CMP2003 Project: Rating Prediction Using Matrix Factorization

Group Name: Data Manipulators

Team Members:

Hüseyin Can Gülkan

Ahmet Erbey

Gökhan Yavuz

Raouf Alipour

December 28, 2024

Abstract

This project implements a recommendation system to predict user-item ratings using matrix factorization and compares its performance with Item-Based Collaborative Filtering (IBCF) using cosine similarity. The matrix factorization approach decomposes the user-item interaction matrix into global averages, user biases, item biases, and latent factors, optimizing parameters with Stochastic Gradient Descent (SGD) to minimize Root Mean Squared Error (RMSE). IBCF, on the other hand, predicts ratings based on item similarities calculated using cosine similarity. Results demonstrate that matrix factorization outperforms IBCF in accuracy and scalability, especially in handling sparse datasets, while also providing insights into user and item biases. This paper also discusses the implementation challenges, evaluation methods, and potential improvements for future research.

1 Introduction

Recommender systems have become integral to personalizing user experiences in domains such as movies, music, and e-commerce. This project aims to predict user-item ratings through matrix factorization, capturing global trends, individual user preferences, and item-specific characteristics.

The dataset comprises user-item-rating triples for training and user-item pairs for testing. The goal is to accurately predict these hidden ratings while minimizing RMSE and ensuring scalability through efficient data structures and algorithms. Alongside, Item-Based Collaborative Filtering (IBCF), a neighborhood-based method, is explored for comparative analysis.

Modern applications, such as Netflix, Amazon, and Spotify, underscore the importance of robust recommendation systems. These systems leverage user interactions and item metadata to suggest relevant items, driving user engagement and sales. However, challenges such as data sparsity, scalability, and evolving user preferences necessitate advanced approaches like matrix factorization.

1.1 Objective

The primary objectives of this project are:

- Implement a matrix factorization model using SGD.
- Develop an IBCF algorithm leveraging cosine similarity.
- Compare the performance of both approaches in terms of RMSE, scalability, and interpretability.
- Investigate efficient data structures to handle sparse datasets.
- Highlight potential improvements for future research and applications.

1.2 Scope of the Study

The study focuses on two key aspects:

- Evaluating the practical implementation of matrix factorization and IBCF.
- Assessing the trade-offs between interpretability, accuracy, and scalability in real-world scenarios.

—

2 Prediction Algorithm: Matrix Factorization

2.1 Explanation of Matrix Factorization

Matrix factorization is a powerful technique widely used in recommendation systems to predict user-item interactions. It decomposes a large, sparse user-item interaction matrix into smaller matrices that capture latent features of users and items. The goal is to model the observed ratings by learning these latent factors, which encapsulate hidden preferences of users and characteristics of items.

In this technique, each user and item is represented by a vector of latent factors. For example, in a movie recommendation system, these factors could correspond to implicit attributes such as genres, directors, or emotional tone. The dot product of a user's latent factor vector and an item's latent factor vector predicts the interaction between the two, which can be interpreted as a rating.

Matrix factorization effectively handles data sparsity, a common challenge in recommendation systems where many user-item interactions are unknown. By focusing on the latent dimensions, it generalizes well to unseen data. The approach is also scalable to large datasets due to efficient optimization methods like Stochastic Gradient Descent (SGD).

Key benefits include:

- Capturing implicit patterns in user-item interactions.
- Reducing dimensionality while preserving meaningful relationships.
- Incorporating biases (e.g., user and item biases) to improve accuracy.

The flexibility and interpretability of matrix factorization make it a cornerstone of modern recommendation algorithms, and it forms the foundation of many hybrid models that integrate additional contextual information.

2.2 Mathematical Formulation

The predicted rating $\hat{R}_{u,i}$ for user u and item i is given by:

$$\hat{R}_{u,i} = \mu + b_u + b_i + U_u \cdot V_i^T$$

Where:

- μ: Global average rating.
- b_u: User bias, capturing user-specific tendencies.
- b_i: Item bias, reflecting item popularity.
- U_u, V_i: Latent factor vectors for user u and item i, respectively.

The objective function is:

$$\text{Objective} = \sum_{(u,i)\in R} \left(R_{u,i} - \hat{R}_{u,i}\right)^2 + \lambda \left(\|U_u\|^2 + \|V_i\|^2 + b_u^2 + b_i^2\right)$$

Where λ is a regularization parameter to prevent overfitting.

2.3 Explanation of λ, α, and numFactors

- λ: This regularization term controls the complexity of the model by penalizing large values of the parameters (latent factors and biases). It helps prevent overfitting by discouraging the model from fitting noise in the training data.
- α: This is the learning rate, which determines the step size during parameter updates in Stochastic Gradient Descent (SGD). A smaller α leads to slower convergence but ensures stability, while a larger α may speed up convergence but risks overshooting the optimal solution.
- numFactors: This refers to the number of latent factors used to represent users and items in the matrix factorization model. Each user and item is characterized by a vector of size numFactors, which captures implicit characteristics such as preferences or item attributes.

2.4 User and Item Latent Factors

- **User Latent Factors:** Represent a user's underlying preferences for certain types of items. For example, a user might have high latent values for genres like action or comedy in a movie recommendation system.
- **Item Latent Factors:** Represent inherent attributes of items, such as genre, price range, or popularity. These factors interact with user latent factors to predict ratings.

Latent factors are learned during training and enable the model to capture implicit patterns in the user-item interaction data.

