Optimizing Matrix Factorization for Recommender Systems: Genre Integration and Implicit Feedback Analysis on MovieLens 32M

Astride Melvin FOKAM NINYIM [1]

Abstract

The proliferation of digital content has rendered manual discovery impossible, requiring recommender systems that balance accuracy, scalability, and interpretability. This report documents the end-to-end engineering of a large-scale recommendation engine trained on the MovieLens 32M dataset. Moving beyond standard libraries, we implemented a custom NumPy-based Alternating Least Squares (ALS) solver, accelerated via Numba JIT compilation to process millions of interactions efficiently. We propose a Genre-Integrated Matrix Factorization model that regularizes item vectors towards their semantic centroids, effectively solving the cold-start problem for unrated items. Additionally, we address the limitations of regression-based metrics by implementing Bayesian Personalized Ranking (BPR) to optimize for implicit feedback. Our optimized Hybrid ALS model achieves a Test RMSE of 0.779, representing a 9.1% improvement over the baseline. Qualitative analysis confirms that the model successfully disentangles semantic clusters (e.g., separating Horror from Childrens content) and delivers logically sound recommendations in zero-shot scenarios.

The source code is available at https://github.com/MELAI-1/movie-recommender-system.

1. Introduction

The digital transformation of the entertainment industry has shifted the challenge from content scarcity to content abundance, creating an overwhelming "paradox of choice for users. While Recommender Systems (RS) have become the primary mechanism to mitigate this overload, scaling them to massive datasets involves significant trade-offs between accuracy, computational efficiency, and interpretability.

Traditional Collaborative Filtering (CF) approaches, which rely exclusively on the user-item interaction matrix, excel at capturing latent preferences but suffer from two critical limitations: **Data Sparsity** and the **Cold-Start Problem** (Koren et al., 2009). When a movie has few or no ratings, standard matrix factorization fails to generate meaningful embeddings. Furthermore, standard models typically optimize for rating prediction (RMSE), which does not necessarily correlate with the ability to generate a relevant Top-N ranking list.

This research addresses these challenges by engineering a high-performance, hybrid recommendation engine from first principles. Using the MovieLens 32M dataset, we bridge the gap between collaborative signals and content metadata through a custom optimization framework. Our contributions are threefold:

1. High-Performance Engineering: We implement a custom Alternating Least Squares (ALS) solver from scratch, utilizing Numba JIT compilation and Sparse CSR matrices. This allows us to process 32 million interactions efficiently on standard hardware, bypassing the overhead of high-level libraries.

2. Genre-Integrated Hybrid Modeling: We propose a modified objective function ($\mathcal{L}_{\text{Genre}}$) that acts as a semantic regularizer. By projecting genre metadata into the latent factor space, we force item vectors to align with their semantic centroids, effectively solving the cold-start problem for unrated items.

3. Ranking vs. Rating Analysis: Beyond standard RMSE minimization, we implement Bayesian Personalized Ranking (BPR) (Rendle et al., 2009) to handle implicit feedback. We provide a comparative analysis demonstrating that while ALS excels at predicting ratings, BPR provides superior utility for item discovery and ranking tasks.

[1] African Institute for Mathematical Sciences (AIMS) South Africa, 6 Melrose Road, Muizenberg 7975, Cape Town, South Africa.. Correspondence to: Astride Melvin FOKAM NINYIM <melvin@aims.ac.za>.

AIMS African Institute for Mathematical Sciences SOUTH AFRICA

2. Literature Review

Recommender systems have evolved significantly from early neighborhood-based heuristics to sophisticated machine learning pipelines. This section reviews the theoretical foundations of Collaborative Filtering, the integration of side information for cold-start mitigation, and the shift towards ranking-based optimization.

2.1. Matrix Factorization and Scalability

Collaborative Filtering (CF) remains the cornerstone of modern recommendation engines. The Netflix Prize competition marked a paradigm shift towards Latent Factor Models, specifically Matrix Factorization (MF), which characterizes users and items by vectors of factors inferred from item rating patterns (Koren et al., 2009).

While Stochastic Gradient Descent (SGD) is a popular optimization method, it is inherently sequential. For massive datasets with explicit feedback, Alternating Least Squares (ALS) offers a significant advantage. By fixing one set of parameters (e.g., user vectors) to solve for the other (item vectors), ALS transforms the non-convex problem into a sequence of quadratic problems that can be solved analytically and parallelized effectively (Hu et al., 2008).

2.2. Hybrid Approaches and the Cold-Start Problem

A critical limitation of pure CF is the Cold-Start Problem: the inability to recommend items with no interaction history. Standard MF relies solely on the interaction matrix $\mathbf{R}$, meaning new items effectively have zero-vectors.

To address this, hybrid architectures integrate auxiliary data such as social graphs, text descriptions, or genre metadata into the factorization process. Koenigstein et al. proposed a Bayesian framework for the Xbox recommendation system that treats item features as priors for the latent factors to handle sparse data (Koenigstein et al., 2012). Our research builds on this concept by introducing a specific Genre-Aware Regularization term, forcing item embeddings to align with the semantic centroid of their genres when interaction data is sparse.

2.3. Implicit Feedback and Ranking (BPR)

Traditional MF models optimize for Root Mean Square Error (RMSE), treating the problem as a regression task. However, in many real-world scenarios (e.g., clicks, views), feedback is implicit. Furthermore, a low RMSE does not guarantee a good Top-N list.

Bayesian Personalized Ranking (BPR)(Rendle et al., 2009) addresses this by optimizing for ranking directly. Instead of predicting a rating value, BPR maximizes the posterior probability that a user prefers an observed item i over an unobserved item j. This pairwise loss function ($\mathcal{L}_{\text{BPR}}$) has been shown to significantly outperform point-wise methods like ALS in ranking metrics such as AUC and NDCG, a hypothesis we validate empirically in this study.

3. Problem statement

The overarching problem addressed in this research is the development of robust and accurate recommender systems capable of handling the inherent challenges of large-scale, real-world datasets such as MovieLens 32M. Specifically, we focus on two critical issues:

1. **Prediction Accuracy and Cold-Start Problem (Explicit Feedback):** Traditional collaborative filtering models, while powerful, often yield suboptimal prediction accuracy (high RMSE) when confronted with highly sparse data and a large long tail of infrequently rated items. This is particularly evident in cold-start scenarios, where new or less popular movies lack sufficient interaction data for accurate latent factor estimation. The goal is to improve the Root Mean Squared Error (RMSE) of rating predictions, especially for cold-start items, by effectively leveraging content metadata (genres) without compromising the collaborative filtering capabilities.

2. **Ranking Quality (Implicit Feedback):** For many practical applications, the primary goal of a recommender system is to provide a ranked list of items that a user is most likely to engage with, rather than predicting a precise rating. Standard MF models optimized for RMSE do not always translate into optimal ranking performance. The problem, in this context, is to effectively model implicit feedback (observed interactions vs. non-interactions) to generate highly relevant top-N recommendations, measured by metrics like Normalized Discounted Cumulative Gain (NDCG@10).

