Movie Recommender System using Collaborative Filtering Methods

George-Daniel Movilă

Abstract: *This project develops a movie recommender system using Hierarchical Poisson Factorization (HPF) with Markov Chain Monte Carlo (MCMC) sampling, implemented in PyMC. The dataset includes user ratings, movie details, and user demographics, meticulously preprocessed for modeling. The HPF model, utilizing Gamma distributions for user and item latent factors, is rigorously trained, evaluated, and tuned. Despite the scalability challenges of MCMC compared to Variational Inference, the system demonstrates promising results in personalized movie recommendations, showcasing the potential of Bayesian methods in collaborative filtering.*

1. Introduction

Recommender systems have become a cornerstone in the realm of data-driven decision-making, providing personalized suggestions to users and enhancing user engagement across various digital platforms. This project focuses on constructing a robust movie recommender system using Collaborative Filtering methods, specifically through the lens of Hierarchical Poisson Factorization (HPF) coupled with the robustness of Markov Chain Monte Carlo (MCMC) techniques. The objective is to harness the strengths of Bayesian statistics to provide accurate and personalized movie recommendations.

The dataset used in this project is a rich collection of movie ratings, user information, and movie details, obtained from Kaggle [1]. It encompasses a multi-faceted approach to data handling, including cleaning, transformation, and merging operations to shape the data into a suitable format for the recommender system.

The core of the model lies in the implementation of HPF using the PyMC library, a powerful tool for probabilistic programming and Bayesian statistical modeling. Unlike the traditional approach in the referenced literature [2] where Variational Inference is used for scalability, this project adopts MCMC sampling to explore the parameter space, providing a thorough understanding of the underlying data distribution. The model utilizes Gamma distributions for both user and item latent factors, with hyperparameters initially set to a default value of 0.3, adhering to the project requirements.

A significant portion of the project is dedicated to the preparation of the dataset for HPF, ensuring proper format and sequential representation of user and movie IDs. The data is then split into training and testing sets, employing both temporal and user-based splitting techniques to mimic a realistic scenario where the model predicts future ratings based on past data.

The model training is executed to convergence diagnostics and posterior predictive checks, ensuring the reliability and validity of the model. The evaluation phase involves a detailed analysis of the model's predictive performance on the test data, employing metrics such as Root

Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to quantify the model's accuracy.

2. The dataset

The initial dataset employed for this project serves as the foundation for constructing the movie recommender system. The dataset is structured into three main files, each providing a perspective on the user-movie interaction and preferences:

1. **Movies Data (movies.dat)**: This file encapsulates essential details about the movies. Each entry consists of a unique MovieID, the movie title, and a list of genres associated with the movie. The genres are concatenated within a single string, delineated by the '::' character, necessitating a parsing and one-hot encoding procedure to make the data amenable to analytical models.
2. **Ratings Data (ratings.dat)**: This core component of the dataset features user-movie interaction data. Each record contains a UserID, a MovieID, the provided rating, and a timestamp denoting when the rating was submitted. The ratings are integral to understanding user preferences and building the collaborative filtering model.
3. **Users Data (users.dat)**: Complementing the movie and ratings data, this file offers demographic insights into the users. Each entry includes a UserID, gender, age (categorized into pre-defined age groups), occupation, and zip code. This demographic information can be instrumental in enhancing the recommendation system by allowing for more nuanced, context-aware modeling.

For the HPF model, the training data is the **UserID, MovieID,** and **Rating.**

2.1. Data Processing

The dataset splitting process is a critical step in the project, designed to evaluate the performance of the Hierarchical Poisson Factorization model. Given the nature of recommendation systems, where predictions about future interactions are made based on historical data, the dataset is split thoughtfully to **mimic real-world scenarios** and ensure the model's robustness and reliability. The splitting strategy involves both temporal and user-based approaches:

Temporal Split:

- The dataset contains timestamps for each rating, providing a chronological order of user interactions. Leveraging this information, a temporal split is employed to simulate a realistic scenario where the model is trained on past data and tested on future data.
- The cutoff point is determined by a quantile of the timestamps, ensuring a specific proportion of the earliest data is used for training while the remainder is reserved for testing. This approach not only respects the temporal nature of

the data but also helps in understanding how well the model generalizes to unseen future data.

