Proposal: Movie Recommendation System using MCMC Methods

Recommendation services have played an important part in our daily lives. After we just finish watching a video on YouTube, for instance, a new selection of videos promptly appears on our screen, tailored to our individual interests. This personalized recommendation list is typically generated according to our previous rating history or preferences via statistical inference methods.

Our group members are:

Xiaoyang Sheng

Yulin Gao

Zicong Xiao

Problem:

The central challenge is predicting a customer's rating, $R \in [0.5, 5]$, for a newly introduced, yet unrated, movie. The prediction relies on a comprehensive history of **customer-movie-rating** tuples at hand. Our ultimate goal is to construct a system capable of predicting the rating score for any **customer-movie** pair within our dataset and subsequently recommending the movie to the user if the rating falls above a predefined threshold, like $R \geq 3.5$.

Dataset:

The *MovieLens 20M Dataset*, available via *Kaggle* (https://www.kaggle.com/datasets/grouplens/movielens-20m-dataset), encompasses a vast repository of **20,000,263** movie rating activities, across **27,278** movies, created by **138,493** customers, since 1995. **The customers are selected at random for inclusion.** The dataset primarily contains:

- *Rating* table

User_id	Movie_id	Rating	Timestamp

1	5999	3.5	2005-04-02 23:55:50
…	…	…	…

Rating table documents the rating activities for movies by customers. Each customer or movie within the dataset can be identified by a unique User_id, or Movie_id. The Rating column takes value from the range 0.5 to 5. The timestamp marks the point when the customer updates the rating.

- *Movie* table

Movie_id	Title	Genres
1	Toy Story (1995)	Adventure | Animation | Children | Comedy | Fantasy
…	…	…

Movie table shares in detail the information of each movie, including its title, publication year, and multiple genre classifications.

A potential limitation in the dataset is the sparse nature of customer ratings, where each customer rates only a limited number of movies, and each movie receive ratings from a relatively small pool of customers, compared with the whole population. We would do data cleaning work by removing inactive customers or outdated movie information.

Method & Plan:

1. Initiate the process by conducting EDA and essential data cleaning work mentioned. All records will be sorted by timestamp. The earlier 80% records for each user will be used as training set, and the remaining 20% will fall in test set instead.
2. Proceed by transforming the *Rating* table into a pivot matrix denoted as $\mathbf{A}^{N\times M}$ where the rows represent customers and the columns represent movies. Each element A[i, j] indicates the rating given by User_i to Movie_j.
3. Apply Bayesian Probabilistic Matrix Factorization (BPMF) algorithm [1] to solve:

$$\mathbf{A}^{N\times M} = \mathbf{U}^{N\times P} * \mathbf{V}^{P\times M}$$

Assume the target distribution $\mathbf{P(A \mid U, V, \Theta)}$ and priors $\mathbf{P(U \mid \Theta_U)}$, $\mathbf{P(V \mid \Theta_V)}$, and repeat the MCMC sampling:

Initialize **U, V, Θ**

For **t** in 1, …, T:

Sample $\mathbf{\Theta_U}$

Sample $\mathbf{\Theta_V}$

For **i** in 1, …, N:

Sample $\mathbf{U_i}$

For j in 1, …, M:

Sample $\mathbf{V_j}$

Compute **U*V** and append to **Results**

4. Adopt metrics, like RMSE to evaluate the performance on training and test set.
5. Improve the existing algorithm to produced more personalized results. We will join the *Rating* table as well as the *Movie* table, and introduce Hierarchical Modelling (HM) to divide customers into multiple groups. We will carry out again the matrix factorization algorithm within each customer group, and compare the performance.

Reference:

[1] Ruslan Salakhutdinov and Andriy Mnih. 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In Proceedings of the 25th international conference on Machine learning (ICML '08). Association for Computing Machinery, New York, NY, USA, 880–887. https://doi.org/10.1145/1390156.1390267