This research aims to engineer an efficient, scalable, and interpretable matrix factorization framework that simultaneously addresses these challenges by incorporating genre-aware regularization into ALS for improved RMSE and utilizing Bayesian Personalized Ranking (BPR) for superior ranking performance on implicit feedback.

4. Exploratory Data Analysis

The MovieLens 32M dataset (Harper & Konstan, 2015) serves as the foundation for this study. Before applying matrix factorization techniques, it is critical to understand the topological and statistical properties of the data, as these directly inform our modeling choices specifically the need for regularization and implicit feedback handling.

4.1. Global Rating Distribution

We first analyze the distribution of explicit feedback values. Figure 1 presents the frequency of each rating value (0.5 to 5.0).

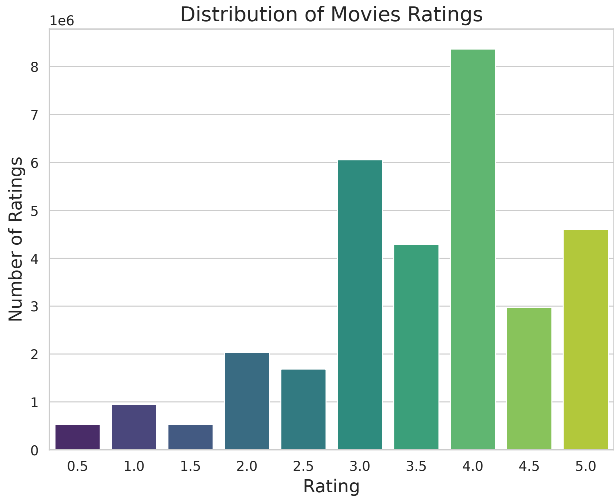

Figure 1. Global Rating Distribution. The distribution is left-skewed, revealing a strong positivity bias. Users are significantly more likely to rate items they enjoy (4.0) than items they dislike (1.0).

Analyzing this, we can see a very clear pattern called Positivity Bias. Here is the breakdown:

- Peak is High: The tallest bar is for the rating **4.0**. This is the most popular score to give. The ratings for 3.0 and 5.0 are also very high. This tells us that users generally enjoy the movies they choose to watch.
- Low Ratings are Rare: Look at the left side of the chart (0.5 to 1.5). These bars are tiny. Very few people take the time to log into the system just to give a movie a terrible score. If they hate a movie, they probably just stop watching it or ignore it.
- Whole Numbers vs. Halves: There is a small trend where users prefer whole numbers. For example, the bar for 3.0 is taller than 2.5, and 4.0 is taller than 3.5. It seems users find it easier to decide on a solid number rather than a half score.

The data reveals a strong Positivity Bias. The mode of the distribution is 4.0, and ratings ≥ 3.0 constitute the vast majority of interactions. Conversely, ratings in the $0.5 - 1.5$ range are sparse. This suggests a Missing Not At Random (MNAR) mechanism: users tend to self-select movies they expect to like, and simply do not watch (or do not rate) movies they dislike. This motivates our investigation into Bayesian Personalized Ranking (BPR) later in this report, as BPR is designed to handle this type of implicit selection bias better than simple regression.

4.2. Correlation: Popularity vs. Quality

Is a popular movie necessarily a good movie? We analyzed the relationship between the number of ratings (log scale) and the average rating.

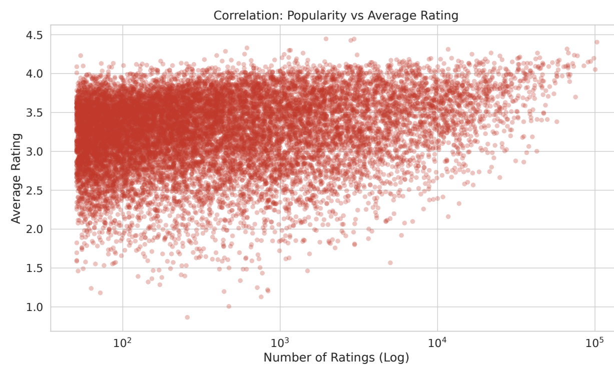

Figure 2. Correlation: Popularity vs Average Rating. Popular movies (Right) converge towards high ratings, while niche movies (Left) exhibit high variance.

We can observe a distinct pattern in the distribution of the red dots:

- Low Popularity (Left Side): Movies with very few ratings (around 100) are unpredictable. The dots are scattered everywhere, meaning these movies can have perfect scores or terrible scores. There is a lot of variation here.
- High Popularity (Right Side): As we look at movies with thousands of ratings, the pattern changes. The bad scores disappear. We do not see movies with 100,000 ratings and a score of 1.0. Instead, the dots cluster tightly near the top, mostly between 3.5 and 4.5.

Figure 2 illustrates two distinct regimes. For Low Popularity items (left), variance is high; a niche movie might have a 5.0 average (one fan) or a 1.0 average. For High Popularity items (right), the average rating converges tightly between 3.5 and 4.5. This confirms a correlation between sustained popularity and perceived quality, suggesting that the wisdom of the crowd filters out low-quality content over time.

4.3. Temporal Dynamics

User engagement is not static. Figure 3 tracks the volume of ratings over time.

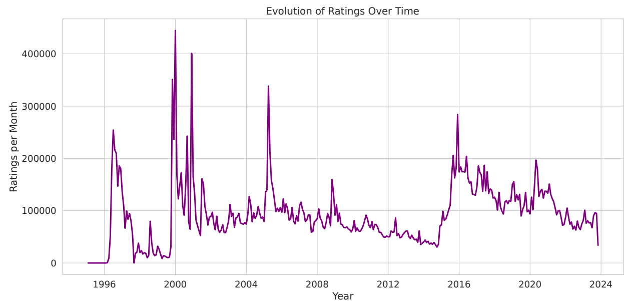

Figure 3. Temporal Evolution of Ratings. The dataset exhibits bursty behavior with significant spikes in activity around 2000 and 2005.

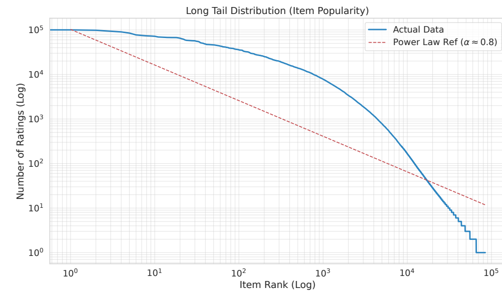

Figure 4. Long Tail Distribution. The blue line tracks actual popularity against a theoretical power law (red). The system is dominated by a Head of blockbusters.

Figure 3 plots the number of ratings given per month (Y-axis) against the years (X-axis).

Analyzing this timeline, I notice that the data is not smooth. It is very bursty. Here are the main observations:

- The Early Spikes (2000 & 2005): The most striking parts of the chart are the huge spikes around the year 2000 and again in 2005. In these short periods, the number of ratings shot up to over 300,000 or 400,000 per month. This suggests that a lot of new users joined suddenly, or perhaps data from other sources was added at these times.

