Personalized Movie Genre Recommendation via Markov Chains

Stochastic Processes (فرآیند های تصادفی)

Instructor: Dr. Razieh Khodsiani

Student: Radmehr Talebi NO: 40006893

Motivation and Overview

While casually reviewing my recent movie-watching history, I noticed an interesting pattern in the genres I was choosing. I would often move from one genre to another in a way that wasn't entirely random; for example, after watching a comedy, I'd frequently switch to a light drama or avoid horror movies late at night. Sometimes I'd revisit the same genre a few times, but then feel bored and switch things up. This observation led me to wonder if my viewing behavior could be modeled as a probabilistic system.

That's when I realized that **Markov chains**; which I was learning about in my stochastic processes course; might be a perfect fit for capturing these genre transitions. They naturally describe systems where the next state (in this case, the next genre) depends only on the current one, which seemed to match how I was making choices. This sparked the idea for this project: to build a simple, yet expressive **Markov chain-based model** that could simulate and analyze how users (starting with myself) transition between movie genres over time.

In this project, we model user behavior as a Markov chain over a set of movie genres. A personalized transition matrix is created based on the user's interests, along with fatigue effects that reduce the likelihood of repeatedly watching the same genres. To reflect realistic behavior, we also include a teleportation mechanism that occasionally redirects the user toward their favorite genres, similar to how PageRank introduces random jumps.

We further analyze the structure and dynamics of the Markov chain by computing the stationary distribution to understand long-term viewing tendencies. We examine whether the chain satisfies the reversibility condition through detailed balance checks. Using simulation, we estimate expected hitting times and cover times to measure how quickly different genres are reached. Additionally, we explore absorbing states to model scenarios where users eventually settle into fixed genre preferences.

Project Goals

- Simulate user behavior across genres as a Markov process.
- Personalize the model using user-defined genre interest vectors.
- Introduce realistic behavioral factors like **genre fatigue**, **random exploration**, and **teleportation**.
- Implement **absorbing states** to model scenarios like stopping behavior.
- Compute and interpret key theoretical properties of Markov chains:
  - **Stationary distribution**
  - **Reversibility (detailed balance)**
  - **Hitting and cover times**
  - **Absorbing state behavior**

Theoretical Foundations

Let S= $\{s_1, s_2, s_3, \ldots, s_n\}$ be the set of movie genres. Transitions between genres are defined via a **stochastic matrix** $P \in R^{n*n}$ where each entry P_{ij} represents the probability of transitioning from genre i to genre j.

The chain is enhanced with:

- **Teleportation** (probability of jumping based on user preference vector u, like in PageRank),
- **Exploration noise** (adds small probability to all transitions),
- **Genre fatigue** (decaying preference for repeatedly watched genres),
- **Absorbing states** (e.g., "Quit watching", where the user gets stuck permanently).

We analyze:

- **Stationary distribution** π, where $P\pi = \pi$
- **Reversibility**, tested via the **detailed balance equation**: $\pi_i P_{ij} = \pi_j P_{ji}$
- **Expected hitting time** from one genre to another, using Monte Carlo simulations

- **Actual hitting times** from individual simulations
- Absorption behavior once the user reaches an absorbing state

Key Features Implemented

- Custom **user profile input** and **normalization**
- Transition matrix generation using a combination of base weights and user behavior
- Simulation of **Markov walks** with fatigue and teleportation
- Analytical evaluation:
    - **Reversibility check**
    - **Stationary distribution computation**
    - **Average and actual hitting times**
    - **Detection of absorption into states**

Explanation of Functions

normalize_preference_vector (vector)

This function ensures the input preference vector is a valid probability distribution by dividing each element by the total sum. If the vector contains only zeros, it raises an error. This step is critical because user preferences must sum to 1 to be used as probabilities in a Markov chain.

---

apply_duplicity_factor (P, watch_history, genres, fatigue_decay = 0.6, min_prob = 0.05)

This function modifies the base transition matrix P to account for **genre fatigue**. It penalizes transition probabilities to genres that have been watched more frequently in the current history using an exponential decay formula. This simulates user boredom or preference shifts. The result is re-normalized to maintain row-stochastic properties.

---

personalize_matrix (P_base, user_vec, watch_history, alpha, exploration_prob)

This function creates a personalized transition matrix by adjusting the base matrix P_base to reflect both the user's preferences and fatigue. It applies a duplicity factor to reduce the probability of repeating recently watched genres (watch_history), then blends this with the user's preference vector using the teleportation factor alpha. To encourage exploration, a small probability exploration_prob is used to mix in uniform randomness across all genres. The final matrix is row-normalized to ensure it remains a valid stochastic matrix.

