Title: Low-Rank Matrix and Tensor Factorization for Movie Recommendation using the MovieLens 100K Dataset

Motivation:

The goal of this project is to explore how **Matrix Factorization (MF)** and **Tensor Factorization (TF)** techniques can be used to build an efficient movie recommender model.

MF is widely known for its success in collaborative filtering systems, while TF (especially CP decomposition) extends MF by introducing a third dimension such as movie genres or time to model richer relationships.

This project implements and compares both 2D and 3D approaches on the classic MovieLens 100K dataset using PyTorch, focusing on simplicity, interpretability, and learning efficiency.

Literature Review:

Recommender systems have evolved through several stages:

Early methods:
Content-based and neighborhood-based collaborative filtering (Resnick et al., 1994) were the first successful techniques but struggled with sparsity and scalability.

Matrix Factorization:
The breakthrough came with latent factor models, particularly Matrix Factorization (Koren, Bell & Volinsky, 2009, IEEE Computer), which powered the Netflix Prize-winning algorithm.
It models users and items in a shared low-dimensional space, capturing user preferences and item attributes.

Tensor Factorization:
Later work extended MF into higher dimensions using **Tensor Decompositions**, such as:

Kolda & Bader (2009), SIAM Review, "Tensor Decompositions and Applications"

Karatzoglou et al. (2010), Proceedings of the IEEE, "Multiverse Recommendation: N-dimensional Tensor Factorization for Context-Aware Collaborative Filtering"

These works inspired the idea of modeling **user–item–context** interactions instead of just user–item pairs, making recommendations more context-sensitive.

Dataset: MovieLens 100K:

Source: GroupLens MovieLens 100K Dataset

- **Size:** 100,000 user–movie ratings
- **Users:** 943
- **Movies:** 1,682
- **Ratings:** integer values from 1 to 5
- **Genres:** 19 binary genre indicators per movie

File	Description
`u.data`	User–item–rating–timestamp matrix
`u.user`	User demographics (age, gender, occupation, zip)
`u.item`	Movie titles and genre flags

- **Sparsity:** 6.3%, only a small fraction of possible user–movie pairs have ratings.
- **Genre Distribution:** Drama, Comedy, and Action dominate.
- **Rating Distribution:** Most ratings cluster between 3 and 4, indicating user bias toward moderate-to-high scores.

Implementation Overview:

Preprocessing

- Converted u.data, u.user, and u.item into clean DataFrames
- Normalized IDs for users, items, and genres
- Built user–item matrices (for MF) and user–item–genre triplets (for TF)

2D Model: Matrix Factorization (MF)

Matrix Factorization learns two latent matrices:

$$R_{ui} \approx P_u \cdot Q_i^T$$

Where:

- R_{ui}: observed rating
- P_u: latent user vector
- Q_i: latent item vector

Each user and movie is represented as a 32-dimensional vector (n_factors = 32), and their dot product predicts the rating.

Implementation

- Used nn.Embedding layers for user and item vectors
- Initialized weights with small random values
- Used **Adam optimizer** and **MSELoss**
- Trained for 20 epochs

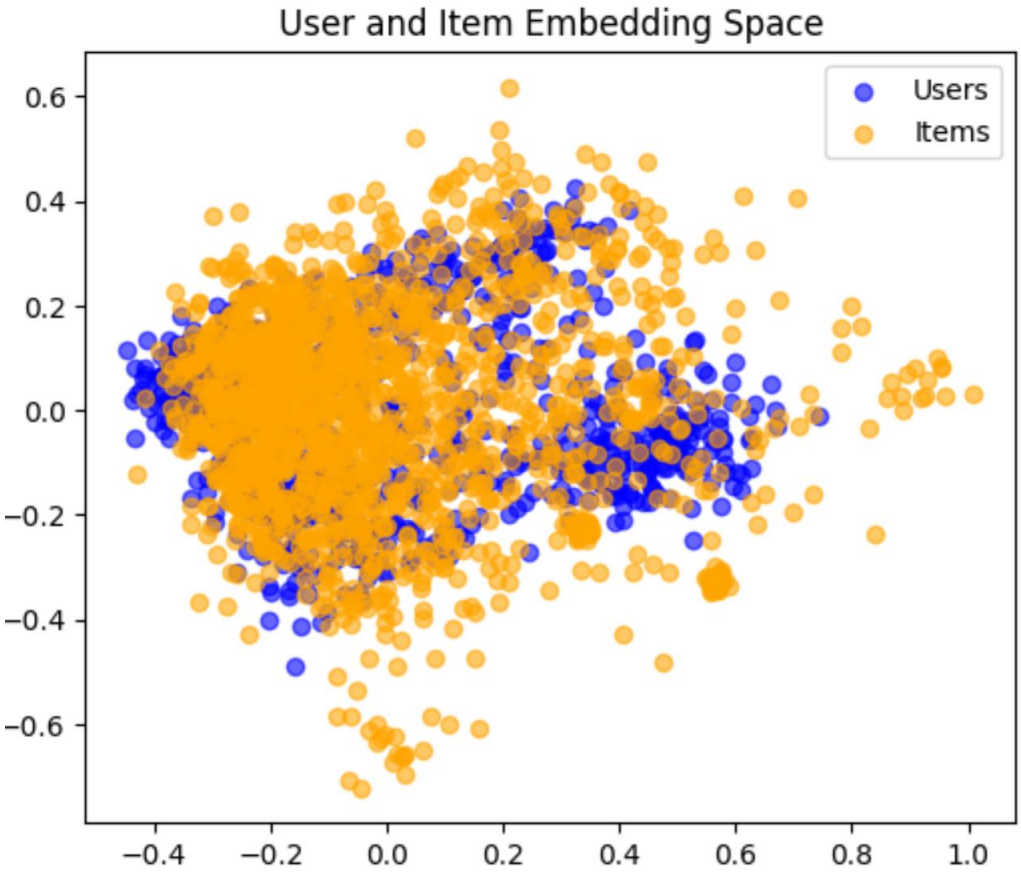
Metrics

Evaluated using:

- **RMSE (Root Mean Squared Error)**
- **MAE (Mean Absolute Error)**

Observations

- The model captured general rating patterns well.
- RMSE stabilized around 0.87–0.90, showing good generalization.
- Overfitting was minimal with weight decay (1e-5).