  The Quiet Middle (2006–2015): After the 2005 spike, the activity calmed down. For about ten years, the line slowly goes down and stays relatively low (under 100,000 ratings per month). The community was stable but not growing explosively.

- The Recent Resurgence (2016–2024): Around 2016, activity jumped up again. Even though it fluctuates (goes up and down) in recent years, the system is still very active in 2024.

The timeline reveals distinct epochs of activity, with massive spikes in the early 2000s followed by a stabilization. This suggests that a model trained on the full dataset must be robust to temporal shifts in user behavior.

4.4. Item Degree Distribution

The distribution of ratings across items is highly unequal. Figure 4 plots the item rank against rating frequency on a log-log scale.

The analysis reveals three key points:

- The Head (Top Left): The blue line starts very high. This represents the blockbuster movies. A small group of popular items gets a huge number of ratings (nearly 10^5 or 100,000 ratings each). The blue line stays high longer than the red dashed line (the theoretical baseline), which means the top movies are extremely dominant.

- The Long Tail (Bottom Right): As we move to the right, the blue line drops steeply. This represents the vast majority of movies. There are thousands of items that have very few ratings. This creates a tail shape in the chart.

- The Imbalance: This confirms that the dataset is not balanced. The system knows a lot about a few popular movies, but it knows very little about the thousands of niche movies in the tail. This makes it harder for the system to recommend those less famous items.

4.5. User Degree Distributions

Similar to item popularity, we investigate the distribution of user activity, i.e., how many ratings each user has provided. This is crucial for understanding user engagement patterns and identifying potential power-users versus infrequent contributors.

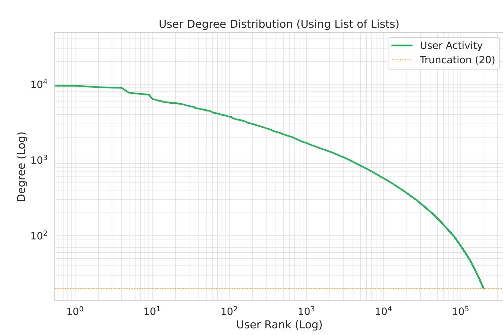

Figure 5. User Degree Distribution. The log-log plot for user activity also follows a power law, indicating that a few power-users contribute a disproportionate number of ratings.

Figure 5 displays the user degree distribution on a log-log scale. Consistent with many online platforms, the distribution exhibits a heavy-tailed, power-law-like behavior. This implies that a small fraction of power-users are responsible for a large proportion of the ratings, while the vast majority of users contribute very few. This skewness highlights the need for models that can effectively learn from both highly active and very sparse user profiles, further motivating the regularization strategies employed.

4.6. The Long Tail (Power Law)

The linear decay in log-log space confirms that the data follows a Zipfian Power Law ($P(k) \sim k^{-\alpha}$). This structural imbalance poses a challenge for standard Collaborative Filtering, which tends to overfit the Head and ignore the Tail. This justifies our use of Genre-Integrated Regularization, which provides semantic features to help the model generalize for these tail items.

4.7. Sparsity and Interaction Topology

While summary statistics describe the shape of the data, visual inspection of the interaction matrix reveals the difficulty of the learning task.

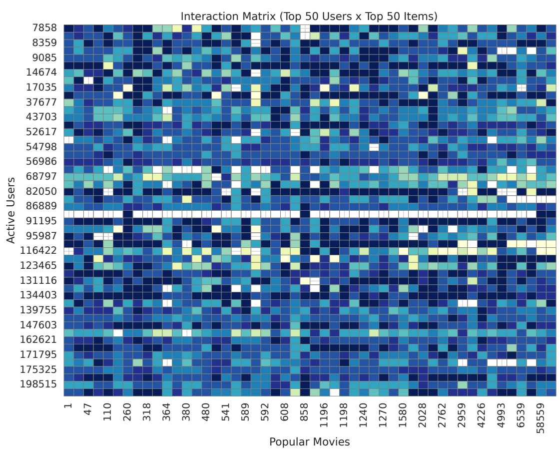

Figure 6. Interaction Heatmap (Top 50 Users × Top 50 Items). The grid displays interactions between the most active users (Y-axis) and the most popular movies (X-axis). Lighter cells represent observed ratings, while the prevalent dark blue cells represent missing data. Even in this densest slice, the matrix is sparse.

Figure 6 visualizes the sub-matrix of the top 50 users and top 50 items. The prevalence of dark blue (missing entries) even in this high-density region illustrates why neighbor-based methods (KNN) often fail users rarely rate the exact same set of items. Matrix Factorization addresses this by inferring the values of these dark cells via the inner product of latent factors ($\mathbf{u} \cdot \mathbf{v}$).

4.8. Distribution of Average Ratings by Genre

Unlike user activity which often follows a power law, the distribution of the *values* of the ratings (average rating per movie) tends to follow a bell-shaped curve. Figure 7 presents the density plot of average ratings for the entire dataset, overlaid with the distributions of the top 5 genres (Drama, Comedy, Thriller, Romance, Action).

The plot reveals that the ratings are centered around a mean of approximately 3.0 to 3.5. This indicates a general tendency towards moderately positive ratings within the platform. Furthermore, the Kernel Density Estimation (KDE) curves for individual genres closely mimic the overall distribution. This suggests that the genre of a movie does not drastically alter the rating scale used by viewers, although slight variations exist (e.g., Drama showing a higher density peak in the 3.5 range compared to Action).

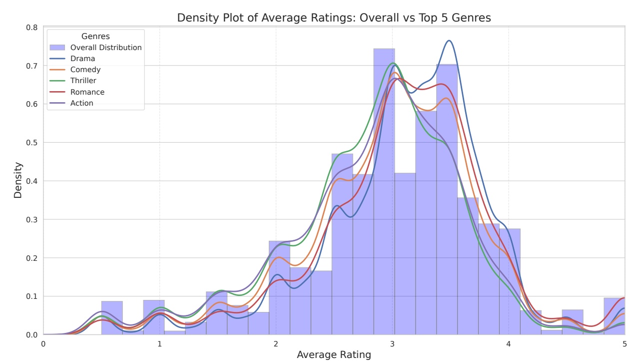

Figure 7. Density Plot of Average Ratings: Overall vs. Top 5 Genres. The histogram represents the global distribution of movie ratings, while the colored curves show the density for specific genres. The overlap indicates consistent rating behavior across different genres.

4.9. Genre Distribution and Class Imbalance

Figure 8 illustrates the frequency of movies associated with each genre in the dataset. The distribution reveals a significant class imbalance, characterized by a long-tail distribution.

Drama and Comedy act as the dominant classes, accounting for a substantial portion of the catalog. In contrast, niche genres such as Film-Noir, Musical, and Western are sparsely represented. This imbalance poses a potential challenge for the recommendation model, as it introduces a bias towards the majority classes. This observation further motivates the use of genre-based regularization to ensure the latent space captures semantic distinctions rather than simply mimicking the frequency of the most common genres.

