

EconCS Final Writeup Draft

Model

The model is a game *between agents*, who represent music listeners. In later sections, we explore expanding the game to simulate *agents *subject* to a recommendation algorithm*.

Round based

- m players as Music Listeners
- N number of songs, $\{s_1, \dots, s_N\}$
- Each player has a *true mean value* per song given as follows:

$$\mu_{i,j} = \mathbf{v}_i \cdot \mathbf{w}_j$$

- \mathbf{v}_i represents the user's preferences over the 3 latent dimensions of song qualities
- \mathbf{w}_j represents the three latent qualities of song j
- $\mathbf{v}_i, \mathbf{w}_j$ are sampled from a mixture of three gaussians with same σ but different centers.
- Each Agent's Utility per listen of song s_j at round t is sampled from a normal distribution given by $u_{i,t}(s_j) \sim \mathcal{N}(\mu_{i,j}, \sigma_i^2)$
 - $\sigma_i^2 = 1$ for all agents unless otherwise specified. We explore agent behavior with various σ_i values later.
- Undirected Edges between Agents represent Agents in Community, "following one another".
 - While an edge exists from $a_1 \rightarrow a_2$, a_1 can see the sampled utilities of a_2 and their estimated mean values per song. (see their preferences + experiences), and vice-versa
 - Ideally, this would be directed, but my implementation is too inefficient. Undirected lets me use a single adjacency matrix and compute w_{ij} with vectors.

Actions at Each Round

1. Each Agent chooses to listen to one of $\{s_1, \dots, s_N\}$ songs, receiving utility $u_{i,t}(s_j)$.
2. Agent chooses to form One edge or None, given options by Model Environment (Not Recommender)
3. Agent chooses to remove any number of edges currently present

These three are the **only** actions available to Agents. Everything else is either automatically done by the Model Environment or Recommender.

Automatically Occurring at Each Round:

After Agents choose a song to listen to,

1. Pearson Correlation Coefficient computed between all agents
2. Generate list of edge candidates (Part of Model Environment, not Agent or Recommender Choice!) via the following:
 1. If two agents listen to the same song, each gets added as a candidate to form an edge for this round only

Agent 0 (Baseline): Epsilon-Greedy Agent

- Choose song with highest empirical mean utility, $\epsilon = 0.1$ chance to sample uniformly from all songs
- Don't choose to form or delete any edges. (Edges don't affect behavior)

Agent 1: "Optimal Agent" based on Collaborative Filtering Bandits

Based on: (Akamura, JMLR 2014

<https://proceedings.mlr.press/v39/nakamura14.pdf>, Li et al. SIGIR '16

<https://arxiv.org/pdf/1502.03473.pdf>

My best attempt at creating an Optimal, Reward maximizing (therefore rational) agent.

- Each agent treats each song as a Bandit in an K-Arm Bandit problem
- Use a Upper Confidence Bound (UCB) Approach with Collaborative Filtering for Estimating Mean Values per song.

Song Choice: Estimated Mean of Song informed by Neighbors

- Confidence bound for Agent i considering song k :

$$UCT_{i,k}^{\text{social}}(t) = \underbrace{\frac{N_{i,k}\hat{\mu}_{i,k} + M_{i,k}\hat{p}_{i,k}}{N_{i,k} + M_{i,k}}}_{\text{Weighted Mean Estimate}} + c\sqrt{\underbrace{\frac{\ln t}{N_{i,k} + M_{i,k}}}_{\text{Exploration Bonus}}}$$

$$M_{i,k} = \sum_{j \in \text{Neighbors}} |w_{ij}| \cdot \mathbb{I}(N_{j,k} > 0)$$

•

$$\hat{p}_{i,k} = \bar{\mu}_i + \frac{\sum_j w_{ij}(\hat{\mu}_{j,k} - \bar{\mu}_j)}{\sum_j |w_{ij}|}$$

$$\bar{\mu}_i = \frac{\text{All rewards for song } s}{\text{num. total listens}}$$

$$w_{ij} = \frac{\sum_{k \in S_{ij}} (\hat{\mu}_{i,k} - \bar{\mu}_i)(\hat{\mu}_{j,k} - \bar{\mu}_j)}{\sqrt{\sum_{k \in S_{ij}} (\hat{\mu}_{i,k} - \bar{\mu}_i)^2} \sqrt{\sum_{k \in S_{ij}} (\hat{\mu}_{j,k} - \bar{\mu}_j)^2}}$$

* $w_{ij} \in [1, -1]$ decreases as my mean and other's mean diverges. 1 = perfect match, 0 = no correlation, -1 = perfect opposite.