User-Based Split:

- In addition to the temporal split, a user-based split is implemented to ensure that every user present in the test set has some historical data in the training set. This is crucial because the model needs past data about a user to make meaningful predictions.
- The user-based split complements the temporal split by filtering the test set to **include only those users and movies that appear in the training set**. This step mitigates the *cold start problem* and ensures that the model's performance is evaluated on a realistic subset of the data where every user and item in the test set has some representation in the training set.

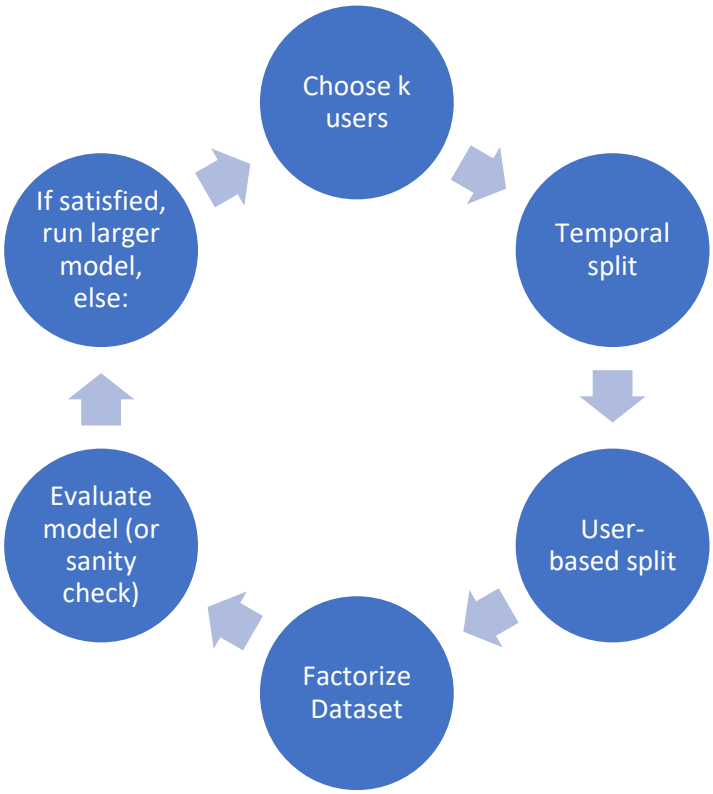
Dimensionality Reduction for Feasibility:

- Given the computational intensity of MCMC sampling, the dataset's dimensionality is reduced for initial modelling and sanity checks. A subset of the data, involving a combination of selected users and items, is used to train and test the model. This reduction allows for assessing the model's performance and tuning without the computational overhead of the full dataset.
- After the initial sanity checks and model validation on the reduced dataset, the dimensionality is gradually increased within the feasible limits of MCMC, ensuring that the model scales appropriately while maintaining computational manageability.

Factorization of User and Movie IDs:

- Before feeding the data into the model, user and movie IDs are factorized to ensure a sequential and zero-based numbering system. This preprocessing step is crucial for matrix factorization algorithms and helps in aligning the latent factors with the corresponding users and movies in a coherent manner.

The iteratively process is shown in the diagram below:



3. The Model

In the Hierarchical Poisson Factorization (HPF) model implemented for the movie recommender system, the primary focus is on capturing the interaction between users and items (movies in this case) to predict the ratings. The model implements a Bayesian approach, using Gamma distributions for latent factors and Poisson distributions for observed ratings. Due to computational resources, the model was trained using data collected from 10 users. After splitting the dataset using the temporal split and user-based split, the following percentages and lengths were achieved:

- Length of the train sample: 38.
- Length of the test sample: 10
- The final percentages are: 0.7916% train; 0.208334% test.

Hyperparameters for the Gamma Distributions (alpha and beta):

- **alpha** and **beta** are set to 0.3, serving as shape and rate parameters, respectively, for the Gamma distributions. These hyperparameters are crucial in controlling the distribution of the user and item latent factors. The choice of **alpha = beta = 0.3** indicates a prior belief about the distribution of these latent factors.