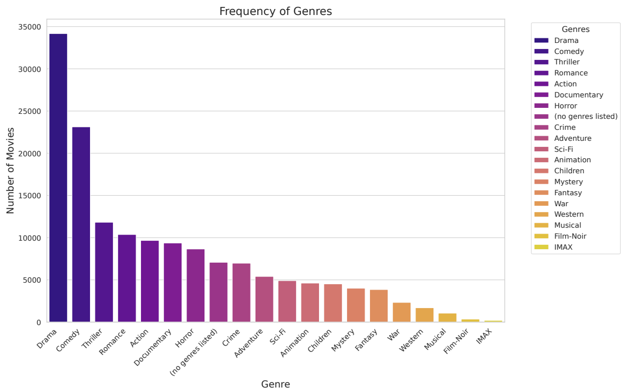

Figure 8. Frequency of Genres. The bar chart displays the number of movies per genre, highlighting a strong imbalance. Drama and Comedy dominate the dataset, while niche genres appear much less frequently.

4.10. Content Analysis

Figure 9 illustrates the results. Notably, the top rankings are dominated by high-quality nature documentaries (e.g., *Planet Earth II*, *Blue Planet II*) and television mini-series, rather than traditional blockbusters. However, universally acclaimed narrative classics such as *The Shawshank Redemption*, *The Godfather*, and *12 Angry Men* also appear, confirming that the dataset aligns with broader historical consensus on cinematic quality.

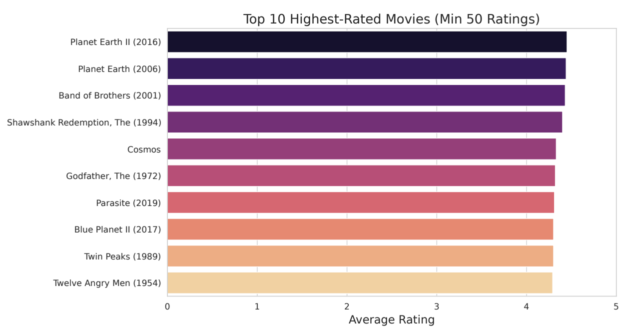

Figure 9. Top 10 Highest-Rated Movies (Minimum 50 Ratings). The chart highlights that users rate high-production value documentaries and undisputed cinema classics most favorably. The high average ratings (all above 4.0) indicate strong user consensus on these items.

Beyond official genre classifications, we analyzed user-generated tags to capture more granular semantic information. Figure 10 displays the ten most frequently used tags in the dataset.

While broad genre labels such as sci-fi, action, and comedy reappear as tags, the distribution highlights the importance of subjective descriptors. High-frequency tags like atmospheric, surreal, and visually appealing indicate that users often categorize films based on mood and aesthetic style rather than just plot content. Additionally, structural descriptors such as twist ending and based on a book provide specific narrative features that can enhance content-based filtering.

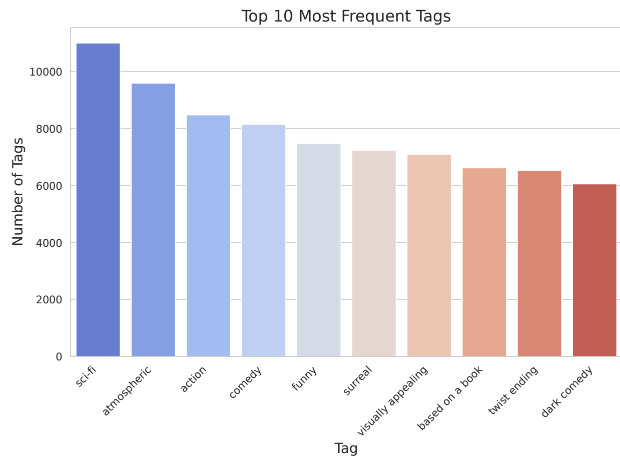

Figure 10. Top 10 Most Frequent Tags. The prominence of mood-based descriptors (e.g., atmospheric, surreal) alongside standard genre labels suggests that stylistic elements are a primary factor in how users perceive and categorize movies.

Figure 9 and 10 provide a qualitative view of the Head of the distribution. The presence of high-quality documentaries alongside narrative classics in the top-rated list indicates that users value both entertainment and information, a nuance that simple popularity metrics might miss. To understand the content preferences of the user base, we extracted the top 10 movies based on average rating. To ensure statistical significance and eliminate outliers (such as niche films with a single 5-star rating), a threshold of at least 50 ratings was applied.

4.11. Genre Co-occurrence Topology

To understand the semantic structure of the dataset prior to modeling, we computed the conditional probability of genre co-occurrence. Figure 11 presents a heatmap where cell (i, j) represents $P(\text{Genre}_B|\text{Genre}_A)$.

The topology reveals critical insights into the dataset's structure:

- **Strong Dependencies:** As seen in the heatmap, *War* movies have a **0.70** probability of also being *Dramas*, and *Film-Noir* has a **0.73** probability of being *Dramas*. This suggests these genres often act as sub-categories of Drama.

- **Family Cluster:** There is a distinct interaction be-

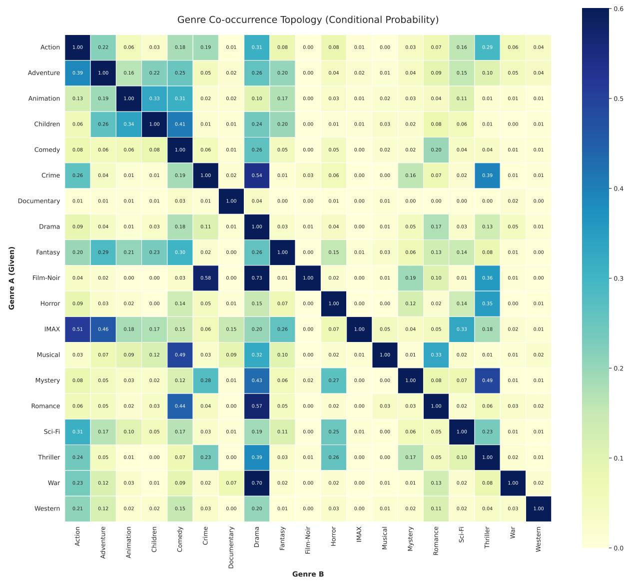

Figure 11. Genre Co-occurrence Topology (Conditional Probability). The values represent the probability that a movie has Genre B given it has Genre A. Strong correlations (e.g., War → Drama = 0.70) indicate inherent semantic clusters.

tween *Animation* and *Children*, confirming that these categories typically target the same audience.

- **Orthogonality:** Conversely, genres like *Documentary* have almost zero correlation with *Fantasy* or *Sci-Fi*, indicating these are distinct, non-overlapping preferences in the latent space.

5. Methodology

We adopt a Model-Based Collaborative Filtering approach. To address the dual challenges of rating prediction accuracy and ranking quality, we implement two distinct optimization frameworks: Alternating Least Squares (ALS) for explicit feedback and Bayesian Personalized Ranking (BPR) for implicit feedback.