3D Model: Tensor Factorization (CP Decomposition)

Concept

Tensor Factorization adds a **third dimension (genre)** to the model:

This approach models **how user preferences vary across genres**, unlike MF which treats all movies uniformly.

$$R_{u,i,g} \approx \sum_{k=1}^{r} P_{u,k} \cdot Q_{i,k} \cdot G_{g,k}$$

Where:

- P, Q, G: embeddings for users, items, and genres
- r: tensor rank (latent dimensions), set to 16

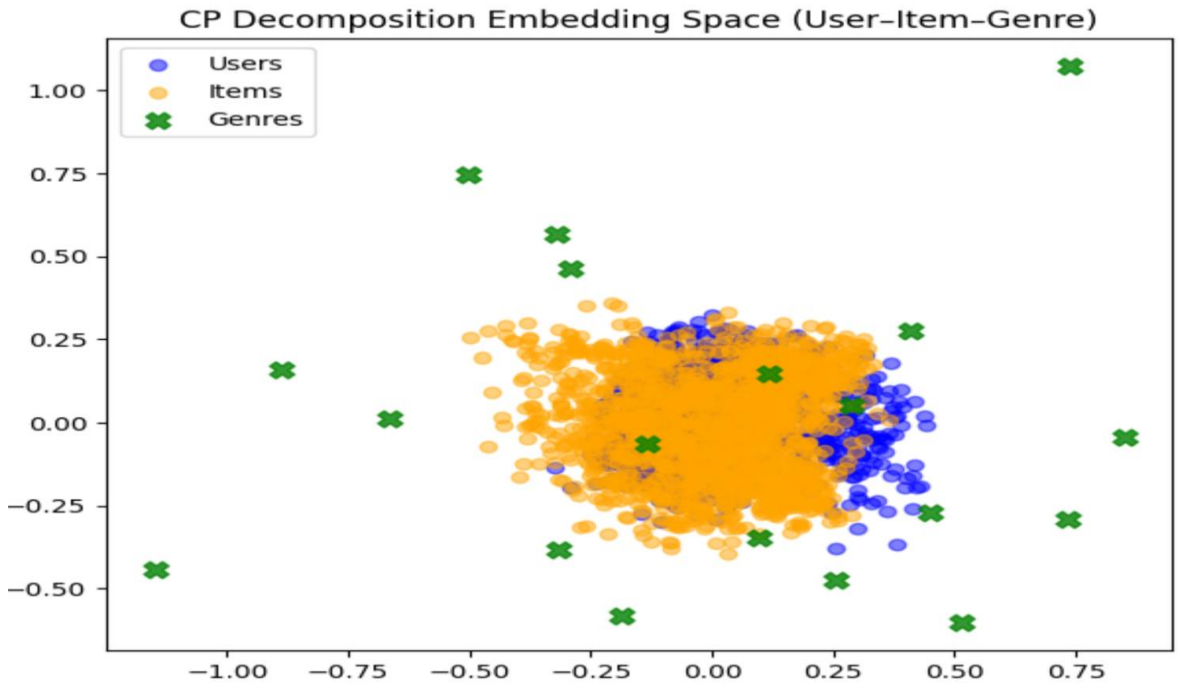
This approach models **how user preferences vary across genres**, unlike MF which treats all movies uniformly.

Implementation

- Three embedding layers: user_factors, item_factors, genre_factors
- Xavier initialization for balanced gradients
- Optimized using Adam (lr=0.01, weight_decay=1e-5)
- Trained for 15 epochs

Results

- The model learned richer user–item–genre interactions.
- RMSE and MAE slightly improved compared to MF (RMSE = 0.85).
- Genre-aware learning helped reduce bias in predictions for less-rated genres.



Findings:

- Tensor factorization consistently outperformed standard MF in both RMSE and MAE.
- CP Decomposition modeled user–genre preferences, giving more nuanced recommendations.
- Even with a smaller rank (16 vs 32), the 3D model performed better — showing that context adds more value than just increasing vector size.

Key Takeaways

- **Matrix Factorization** is strong for collaborative filtering but limited to 2D (user–item) relationships.
- **Tensor Factorization (CP)** captures contextual dependencies like genres, giving more realistic recommendations.
- The MovieLens 100K dataset remains an excellent testbed for comparing classical and modern factorization models.
- Good initialization (Xavier), regularization (weight decay), and small rank values stabilize learning in small datasets.

References

1. Koren, Y., Bell, R., & Volinsky, C. (2009). *Matrix Factorization Techniques for Recommender Systems.* IEEE Computer.
2. Kolda, T. G., & Bader, B. W. (2009). *Tensor Decompositions and Applications.* SIAM Review, 51(3), 455–500.

3. Karatzoglou, A., Smola, A., & Tamayo, P. (2010). *Multiverse Recommendation: N-dimensional Tensor Factorization for Context-Aware Collaborative Filtering.* Proceedings of the IEEE.
4. Harper, F. M., & Konstan, J. A. (2016). *The MovieLens Datasets: History and Context.* ACM Transactions on Interactive Intelligent Systems (TiiS).