---

compute_stationary (P, tol=1e-10, max_iter = 10000)

This computes the **stationary distribution** of a Markov chain using iterative multiplication until convergence. The stationary distribution represents long-run probabilities of being in each genre and is used to evaluate reversibility and ergodic behavior.

---

check_reversibility (P, pi, tol=1e-6)

This checks the **detailed balance condition**, a mathematical criterion for reversibility:

$$\pi_i P_{ij} = \pi_j \, P_{ji}$$

for all i, j If this holds (within tolerance), the chain is **reversible**, which implies time-reversible dynamics and a certain symmetry in transitions.

---

reversibility_report (P, genres, user_profile_name)

This function prints and returns whether the Markov chain corresponding to a user is reversible. It uses the stationary distribution and detailed balance check, and reports this in human-readable form.

---

check_absorbing_states (P, genres)

This function looks for **absorbing states** — genres where once entered, the chain cannot leave (i.e., rows in P with a 1 on the diagonal and 0 elsewhere). This models situations like a user quitting the system (e.g., watching a "Quit watching" genre).

---

predict_next_genre (P, current_index)

Given the current genre, this function uses the personalized matrix to randomly choose the next genre based on transition probabilities. It prints the current genre, probabilities, and predicted next genre — useful for simulating realistic user transitions.

---

simulate_predictions (P, start_index, steps=5)

Starting from the initial genre index, this function predicts the next steps genres by repeatedly calling predict_next_genre. It simulates a user's likely next preferences and returns a list of predicted genres for the upcoming steps.

---

hitting_time_all (P, start_state, max_steps=1000, num_simulations=50)

For each target genre, this function estimates the expected number of steps (hitting time) it takes to reach that genre starting from start_state. It runs multiple random walk simulations limited by max_steps and averages the results to provide hitting times to all genres.

Additional Notes:

- user_profiles and user_watch_histories define behavioral input and memory.
- alpha controls how strongly a user's preferences are included in the transition matrix (like teleportation in PageRank).
- exploration_prob injects randomness into the model to simulate unpredictable choices.
- In the check_reversibility and **compute_stationary** functions, the parameter tol=1e-6 represents a **numerical tolerance** level used to account for small floating-point inaccuracies during calculations. Because floating-point arithmetic in computers is not perfectly precise, direct equality checks between real numbers often fail due to tiny rounding errors. Instead of requiring exact equality for the detailed balance condition

  $$\pi_i P_{ij} = \pi_j \, P_{ji}$$

  the function checks whether the difference between the two sides is smaller than the tolerance (set here as 10^{-6}) If this condition is met for all pairs (i, j) the Markov chain is considered reversible.

  Thus, **tol** defines the threshold below which two floating-point values are treated as equal, enabling robust and practical reversibility checks despite numerical precision limits.

Conclusion

This project demonstrates the power and flexibility of Markov chains in modeling sequential decision-making under uncertainty. By combining real-world behavior modeling (fatigue, dropout) with formal Markov chain properties (stationarity, hitting times, absorption), the system provides both insightful simulations and theoretical analysis.

Such a model can be further generalized to domains like:

- Music or video streaming services
- Educational content progression

Example of Run

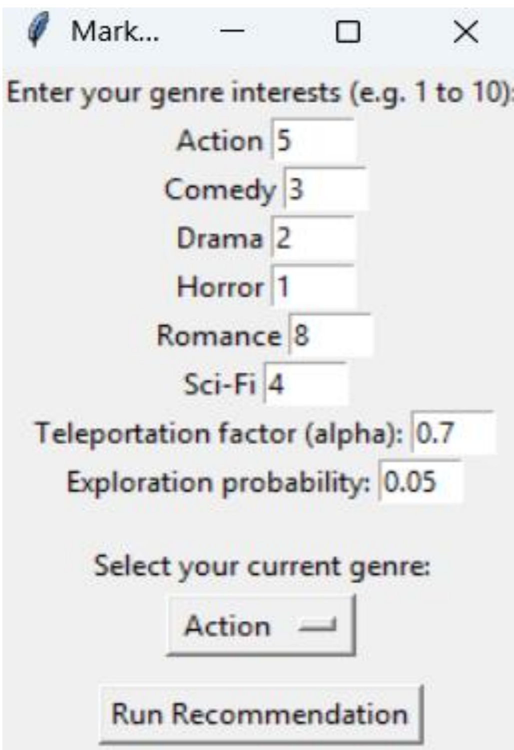

Figure 1 Sample Input

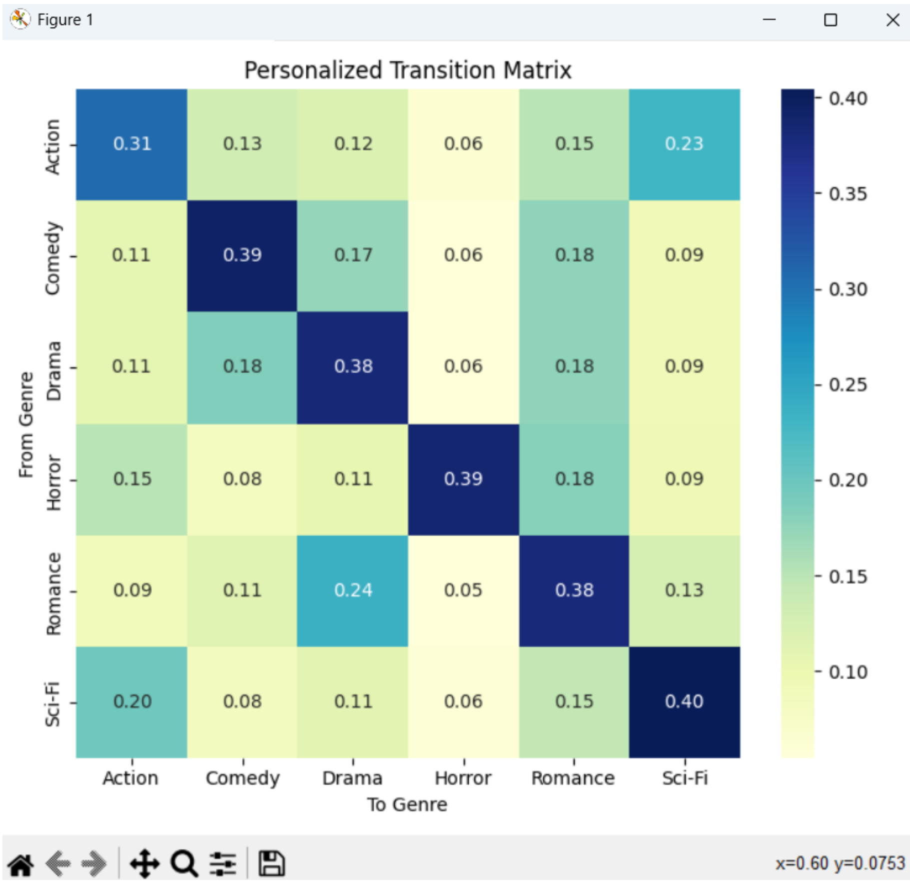

Figure 2 Generated Transition Matrix Based on Input

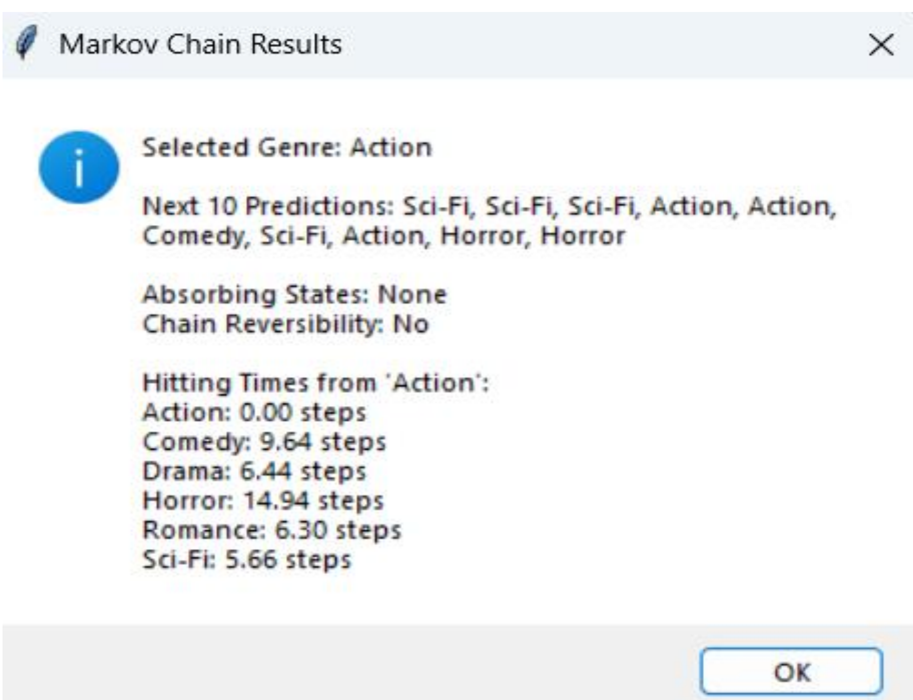

Figure 3 Output containing Prediction and checking absorbing states, Reversibility and Hitting Time