2.5 Optimization Using Stochastic Gradient Descent (SGD)

Parameters are updated iteratively using SGD:

$$\begin{aligned} b_u &\leftarrow b_u + \alpha \left(e_{u,i} - \lambda b_u\right) \\ b_i &\leftarrow b_i + \alpha \left(e_{u,i} - \lambda b_i\right) \\ U_{u,k} &\leftarrow U_{u,k} + \alpha \left(e_{u,i} V_{i,k} - \lambda U_{u,k}\right) \\ V_{i,k} &\leftarrow V_{i,k} + \alpha \left(e_{u,i} U_{u,k} - \lambda V_{i,k}\right) \end{aligned}$$

2.6 Advantages of Matrix Factorization

Matrix factorization provides several advantages over traditional methods:

- **Latent Factor Modeling:** Captures implicit user preferences and item characteristics.
- **Bias Handling:** Explicitly accounts for user and item biases, improving prediction accuracy.
- **Scalability:** Handles large datasets efficiently through SGD and sparse representations.
- **Flexibility:** Adapts to different types of input data, including implicit feedback.

—

3 Implementation

This section describes the key functions implemented in the recommendation system.

3.1 Initialize Matrices and Biases

Function Name: `initializeMatrices()`

This function initializes the latent factor matrices U and V with small random values. It also sets the user and item biases to zero. Random initialization ensures that the optimization process begins from an unbiased starting point.

3.2 Compute Global Average Rating

Function Name: `computeGlobalAvgRating()`

This function calculates the global average rating across all training data. The global average serves as a baseline for predictions when user- or item-specific information is unavailable. It is computed as:

$$\mu = \frac{\sum_{(u,i)\in R} R_{u,i}}{|R|}$$

where $|R|$ is the total number of ratings.

3.3 Predict Rating

Function Name: `predictRating(int userID, int itemID)`

This function predicts the rating for a specific user-item pair using the formula:

$$\hat{R}_{u,i} = \mu + b_u + b_i + \sum_{k=1}^{f} U_{u,k} V_{i,k}$$

where f is the number of latent factors. The prediction is clamped to the range [1, 5].

3.4 Stochastic Gradient Descent (SGD)

Function Name: `stochasticGradientDescent()`

This function optimizes the parameters U, V, b_u, b_i by minimizing the prediction error:

$$e_{u,i} = R_{u,i} - \hat{R}_{u,i}$$

The parameters are updated iteratively to reduce $e_{u,i}$, ensuring better predictions.

3.5 Main Function

Function Name: `main()`

The main function orchestrates the following steps:

- Reads training and test datasets.
- Computes the global average rating.
- Initializes latent factors and biases.

- Trains the model using SGD.
- Predicts ratings for test user-item pairs.

—

4 Comparison: IBCF Using Cosine Similarity vs. Matrix Factorization

4.1 Item-Based Collaborative Filtering (IBCF)

IBCF predicts ratings based on item similarities using cosine similarity:

$$\text{Similarity}(i,j) = \frac{\sum_{u \in U} R_{u,i} R_{u,j}}{\sqrt{\sum_{u \in U} R_{u,i}^2} \sqrt{\sum_{u \in U} R_{u,j}^2}}$$

The predicted rating is:

$$\hat{R}_{u,i} = \frac{\sum_{j \in N(i)} \text{Similarity}(i,j) \cdot R_{u,j}}{\sum_{j \in N(i)} |\text{Similarity}(i,j)|}$$

Where $N(i)$ is the set of similar items.

4.2 Comparison

Aspect	IBCF with Cosine Similarity	Matrix Factorization
Simplicity	Easy to implement	More complex(requires optimization)
Interpretability	High	Low (latent factors)
Accuracy	Moderate	High (handles sparsity better)
Scalability	Limited for large datasets	Scales well
Bias Handling	Ignores biases	Explicitly models biases

Table 1: Comparison of IBCF and Matrix Factorization

4.3 Limitations of IBCF

- **Sparsity Issues:** Requires sufficient overlap in ratings to compute similarities.
- **Scalability Challenges:** Computationally expensive for large datasets.
- **Static Nature:** Does not adapt well to dynamic user behavior.

—

5 Evaluation Metric: RMSE

The Root Mean Squared Error (RMSE) measures prediction accuracy:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Where:

- y_i: Actual rating.
- $\hat{y}_i$: Predicted rating.
- n: Total number of predictions.

A lower RMSE indicates better accuracy.

—

6 Conclusion and Future Work

Matrix factorization proves to be a robust method for rating prediction, outperforming IBCF in accuracy and scalability. Efficient data structures, such as hash maps, ensure scalability for large datasets. Stochastic Gradient Descent (SGD) ensures optimal parameter tuning, balancing accuracy and computational efficiency.

Future work may include:

- **Hybrid Models:** Combining matrix factorization with neighborhood-based methods for enhanced performance.
- **Deep Learning Approaches:** Exploring Neural Collaborative Filtering and Autoencoders for advanced latent factor modeling.
- **Dynamic Systems:** Developing models that adapt to evolving user preferences in real-time.
- **Alternative Metrics:** Investigating the use of Mean Absolute Error (MAE) and Precision/Recall alongside RMSE.
- **Explainability:** Enhancing model interpretability to improve user trust and engagement.

7 Resources

The following resources were utilized for the development and understanding of the recommendation system in this project:

- Koren, Y., Bell, R., Volinsky, C. (2009). *Matrix Factorization Techniques for Recommender Systems*. IEEE Computer Society. DOI: 10.1109/MC.2009.263.
- Ricci, F., Rokach, L., Shapira, B. (2011). *Introduction to Recommender Systems Handbook*. Springer.

- He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T. (2017). Neural Collaborative Filtering. Proceedings of the 26th International Conference on World Wide Web.
- Scikit-learn Documentation on Collaborative Filtering
- Towards Data Science Blog: Matrix Factorization for Recommender Systems.

These resources provided foundational concepts, implementation strategies, and datasets for experimentation and validation of the recommendation system.