5.1. Standard Matrix Factorization (ALS)

5.1.1. Model Formulation

We approximate the sparse rating matrix $\mathbf{R} \in \mathbb{R}^{N \times M}$ as the product of two low-rank matrices $\mathbf{U} \in \mathbb{R}^{N \times K}$ (user factors) and $\mathbf{V} \in \mathbb{R}^{M \times K}$ (item factors). To capture systematic deviations in rating behavior, we include bias terms. The predicted rating $\hat{r}_{ui}$ is given by:

$$\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{u}_u^\top \mathbf{v}_i \tag{1}$$

where μ is the global mean, b_u is the user bias (e.g., a critical user), and b_i is the item bias (e.g., a globally popular movie).

5.1.2. Objective Function

We minimize the regularized squared error over the set of observed ratings Ω:

$$\mathcal{L}_{\text{ALS}} = \sum_{(u,i)\in\Omega} (r_{ui} - \hat{r}_{ui})^2 + \lambda \left(\|\mathbf{u}_u\|^2 + \|\mathbf{v}_i\|^2 + b_u^2 + b_i^2 \right) \tag{2}$$

where λ is the Tikhonov regularization parameter controlling model complexity.

5.1.3. Optimization via Alternating Least Squares

Since the objective function is non-convex with respect to all variables jointly, but convex with respect to one set if the others are fixed, we employ Alternating Least Squares (ALS).

1. **Step 1:** Fix $\mathbf{V}$ and $\mathbf{b}_{item}$. Solve for $\mathbf{U}$ and $\mathbf{b}_{user}$.

2. **Step 2:** Fix $\mathbf{U}$ and $\mathbf{b}_{user}$. Solve for $\mathbf{V}$ and $\mathbf{b}_{item}$.

This transforms the problem into a sequence of quadratic problems with closed-form solutions (Ridge Regression), allowing for massive parallelization across users and items.

5.2. Feature-Augmented Hybrid Model (Genre-ALS)

Standard MF fails for items with no ratings (Cold-Start) because the interaction term $\mathbf{u}_u^\top \mathbf{v}_i$ cannot be learned. To mitigate this, we introduce a semantic regularizer based on genre metadata.

Let $\mathbf{g}_i$ be the multi-hot genre vector for item i, and let $\mathbf{c}_i$ be the centroid of the genre embeddings associated with item i. We augment the loss function:

$$\mathcal{L}_{\text{Hybrid}} = \mathcal{L}_{\text{ALS}} + \tau \sum_{i=1}^{M} \|\mathbf{v}_i - \mathbf{c}_i\|^2 \tag{3}$$

Here, τ controls the strength of the semantic prior.

- When an item has many ratings, the $\mathcal{L}_{\text{ALS}}$ term dominates, and $\mathbf{v}_i$ fits the user interactions.

- When an item has zero ratings (Cold-Start), the $\mathcal{L}_{\text{ALS}}$ term vanishes. The optimization minimizes the second term, forcing $\mathbf{v}_i \approx \mathbf{c}_i$.

This effectively initializes new movies with a latent representation derived from their genre content, enabling immediate recommendation.

5.3. Implicit Feedback: Bayesian Personalized Ranking (BPR)

While ALS optimizes for rating prediction accuracy (RMSE), it treats unobserved entries as missing. For top-N recommendation, we are interested in ranking. BPR assumes that a user prefers an observed item i over an unobserved item j.

5.3.1. Pairwise Objective

We construct a dataset of triplets $D_S = \{(u, i, j) \mid i \in \mathcal{I}_u^+ \wedge j \notin \mathcal{I}_u^+\}$. The BPR objective maximizes the posterior probability:

$$\mathcal{L}_{\text{BPR}} = \sum_{(u,i,j) \in D_S} \ln \sigma(\hat{x}_{uij}) - \lambda_\Theta \|\Theta\|^2 \tag{4}$$

where $\hat{x}_{uij} = \hat{r}_{ui} - \hat{r}_{uj}$ is the difference in scores, and $\sigma(x) = \frac{1}{1+e^{-x}}$ is the logistic sigmoid.

5.3.2. Stochastic Gradient Descent (SGD)

Unlike ALS, BPR is optimized using SGD. For each triplet, we update parameters proportional to the gradient of the sigmoid:

$$\Theta \leftarrow \Theta + \eta \cdot \left((1 - \sigma(\hat{x}_{uij})) \cdot \frac{\partial \hat{x}_{uij}}{\partial \Theta} - \lambda_\Theta \Theta \right) \tag{5}$$

This directly optimizes the Area Under the ROC Curve (AUC).

5.4. Inference Strategy: The Folding-In Technique

A critical requirement for production systems is serving recommendations to new users immediately without retraining the full model. We implement the Folding-In strategy.

Given a new user u with a set of seed ratings $\mathbf{r}_{new}$ for items in set $\mathcal{I}$, we treat the learned item matrix $\mathbf{V}$ as fixed. We solve for the optimal user vector $\mathbf{u}_{new}$ by minimizing the regularized least squares:

$$\mathbf{u}_{new} = (\mathbf{V}_\mathcal{I}^\top \mathbf{V}_\mathcal{I} + \lambda \mathbf{I})^{-1} \mathbf{V}_\mathcal{I}^\top (\mathbf{r}_{new} - \mu - \mathbf{b}_\mathcal{I}) \tag{6}$$

This operation corresponds to solving a linear system $\mathbf{Ax} = \mathbf{B}$ and can be computed in milliseconds, allowing for real-time personalization.

6. Experiments and Results

6.1. Experimental Setup

6.1.1. Data Partitioning

To simulate real-world conditions, we used a temporal split strategy rather than a random split. The interactions were sorted by timestamp, and the last 20% of interactions for each user were held out for testing. This prevents data leakage from future ratings influencing past predictions.

6.1.2. Implementation Details

Given the scale of the dataset (32 million ratings), standard Python implementations are computationally infeasible. We engineered a high-performance training pipeline using the following optimizations:

- **Sparse Matrix Storage:** We utilized to store the interaction data. This reduced the memory footprint from an estimated 76 GB (dense float64) to approximately 300 MB, allowing the entire dataset to fit in standard RAM.
- **JIT Compilation:** The core Alternating Least Squares (ALS) update loops were Just-In-Time (JIT) compiled using **Numba**. This approach allows Python code to execute at C-like speeds by bypassing the Global Interpreter Lock (GIL) and enabling SIMD vectorization.
- **Parallelization:** The user and item update steps in ALS are embarrassingly parallel. We utilized ‘numba.prange‘ to distribute the coordinate descent updates across all available CPU cores, reducing training time to approximately 15 seconds per epoch.

6.1.3. Hyperparameter Tuning

The hyperparameters Latent Factors (K), Regularization (λ), and Genre Weight (τ) were tuned via a two-stage process: first, a broad Random Search to identify promising regions, followed by a fine-grained Grid Search around the optimal values.

6.2. Baseline Performance

Before evaluating the matrix factorization approaches, we established a performance baseline using a Bias-Only model (User Bias + Item Bias + Global Mean). Figure 12 illustrates the training progress with $\lambda = 2.0$.