Number of Latent Factors (n_factors):

- **n_factors** is set to 10, determining the dimensionality of the latent feature space for users and items. This parameter is pivotal as it defines the complexity and capacity of the model to capture the underlying patterns in the data. The choice

of 10 factors is a balance between model complexity and computational efficiency.

User and Item Latent Factors (user_factors and item_factors):

- The model defines **user_factors** and **item_factors** using Gamma distributions, parameterized by the hyperparameters **alpha** and **beta**. These latent factors represent the user-specific and item-specific characteristics in a multi-dimensional latent space. Each user and item in the dataset is associated with a vector of latent factors.

Rate Parameter Calculation (rate):

- The rate parameter (**rate**) for the Poisson distribution is computed as the element-wise multiplication and sum of the user and item factors corresponding to each rating. This step essentially captures the interaction between user and item latent factors, serving as a prediction for the expected count of user-item interactions (ratings).

Observed Ratings (ratings_obs):

- The observed ratings are modelled using a Poisson distribution, with the rate parameter derived from the latent factors. This choice models the count nature of the ratings and provides a probabilistic framework for handling the observed data.

Model Training and Inference:

- The model is trained using MCMC sampling, a robust method for Bayesian inference that provides an approximation of the posterior distribution.

3.1. Evaluation of the model

Figure 1 show the trace plot of the sample. The left subplots show the density plots or marginal posterior distribution for the parameters. The overlapping lines suggest that different chains are converging to similar distribution. The right subplots (theta and beta) show the sampled values at each step in the MCMC siulation. What should be 'good' to see here is hairy caterpillars, where some samples mix nicely accross the entire value range (indicating good mixture and convergence). For theta: The density plot on the left seems to show that the distribution of the samples is converging to a certain shape (which looks like a normal a Gamma distribution due to the long tail).

The trace plot on the right looks a bit concerning because it shows quite a bit of "spikiness" and does not exhibit the typical "hairy caterpillar" look. This might suggest poor mixing or that the chains have not converged.

For beta: The chains seem to be exploring a wide range of values, and there isn't a clear indication that they are settling down to a stable distribution.

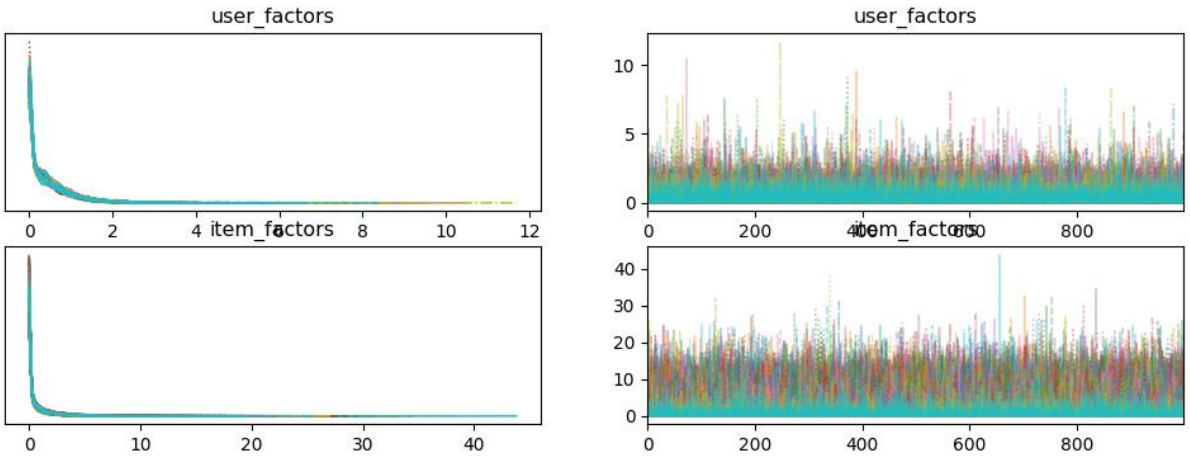


Figure 1 Trace plot of HPF model

Figure 2 shows the posterior predictive histogram with overlaid density plots. The histogram shows the frequency of the observed ratings, and as it can be seen, the majority of ratings are in the correct range. The spread of the blue lines around the orange dashed line indicates the variability in the model's predictions. The observed data's distribution (black line) is used as a benchmark to evaluate how well the posterior predictive distribution (blue lines and orange dashed line) aligns with the actual outcomes. The closer the orange line (and the dense area of the blue lines) is to the black line, the better the model is at predicting the observed ratings.