- $N_{i,k}$ = number of times agent i listened to song k .
- S_{ij} = set of songs both Agent i and Agent j have listened to at least once.
- **Agent** chooses song with highest UCB value.
- For *Computational efficiency*, w_{ij} is calculated between all agents automatically

Edge Creation Choice:

- Choose $\text{argmax}_j w_{ij}$ and $w_{ij} > K$, for some pre-set threshold K .

Edge Removal Choice:

- If $w_{ij} < K$ for more than $T = 3$ (preset number) of rounds consecutively, remove the edge.

This implements strategic choices for decisions modeling the following

1. Choosing by giving a value for "following" somebody else -- music exploration, vs. greedy choice of my favorite song
2. When and which connections to form or break

Agent-Only Results

Fixed Test Parameters:

- 120 Rounds
- 10 Songs

- v_i and w_j both sampled from mixture of three gaussians with center at $(-1, -1, -1), (0, 0, 0), (1, 1, 1)$ and $\sigma = 1$ for each.
- Fixed Seed=42

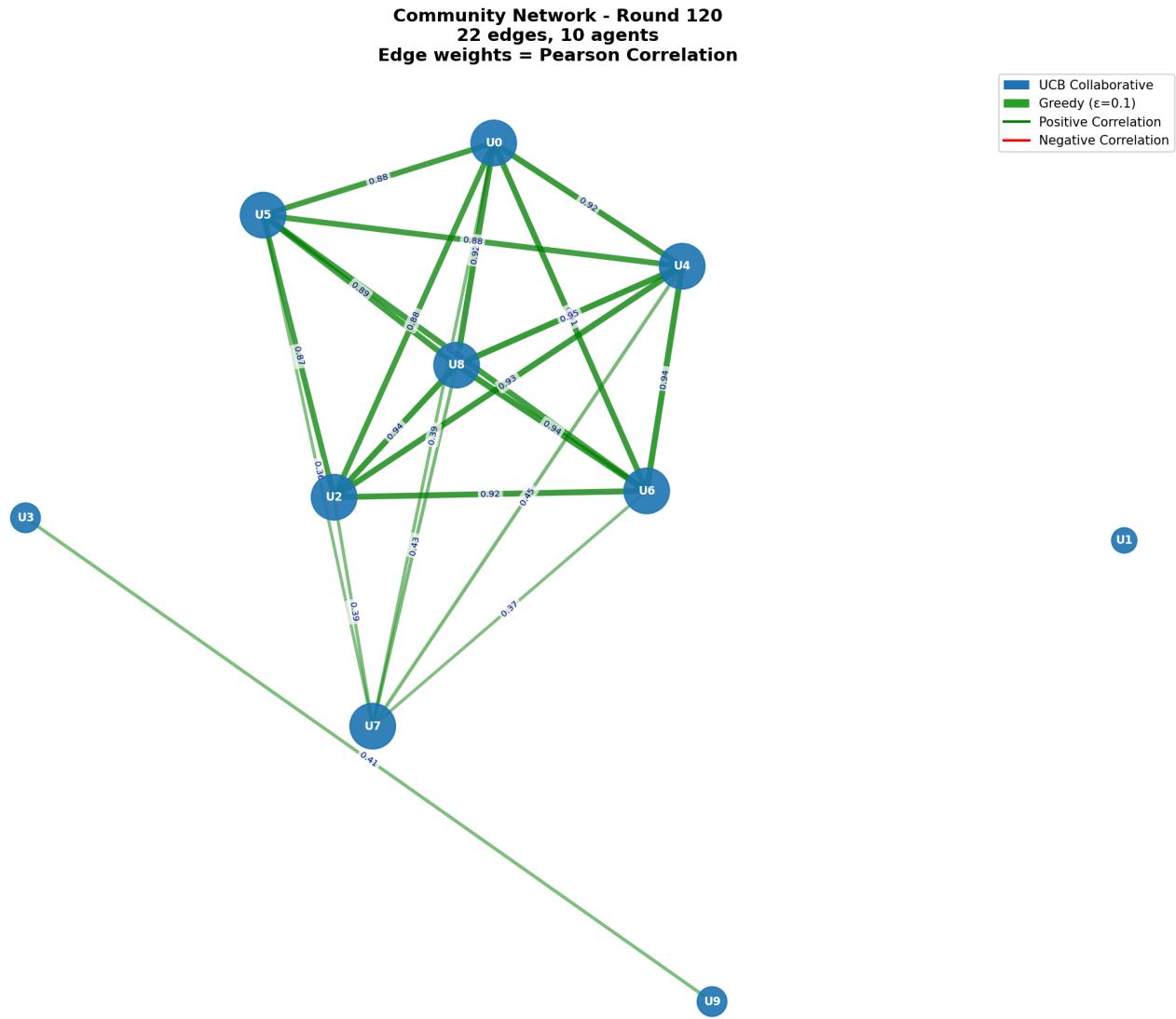
10 Collaborative Filtering UCB Agents

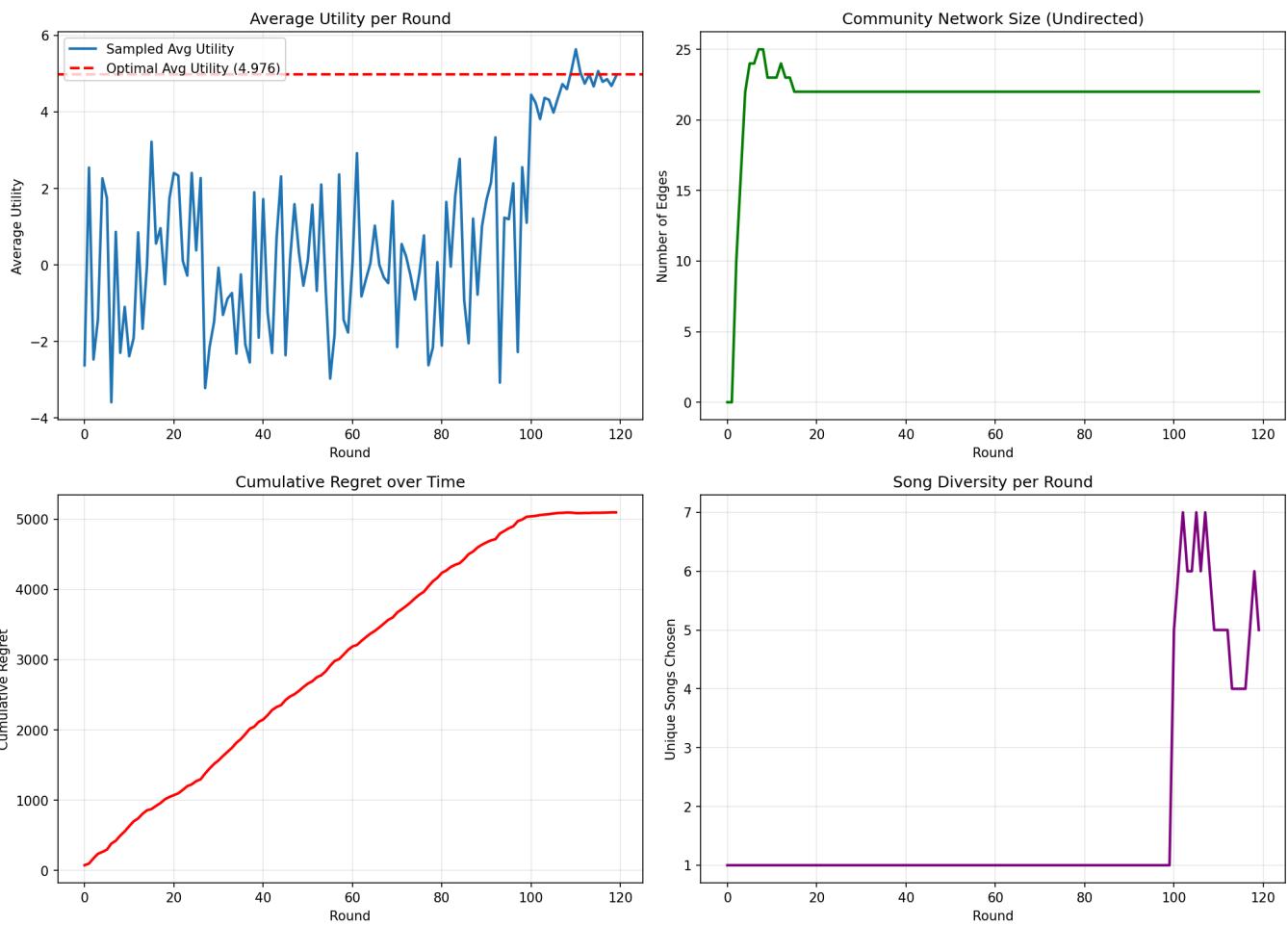
Hyper parameters:

$$c = 0.3$$

$$K = 0.3$$

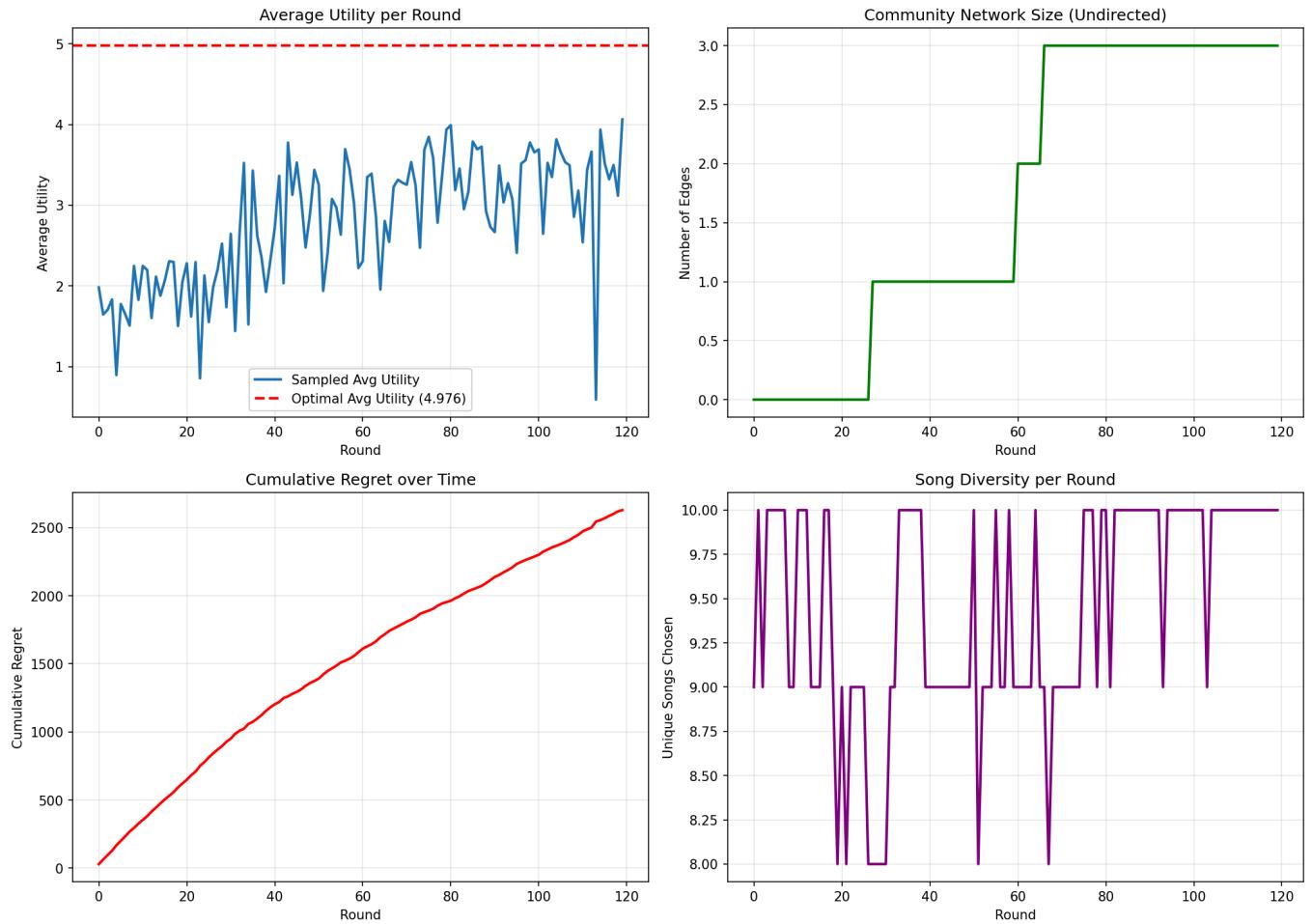
Final Network:





10 Epsilon Greedy Agents

$$\epsilon = 0.1$$

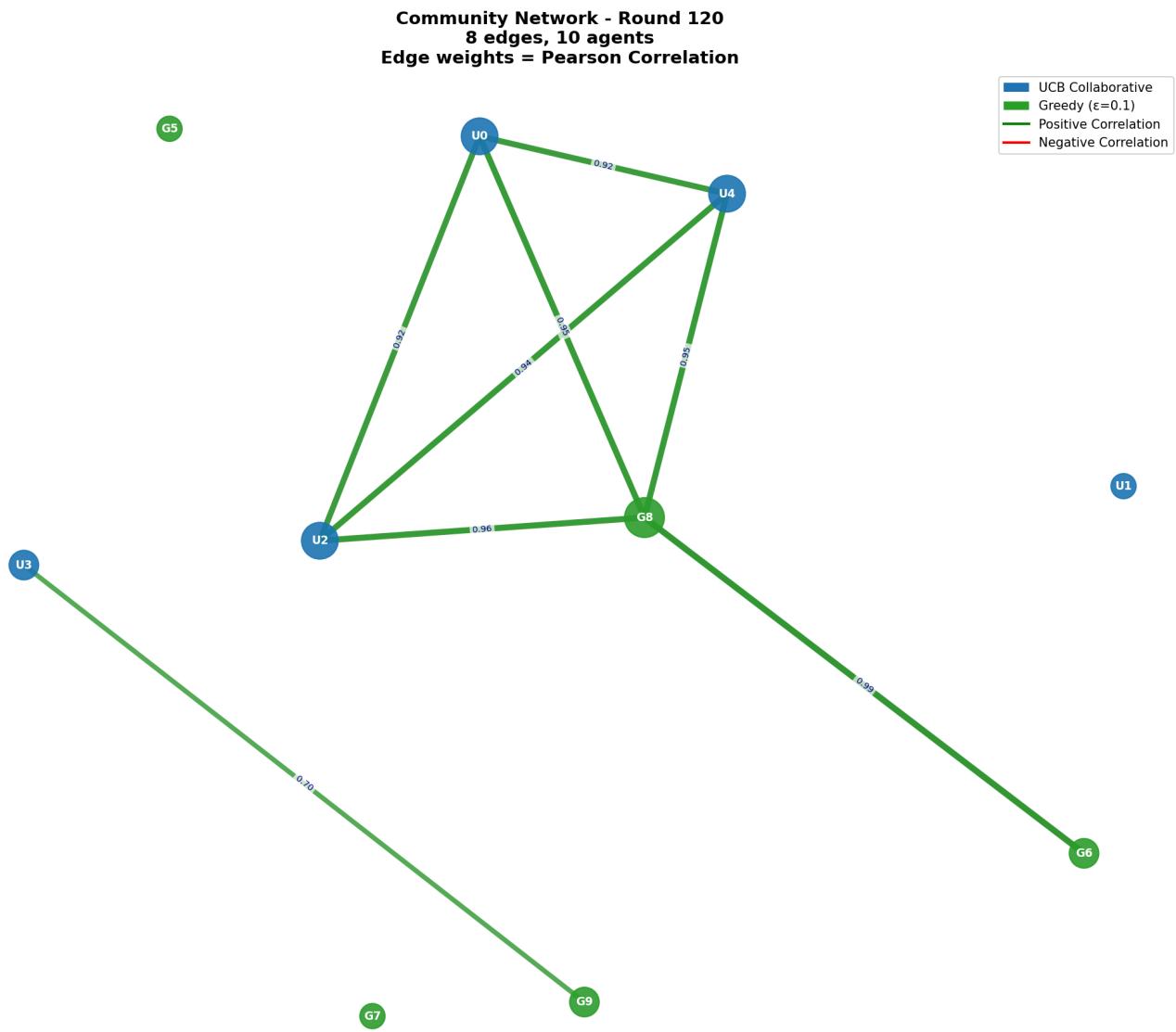


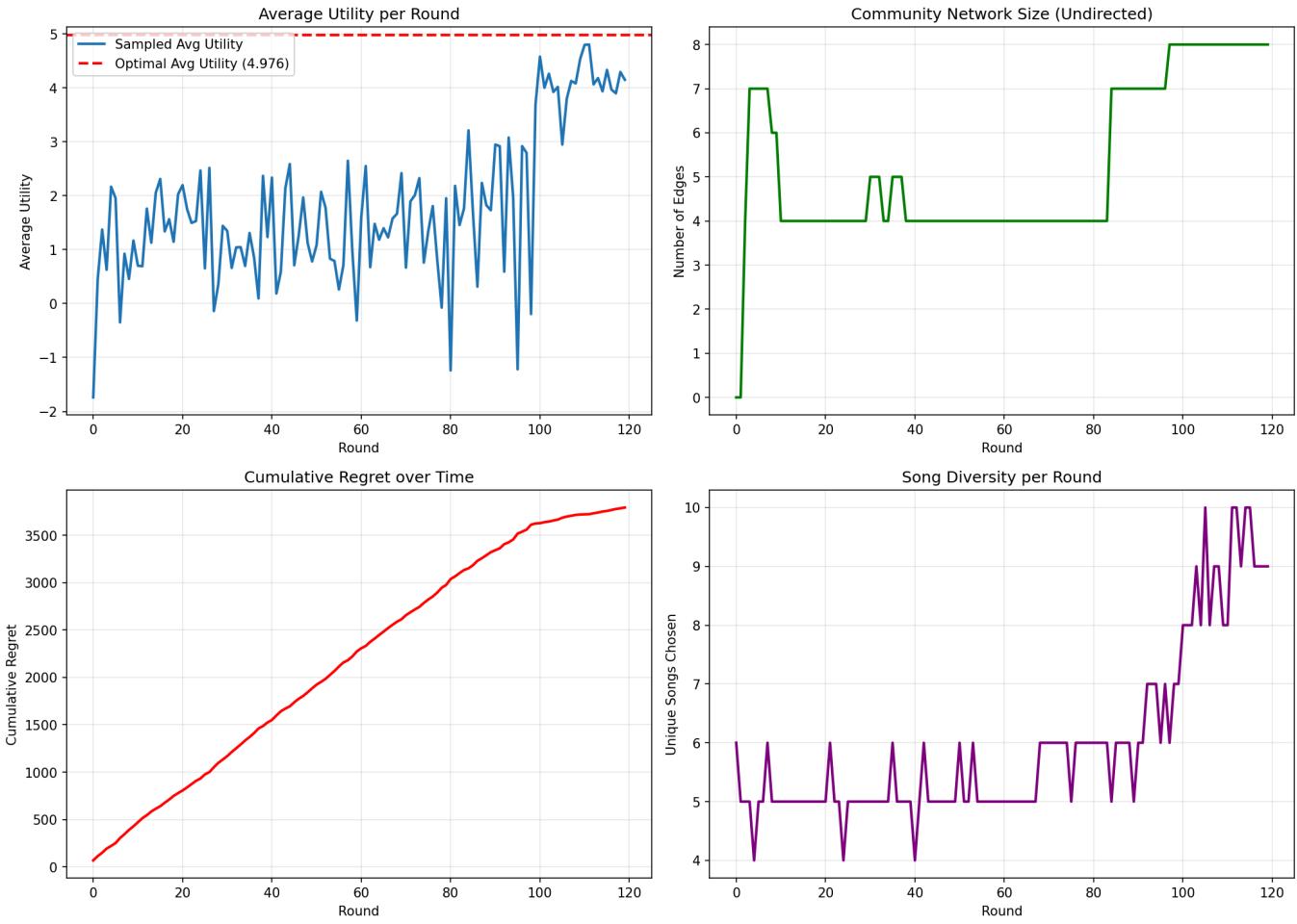
Ignore Community Network size, it doesn't influence agent behavior.

Greater Average Utility, Lower Cum. Regret Show UCB Agent is superior to baseline of Greedy agents.

5 Eps-Greedy + 5 UCB

Same hyper parameters





So Greedy Agents can benefit the mean value estimations of the UCB agents.

Recommenders

I tried to model the affect of a recommendation algorithm on this game, looking for changes in average utility, song choice, and edge forming over each round. Specifically, if recommenders change convergence.

Available Actions:

- At the start of each round, the recommender can suggest one song and report a mean utility value
 - Agents will use the reported value to choose select between the recommendation and other songs.