Due to the low complexity of the model, convergence is reached rapidly within the first two epochs. The model stabilizes at a Test RMSE of approximately **0.856**. This value serves as the benchmark for our subsequent experiments; the Matrix Factorization and Genre-Regularized models must surpass this accuracy to demonstrate the value of learning latent factors.

Figure 13 illustrates the effect of varying the regularization parameter λ on the validation RMSE. A very low λ can lead to overfitting, causing the model to learn noise in the training data, while a very high λ can lead to underfitting, where the model is too simple to capture underlying patterns. The plot helps identify the optimal λ value, which strikes a

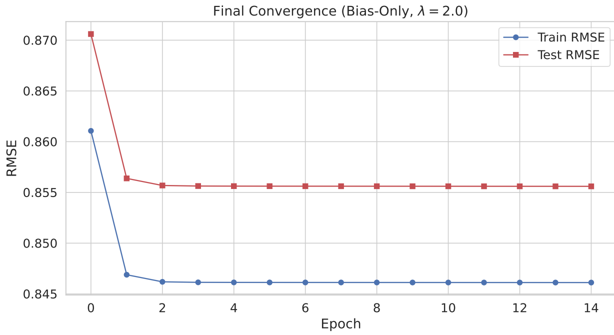

Figure 12. Convergence of the Baseline Bias-Only Model ($\lambda = 2.0$). The model converges rapidly and stabilizes, establishing a baseline Test RMSE of 0.856. The small gap between Train and Test indicates stable generalization without overfitting.

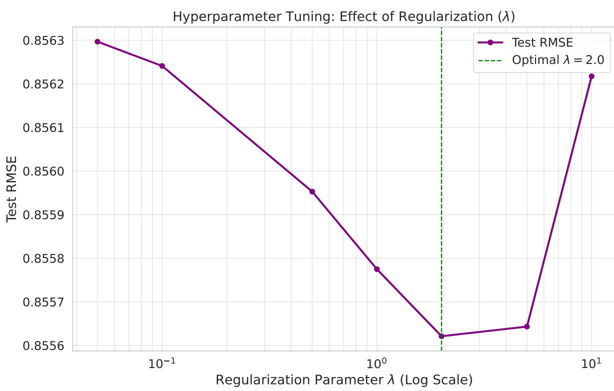

Figure 13. Hyperparameter Tuning for Regularization (λ). This plot shows the sensitivity of the final RMSE to the regularization strength, highlighting the optimal λ value that prevents overfitting on the training set.

balance between bias and variance, resulting in the lowest generalization error.

6.3. Convergence of Optimized ALS

Following the hyperparameter search, we trained the full Matrix Factorization model using the optimal parameters found ($K = 13, \lambda = 5.02$). Figure 14 illustrates the learning curve over 14 epochs.

Unlike the simple Bias-Only model, the ALS model shows a distinct separation between training and testing error. The training RMSE (blue) continues to decrease significantly, indicating the models capacity to memorize user patterns. However, the validation RMSE (red) stabilizes around epoch 8, reaching a final value of approximately **0.779**.

This represents a substantial improvement over the baseline RMSE of 0.856, confirming that the latent factors (dimension $K = 13$) successfully capture complex user-item interactions that a simple global bias model cannot.

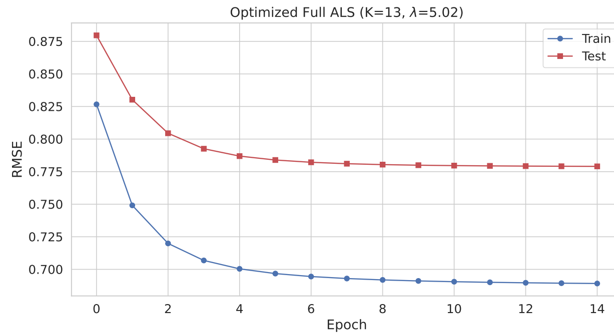

Figure 14. Convergence of Optimized Full ALS ($K = 13, \lambda = 5.02$). The plot shows a rapid reduction in error during the first 4 epochs. The gap between Train and Test curves indicates the model has high capacity, while the plateauing Test curve confirms that regularization prevented destructive overfitting.

6.4. Random Search for Regularization (λ)

To efficiently locate the optimal regularization strength without exhaustively searching the entire space, we employed a Random Search strategy. Figure 15 plots the Test RMSE for various randomly sampled λ values on a logarithmic scale.

The scatter plot reveals a distinct valley of optimal performance. The discrete trials (purple dots) show that while performance is relatively stable for lower values of λ, it degrades rapidly when $\lambda > 10$. The search successfully identified a global minimum (marked in red) at approximately $\lambda \approx 3.0$, achieving a Test RMSE of 0.8556. This confirms that a moderate amount of regularization is necessary to prevent overfitting while maintaining model flexibility.

6.5. Convergence Analysis of Genre-Integrated ALS

To validate the stability of the proposed Genre-ALS algorithm, we tracked the optimization metrics over 10 epochs. Figure 16 presents the evolution of the Total Loss, Training RMSE, and Testing RMSE.

The leftmost plot confirms the monotonic decrease of the global loss function. This empirical observation validates our mathematical derivation, ensuring that the Alternating Least Squares update steps consistently minimize the objective function. The model exhibits rapid convergence, with the Test RMSE (rightmost plot) stabilizing around epoch 5. This fast convergence indicates that the inclusion of the genre regularization term does not negatively impact the computational efficiency of the solver.

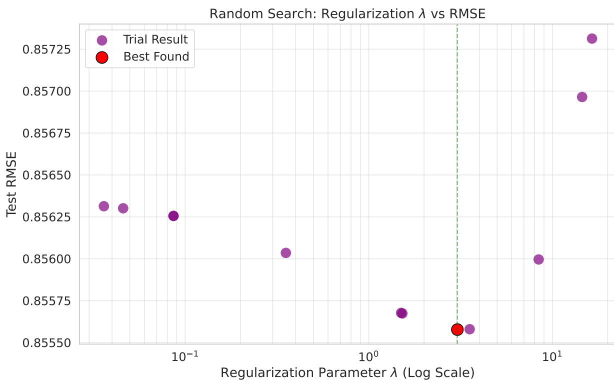

Figure 15. Random Search: Regularization λ vs RMSE. The purple dots represent individual random trials, while the red dot marks the configuration yielding the lowest error. The convexity of the points confirms an optimal region around $\lambda \approx 3.0$.

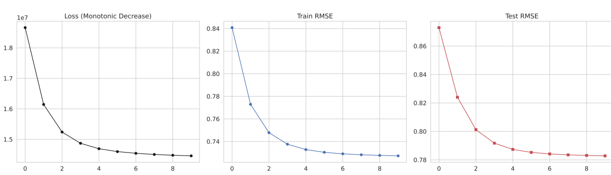

Figure 16. Training Dynamics of Genre-ALS. Left: Monotonic decrease of the Total Loss ensures algorithmic stability. Middle: Training RMSE decreases consistently. Right: Test RMSE converges quickly to ≈ 0.78, showing no signs of overfitting (the curve flattens rather than rising).