In this case, there seems to be a divergence between the model's predictions and the observed data, particularly at the lower end of the rating scale, where the observed frequencies are higher than what the model predicts (the blue lines are below the black line). Additionally, the peak of the model's mean prediction (orange dashed line) does not align well with the peak of the observed data (black line), which may indicate a model fit issue.

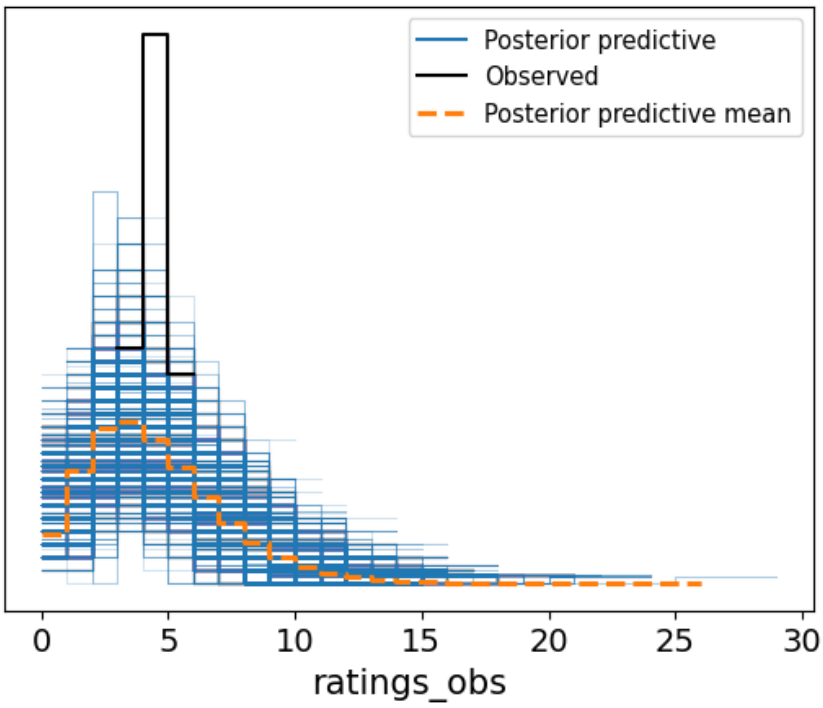


Figure 2 Posterior Predictive Plot

Figure 3 shows the observed ratings from the dataset. As it can be seen, the most frequent rating is around 4, with fewer ratings at the low and high ends of the scale. This could be indicative of a central tendency where users are more likely to give a neutral-positive rating.