E.g. If recommender reports song, utility $(s_k, \hat{\mu}_{i,k})$ UCB Agent i will append $s_k, \hat{\mu}_{i,k}$ to the list of songs to choose from, and select the max from that list.

For each agent, *if it chooses the recommendation*, the agent *cannot* form an edge this round.

Recommender 1: "Optimal" Recommender

This represents an "optimal" music recommender in the sense of classical recommender design (Celma, 2010).

1. Knows true mean values for each agent for each song
 2. Recommends "new" songs to agents
- It does this by scoring each song for each agent:

$$r_i(s_k) = \frac{N_{i,k}}{\text{Round Number}} \cdot \mu_{i,k}$$

Then recommending the song with highest score, reporting the true $\mu_{i,k}$ value.

Recommender 2: "Profit-aware" Recommender

This represents a simple "profit" incentivized music recommender. It is the same as the Optimal Recommender but with an additional profit term p_k in the score computation.

- $\{p_1, \dots, p_N\}$ sampled uniformly, i.i.d from $[0, 1]$.

$$r_{i,profit}(s_k) = r_i(s_k) \cdot p_k$$

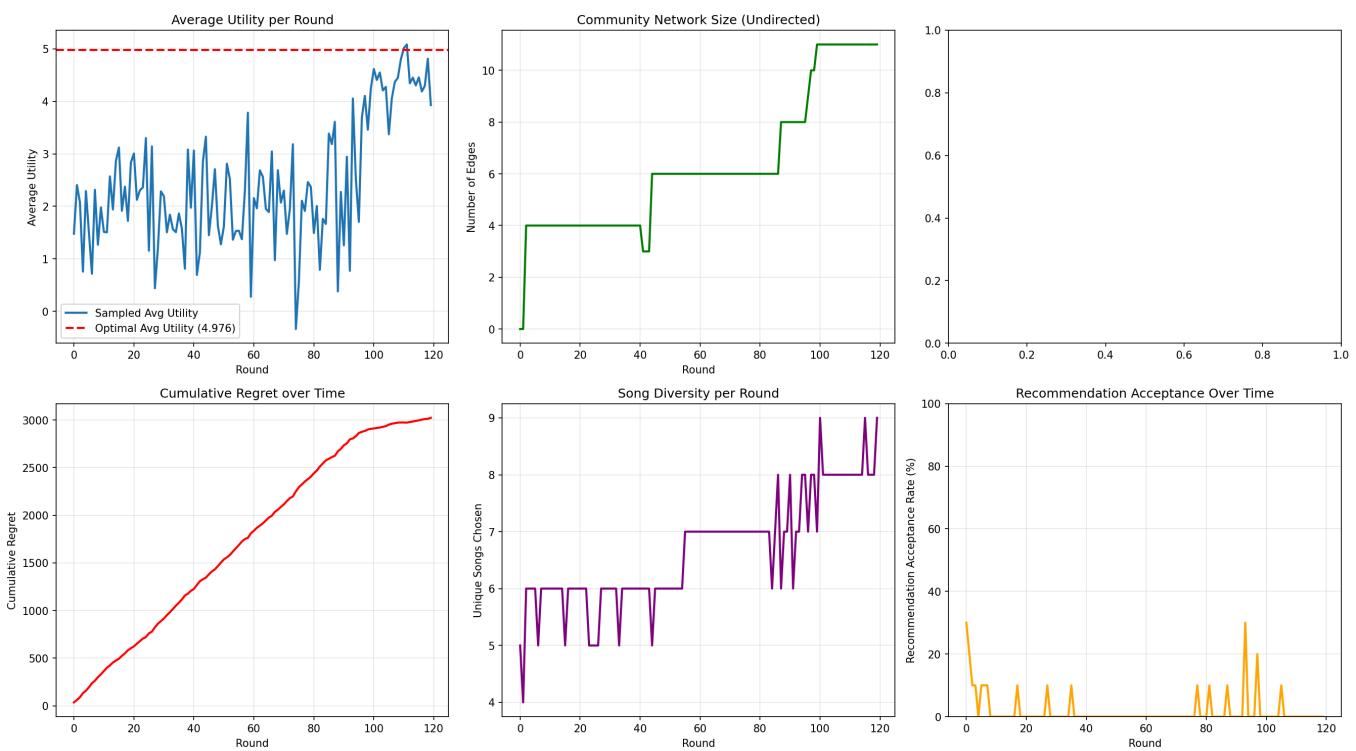
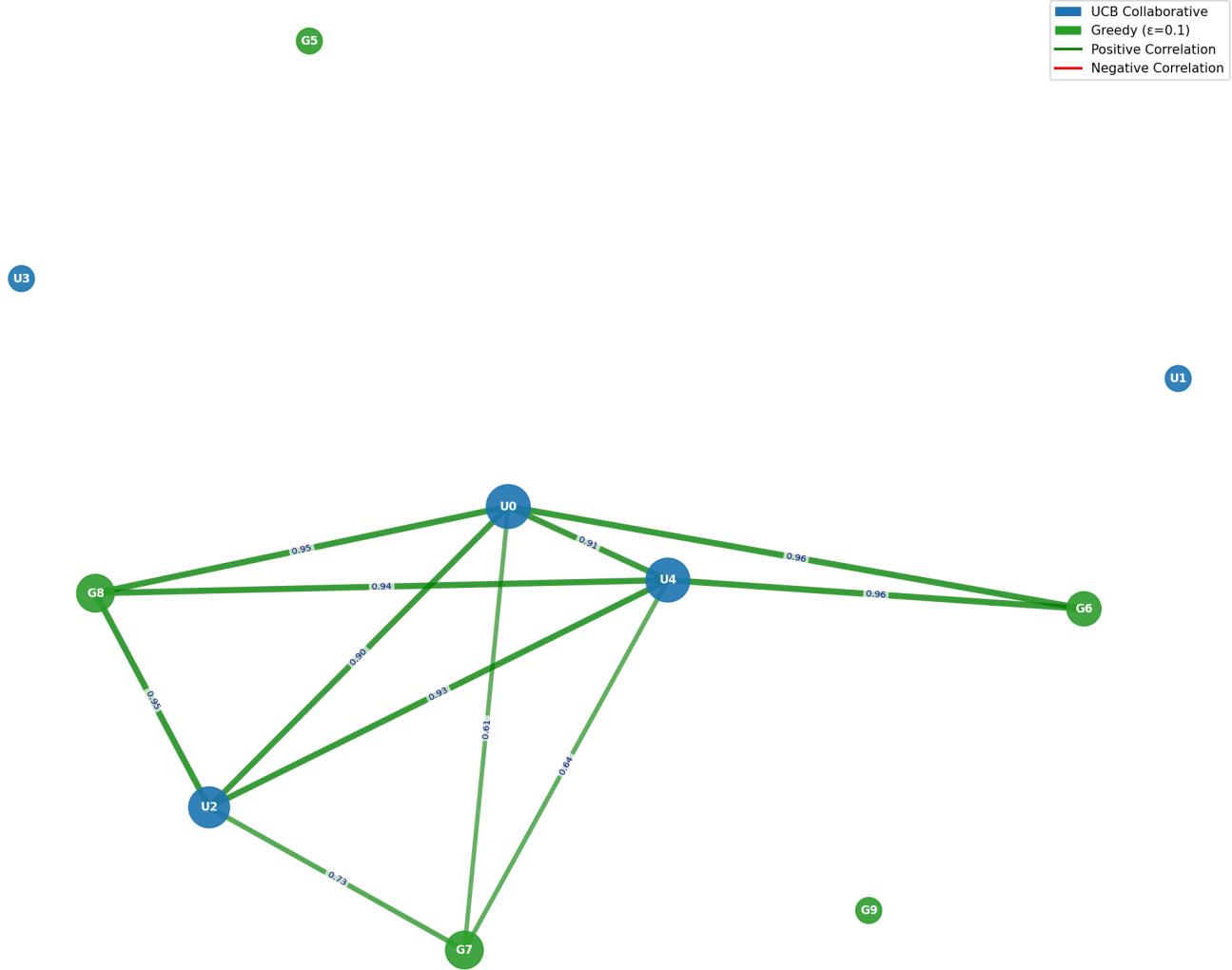
Then, it recommends the song with highest score, reporting the true $\mu_{i,k}$ value still.

Results: 5 Greedy + 5 UCB

Same hyperparameters as before

Recommender 1: "Optimal"

Community Network - Round 120
11 edges, 10 agents
Edge weights = Pearson Correlation



Recommender 2: "Profit"

Community Network - Round 120
9 edges, 10 agents
Edge weights = Pearson Correlation