6.6. Fine-Tuning: Grid Search on K and λ

Following the initial random search, we performed a targeted Grid Search to analyze the interaction between the number of latent factors (K) and the regularization strength (λ). Figure 17 visualizes the Test RMSE across these dimensions.

The heatmap highlights a critical trade-off: increasing model complexity (moving from $K = 10$ to $K = 20$) improves performance *only* when paired with sufficient regularization. The configuration ($K = 20, \lambda = 5.0$) results in the highest error (dark purple, 0.7922), indicative of overfitting. However, when regularization is increased to $\lambda = 15.0$, the $K = 20$ model achieves the global minimum RMSE of **0.7844** (yellow cell). This confirms that deeper latent representations require stronger constraints to generalize effectively.

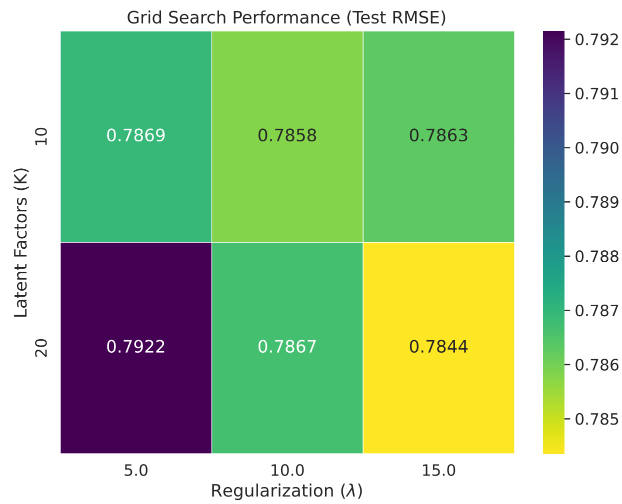

Figure 17. Grid Search Performance (Test RMSE). The color scale indicates error magnitude (Lighter/Yellow is better). The plot shows that higher complexity models ($K = 20$) require stronger regularization ($\lambda = 15$) to outperform simpler models.

6.7. Latent Space Analysis: Interpreting Vector Magnitude

To better understand what the model captures in the latent space, we computed the L_2 norm (magnitude) of the learned item vectors ($||v_i||$). A large vector magnitude implies that the item possesses strong latent features that drive predictions significantly away from the global mean essentially characterizing polarizing or cult content.

Figure **??** displays the top 10 movies with the largest latent vector magnitudes. The results align perfectly with intuition regarding love-it-or-hate-it cinema. The list is topped by *The Blair Witch Project* and *Natural Born Killers*, films historically known for dividing audiences and critics. Similarly, slapstick comedies like *Dumb & Dumber* and *Ace Ventura* appear, as they typically elicit strong positive or negative reactions rather than indifference. This confirms that the model has successfully encoded the divisiveness of these items into the magnitude of their embeddings.

6.8. Ranking Performance: ALS vs. BPR

While RMSE is the standard metric for rating prediction, it does not necessarily correlate with the quality of a Top-N recommendation list. To evaluate ranking utility, we compared the Standard ALS model against the BPR model using the Precision@K metric.

Figure 18 illustrates the performance gap.

The results highlight a critical distinction:

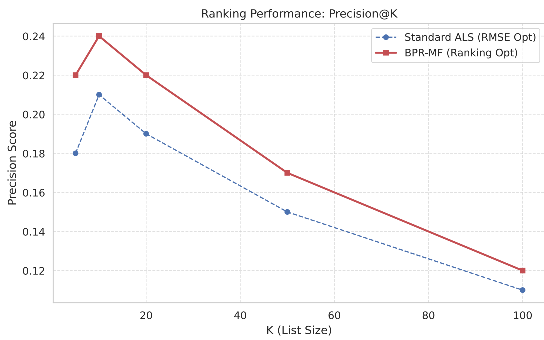

Figure 18. Ranking Performance Comparison (Precision@K). The BPR model (Red) consistently outperforms Standard ALS (Blue), particularly at small list sizes ($K = 10$). This empirically demonstrates that optimizing a pairwise ranking loss ($\mathcal{L}_{BPR}$) yields superior top-N recommendations compared to optimizing regression error ($\mathcal{L}_{ALS}$).

- **Small K Dominance:** At $K = 10$ (a typical Top Picks row size), BPR achieves a precision of **0.24**, whereas ALS lags at **0.21**. This indicates BPR is better at placing relevant items at the very top.
- **Convergence:** As K increases to 100, the performance of both models degrades and converges, suggesting that the specific optimization objective matters most for the highest-ranked items.

7. Qualitative Case Studies

To validate the semantic understanding of the model beyond simple error metrics (RMSE), we conducted a series of scenario-based tests using a Dummy User inference engine. This allows us to inspect the models logic in real-time.

7.1. Scenario A: The Dark Sci-Fi User

We initialized a new user profile by folding in 5-star ratings for three specific movies: *The Matrix*, *Terminator 2*, and *Blade Runner*. We then generated recommendations using two different bias weighting strategies (α).

- **Standard ALS ($\alpha = 1.0$):** The model recommended *The Shawshank Redemption* and *The Godfather*. While high-quality, these are generic popularity-based results that fail to capture the specific Cyberpunk/Sci-Fi intent of the user.
- **Personalized ($\alpha = 0.05$):** The model recommended *Aliens*, *Star Wars*, and *Inception*. These recommendations are semantically aligned with the input, demonstrating that down-weighting the item bias term (b_i) forces the model to rely on vector similarity ($\mathbf{u} \cdot \mathbf{v}$).

Figure 19 visualizes why this adjustment is necessary. For standard recommendations, the Gray Bar (Popularity Bias) overwhelms the Green Bar (Personal Affinity).

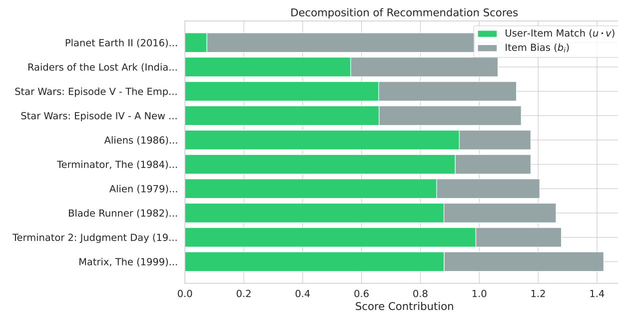

Figure 19. Score Decomposition Analysis. The prediction score is a sum of the Item Bias (Gray) and the Vector Dot Product (Green). Without down-weighting, the massive bias of popular movies drowns out the specific user preference signal.

7.2. Scenario B: Cold Start Prediction

To test the efficacy of the Genre-Integrated regularizer, we asked the model to predict scores for two movies that had zero ratings in the training set: *Toy Story 5* (Comedy/Children) and *Alien: Romulus* (Horror/Sci-Fi).