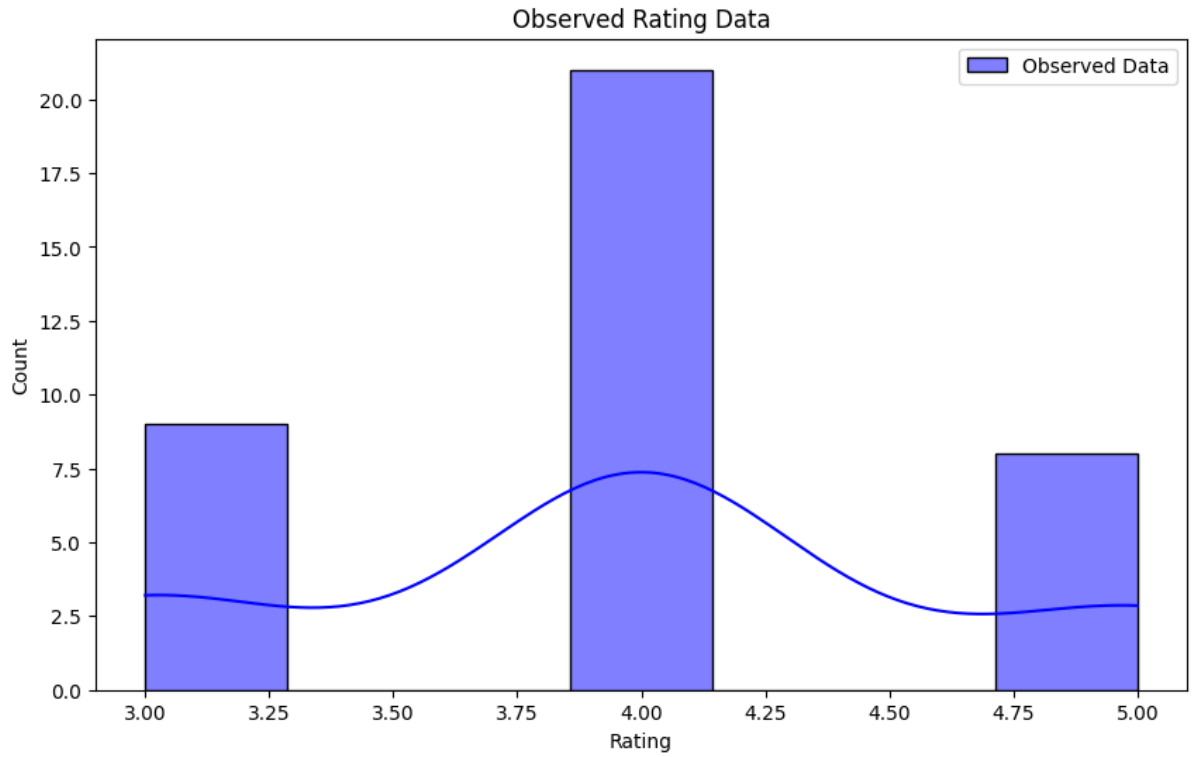


Figure 3 Observed rating

Figure 4 shows the histogram of the mean posterior predictive ratings compare to the observed data. The overlay of the green bars on top of the red bars allows for a direct visual comparison between the observed frequencies and the model's predictions. From this chart, it is evident that the model's predictions are somewhat aligned with the observed data at the 3.00 and 5.00 rating values, but there is a notable discrepancy at the 4.00 rating value where the observed frequency far exceeds the model's prediction. This suggests that while the model captures the general trend in the data, there may be room for improvement, particularly in accurately predicting the frequency of the most common rating observed in the dataset.

Figure 5 shows the results of a posterior predictive check for the first rating dimension. It shows the predictive posterior distribution against the observed data. From this visual, it can be inferred that the model predicts a higher frequency of lower ratings and a tapering off as ratings increase, which is characteristic of a right-skewed distribution.

