -0.6321492583604867,
0.3288507416395132, 0.09285074163951323, 0.2838507416395133]
[INFO] t: 800, benefits: [0.1628507416395133, -0.6321492583604867,
0.33885074163951323, 0.3108507416395132, 0.2958507416395133]
```