PageRank Algorithm

1. Introduction

The PageRank algorithm is a widely used technique for ranking nodes in a graph based on their relative importance. Originally developed by Larry Page and Sergey Brin, it is primarily used in search engines but has applications in other domains, such as recommendation systems. This report explores how the PageRank algorithm can be applied to develop a Movie Recommendation System, utilizing the Power Method and Limiting Probability for computation.

2. PageRank Algorithm

2.1 Definition and Concept

PageRank assigns a numerical value to each node (movie) in a graph, representing its importance based on the connections (links or relationships) with other nodes. The fundamental idea is that a node is more important if it is connected to other important nodes.

2.2 Mathematical Formulation

The PageRank score of a node P is given by:

$$PR(P) = (1 - d)/N + d * \Sigma (PR(P_i) / L(P_i))$$

where:

- $PR(P)$ = PageRank score of node P.

- d = Damping factor (usually set to 0.85), which represents the probability of following a link.

- N = Total number of nodes in the graph.

- P_i = Nodes linking to P.

- $L(P_i)$ = Number of outgoing links from P_i.

2.3 Computation using the Power Method

The Power Method is an iterative technique used to compute the dominant eigenvector of a matrix. Since the PageRank computation relies on the transition probability matrix, the Power Method is an effective approach for determining steady-state probabilities.

Steps:

1. Initialize the PageRank vector PR with equal probabilities for all nodes.

2. Multiply it iteratively by the transition matrix M: PR_new = M * PR_old

3. Continue until the difference between iterations is below a small threshold (convergence).

2.4 Limiting Probability

In Markov Chains, the Limiting Probability refers to the steady-state probability distribution of the system. As the Power Method iterates, the PageRank vector converges to this steady-state, ensuring that each movie in the recommendation system receives a stable importance score.

3. Project Idea: Movie Recommendation System

3.1 Problem Statement

Traditional movie recommendation systems rely on collaborative filtering or content-based filtering. However, these methods may not fully utilize the relationships between movies based on genres, storyline similarities, or ratings. By applying the PageRank algorithm, we can model movie recommendations as a graph-based ranking problem.

3.2 Graph Representation

- Nodes: Movies

- Edges: Similarity-based connections between movies

- Edge Weights: Cosine similarity score (based on storyline & genre) + rating-based weights

3.3 Application of PageRank

1. Construct a graph where movies are linked based on similarity.

2. Use the transition probability matrix to represent how users navigate between movies.

3. Apply PageRank with the Power Method to rank movies.

4. Recommend movies with the highest scores to users.

3.4 Advantages of Using PageRank

- Handles indirect relationships between movies.

- Does not suffer from cold-start issues as in collaborative filtering.

- Scalable for large datasets.

4. Implementation Overview

4.1 Data Collection

- Used a curated movie dataset containing:
    - **Titles**, **genres**, **plot summaries**, and optionally **user ratings**.
- Applied **TF-IDF vectorization** on movie plot summaries to capture textual features.
- Computed **cosine similarity** between movie vectors to establish **weighted edges** in a similarity graph.

4.2 Algorithm Execution

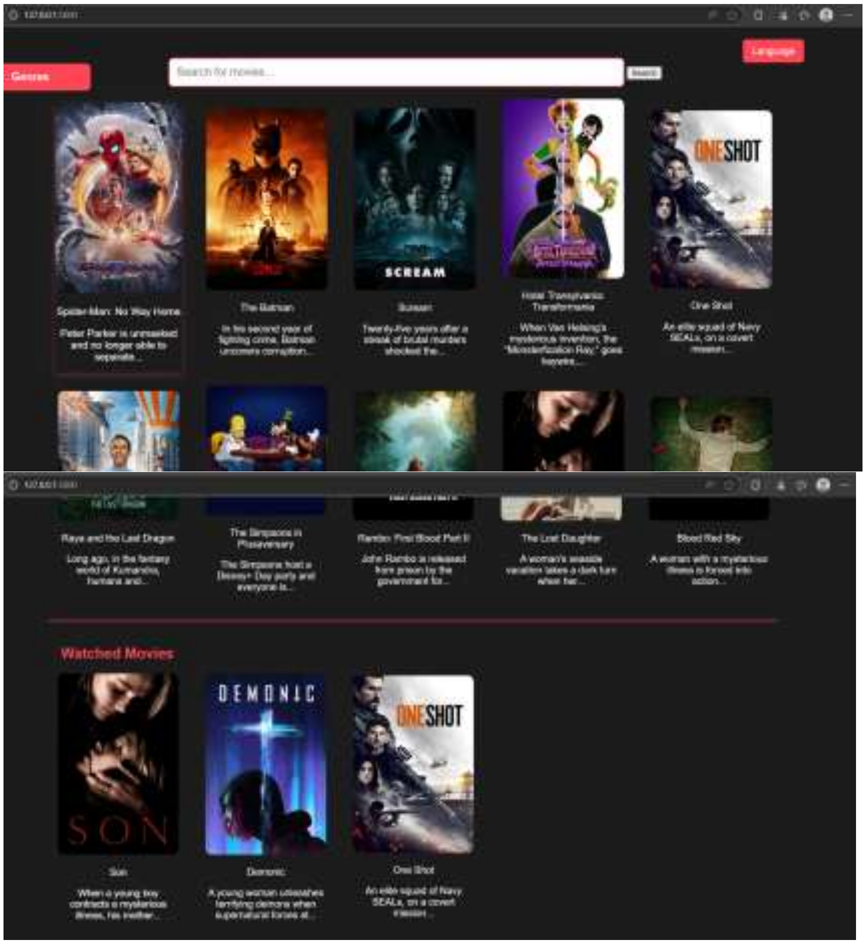
- **Graph Construction**
    - Created a similarity graph where each node is a movie.
    - Edges formed between similar movies using cosine similarity and optionally weighted with user ratings.
- **Transition Matrix**
    - Built a **stochastic transition matrix** from edge weights.
    - Integrated both **textual similarity** and **user rating weightage** (if available).
- **PageRank Computation**
    - Applied the **Power Iteration Method** to compute steady-state probabilities.
    - Each node's score represents its relative importance or recommendation strength.
- **Recommendation Ranking**
    - Sorted movies by their PageRank scores.
    - Top-ranked neighbors of the queried movie are recommended.

4.3 Evaluation

- Compared the system's recommendations with standard **collaborative filtering** results.
- Evaluation metrics included:

    - **User Feedback** (optional): through surveys or test users to assess satisfaction and relevancy.

5. Demo



6. Conclusion

By leveraging the PageRank algorithm, Power Method, and Limiting Probability, we can build an efficient movie recommendation system that ranks movies based on their intrinsic relationships. This approach improves recommendation quality and provides a robust alternative to traditional filtering methods.