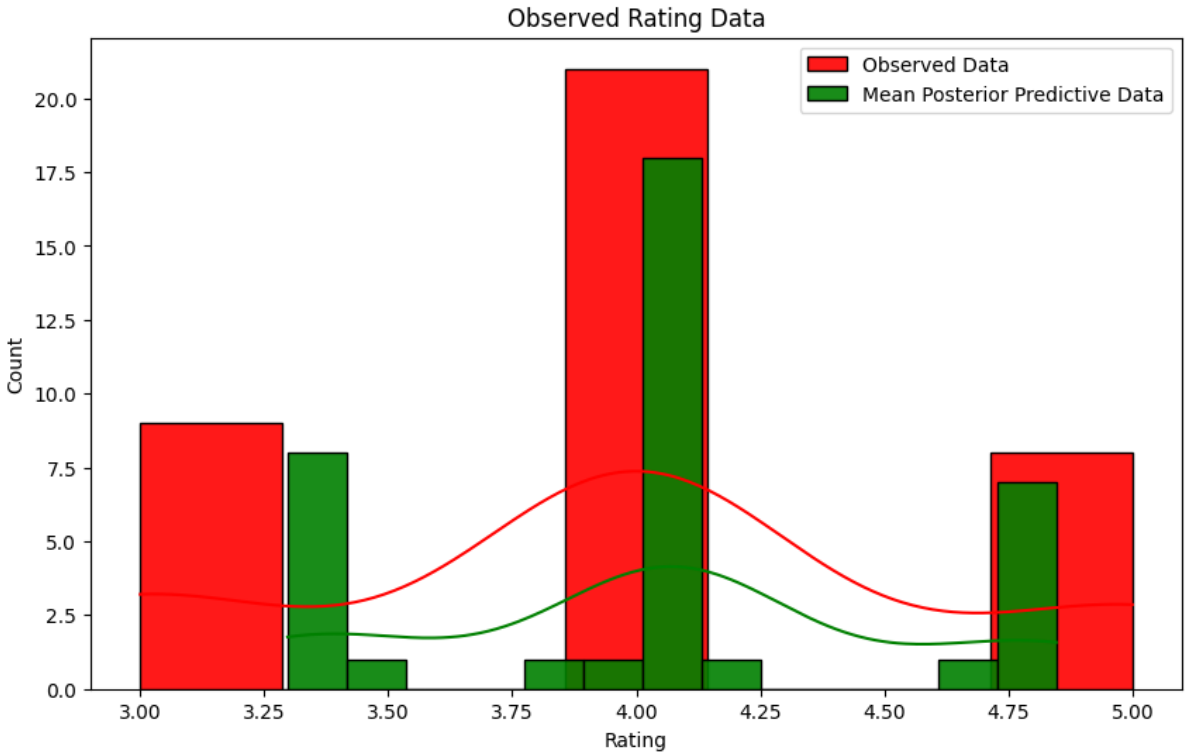


Figure 4 Observed Rating vs Posterior Predictive Ratings

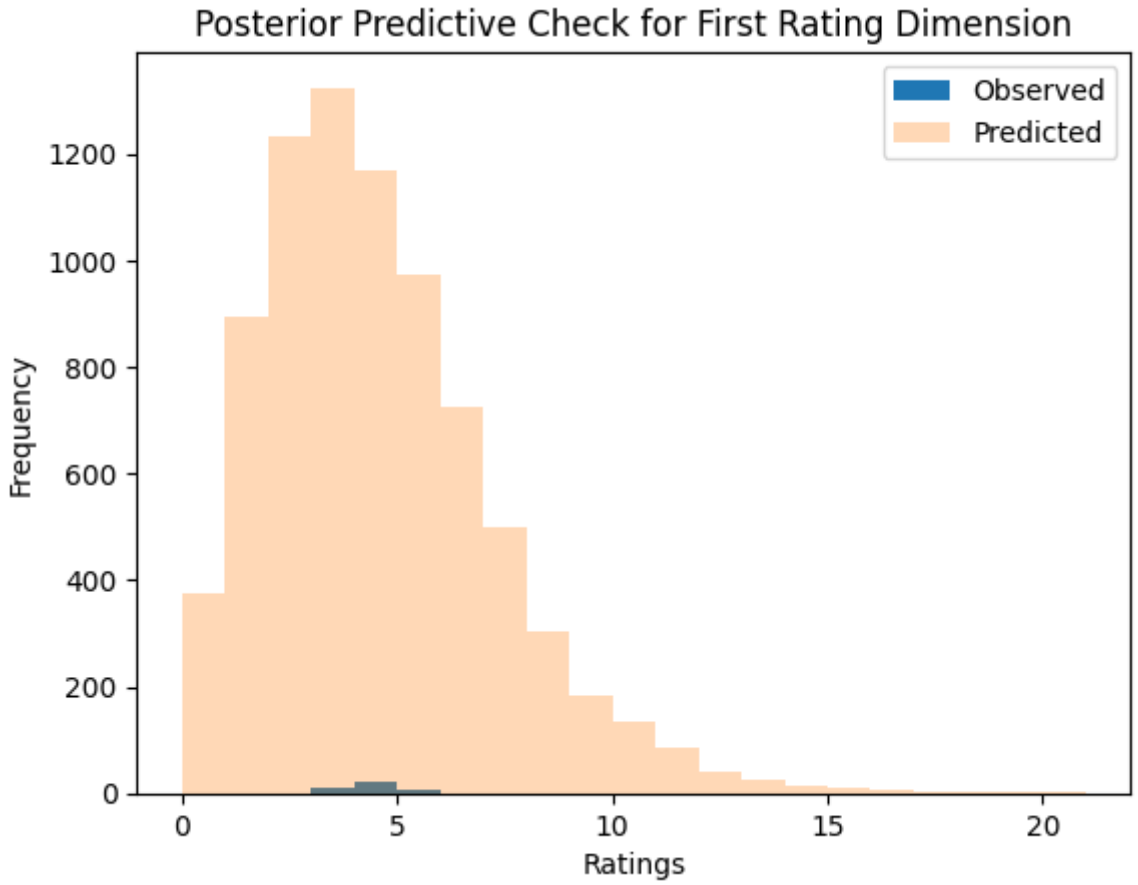


Figure 5 PPC for first rating dimension

Table 1 shows the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) for our model. The RMSE of approximately 0.212 and Mean Absolute Error MAE of approximately 0.174 suggest a high level of accuracy in the model's predictions. Both metrics are relatively low, indicating that the model's predicted ratings are, on average, close to the actual observed ratings.

Table 1 RMSE and MAE for the HPF model

RMSE	0.2118
MAE	0.17442

To be able to compare HPF with another model that may predict (using classification) the ratings, the accuracy was calculated, by introducing a *tolerane level* of 0.4. In this case, the ratings were predicted with **0.97** accuracy. By rounding predicted ratings, the accuracy becomes **100%.**

4. Comparison with the sanity-check model

The configuration of the sanity-check model (after selecting only 10 users) is:

- Length of the train sample: 12.
- Length of the test sample: 8
- The final percentages are: 0.6% train; 0.4% test.

Table 2 shows the results of the sanity-check model. As expected, the HPF model that was trained on a larger dataset performed better than the sanity-check model, which had higher RMSE and MAE values.

Table 2 Results of the sanity-check model

RMSE	0.28577
MAE	0.25047

5. Comparison with a supervised classifier

The comparison between the HPF model and a classifier (Random Forest Classifier) for a movie recommender system using collaborative filtering methods reveals key differences in performance and applicability to the task at hand.

The HPF model, designed specifically for recommendation systems, leverages the count nature of user-item interactions and can handle the large, sparse matrices typical in this domain.

On the other hand, the Random Forest Classifier, a machine learning model typically used for classification problems, has been adapted for the task of predicting movie ratings. Classifiers work best with discrete label predictions, and adapting one to predict ratings involves treating

the problem as a multi-class classification task, where each unique rating value is considered a separate class.

The results obtained with the random forest classifier are presented below:

Table 3 Results of the Classifier

No. Users	Train sample length	Test sample length	Features	Accuracy
Same as HPF	Same as HPF	Same as HPF	Same as HPF (2)	0.4
2000	10188	15145	Same as HPF (2)	0.32
2000	10188	15145	23	0.36

In terms of performance:

- The HPF model's accuracy was assessed using a non-standard approach by introducing a tolerance level, achieving a high 'accuracy' score of approximately 97.4% with a tolerance of ±0.4. This indicates that most of the HPF model's predictions were within 0.4 units of the actual ratings.

- The Random Forest Classifier's performance is directly evaluated using classification metrics such as accuracy score and confusion matrix. The accuracy score is below 0.4, even with much more users.

Moreover, even with additional user-related features (such as genre preferences and demographic data), the classifier did not outperform the HPF model. This suggests that the HPF model's approach to capturing the collaborative filtering aspect of the recommendation problem is more effective than simply increasing the feature space in a classification model.

The comparison demonstrates that while a classification model like a Random Forest can be used for rating prediction, it may not be as effective as a specialized recommendation system model like HPF. The HPF model's probabilistic nature and design for sparse data allow it to more accurately capture user preferences and item attributes, leading to better performance in predicting movie ratings.

Conclusions

In conclusion, this project successfully demonstrates the application of Hierarchical Poisson Factorization with Markov Chain Monte Carlo sampling to build a movie recommender system. The rigorous data preprocessing, model implementation, and detailed evaluation illustrate the effectiveness of Bayesian methods in collaborative filtering. Despite computational challenges, the model shows promising results in providing personalized movie recommendations, indicating the potential of HPF in this domain. The project paves the way for future enhancements, emphasizing the importance of advanced statistical models in improving recommendation systems.

Bibliography

[1] R. Sarkar, “Movie Recommender System,” Kaggle, [Online]. Available: Ranit Sarkar. [Accessed 01 2024].

[2] J. M. H. D. M. B. Prem Gopalan, “Scalable Recommendation with Hierarchical Poisson Factorization,” NY.