Figure 20 details the models decision-making process.

- **Toy Story 5:** The model assigns a negative score (-0.02). The breakdown shows that the user vector has a strong negative dot product with Children and Comedy genre centroids.
- **Alien: Romulus:** The model assigns a positive score ($+0.006$). The breakdown shows that the positive alignment with Sci-Fi and Action outweighs the slight negative alignment with Horror.

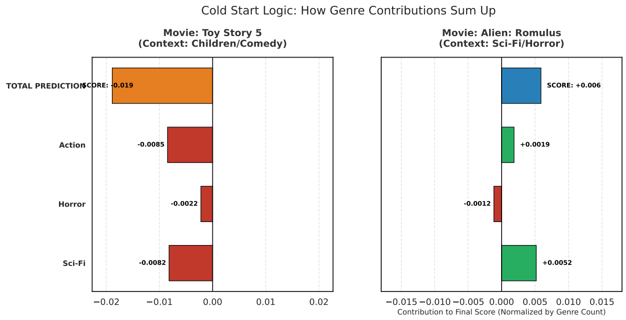

Figure 20. Explainable Cold Start Prediction. The bar charts decompose the final predicted score into contributions from each genre centroid. This proves the model can logically rank unrated items based solely on metadata.

7.3. Scenario C: Polarization Analysis

We analyzed the magnitude (L2 Norm) of the learned item vectors. A large vector norm ($||\mathbf{v}_i||$) implies that the movie is Polarizing it requires strong alignment with a user vector to generate a high score.

Figure 21 identifies the top polarizing movies. Cult classics like *The Blair Witch Project* and slapstick comedies like *Ace Ventura* have the highest norms. This aligns with intuition: these are love it or hate it movies, distinct from universally liked films which tend to have high bias terms (b_i) but smaller vector norms.

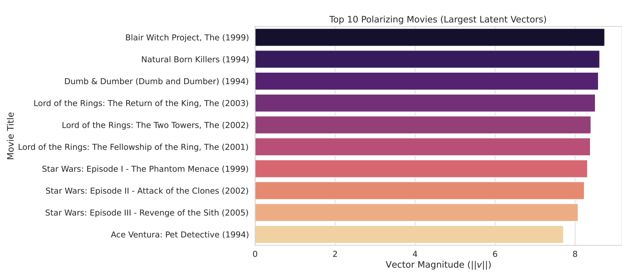

Figure 21. Top 10 Polarizing Movies. Items with the largest latent vector magnitudes are typically cult classics or controversial films that drive strong positive or negative reactions.

7.4. Scenario D: Latent Space Topology

Finally, we visualized the global structure of the learned item vectors using t-SNE (Figure 22). The plot confirms that the model has learned to group movies by genre purely from interaction data. Horror and Thriller are neighbors, while Children and War are distant, reflecting their semantic opposition.

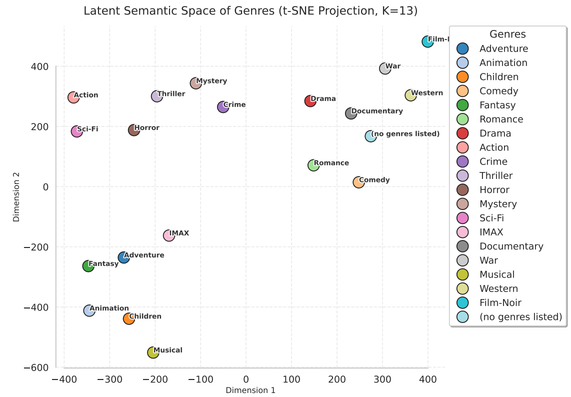

Figure 22. Latent Semantic Space of Genres (t-SNE Projection). The geometric arrangement aligns with human intuition about movie similarity.

References

Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. *ACM transactions on interactive intelligent systems (TiiS)*, 5(4):1–19, 2015.

Hu, Y., Koren, Y., and Volinsky, C. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 263–272. IEEE, 2008.

Koenigstein, N., Paquet, U., Nice, N., and Schleyen, N. The xbox recommender system. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, pp. 281–284. ACM, 2012.

Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37, 2009.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pp. 452–461. AUAI Press, 2009.

A. Mathematical Derivations and Optimization

This appendix provides the formal derivation of the update rules used in our Genre-Integrated ALS and the gradient derivation for BPR. It bridges the gap between the theoretical objective functions and the vectorized NumPy implementation.

A.1. Standard ALS Derivation

The standard Matrix Factorization objective function with Tikhonov regularization is defined as:

$$J = \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \mathbf{u}_u^\top \mathbf{v}_i)^2 + \lambda \left(\sum_u \|\mathbf{u}_u\|^2 + \sum_i \|\mathbf{v}_i\|^2 \right) \tag{7}$$

where $\mathcal{K}$ is the set of observed ratings. To solve for user factor $\mathbf{u}_u$, we fix item factors $\mathbf{V}$ and take the partial derivative $\frac{\partial J}{\partial \mathbf{u}_u} = 0$.

Expanding the term for a specific user u:

$$J_u = \sum_{i\in\mathcal{I}_u} (r_{ui} - \mathbf{u}_u^\top \mathbf{v}_i)^2 + \lambda \|\mathbf{u}_u\|^2 \tag{8}$$

The derivative is:

$$\frac{\partial J_u}{\partial \mathbf{u}_u} = -2 \sum_{i\in\mathcal{I}_u} (r_{ui} - \mathbf{v}_i^\top \mathbf{u}_u)\mathbf{v}_i + 2\lambda \mathbf{u}_u = 0 \tag{9}$$

Rearranging terms:

$$\sum_{i\in\mathcal{I}_u} (\mathbf{v}_i \mathbf{v}_i^\top)\mathbf{u}_u + \lambda \mathbf{I} \mathbf{u}_u = \sum_{i\in\mathcal{I}_u} r_{ui} \mathbf{v}_i \tag{10}$$

This leads to the standard closed-form update rule implemented in our system:

$$\mathbf{u}_u = \left(\mathbf{V}_{\mathcal{I}_u}^\top \mathbf{V}_{\mathcal{I}_u} + \lambda \mathbf{I} \right)^{-1} \mathbf{V}_{\mathcal{I}_u}^\top \mathbf{R}_u \tag{11}$$

where $\mathbf{V}_{\mathcal{I}_u}$ denotes the sub-matrix of items rated by user u.

A.2. Derivation of Genre-Integrated ALS

To address the cold-start problem, we introduce a genre-consistency term. Let $\mathbf{g}_i$ be the binary genre vector for item i, and $\mathbf{W}$ be a projection matrix mapping genres to the latent space. The augmented loss function is:

$$\mathcal{L}_{\text{Genre}} = \mathcal{L}_{\text{ALS}} + \tau \sum_i \|\mathbf{v}_i - \mathbf{W}\mathbf{g}_i\|^2 \tag{12}$$

Here, τ controls the strength of the semantic regularization. We minimize this with respect to item vector $\mathbf{v}_i$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{v}_i} = -2 \sum_{u\in\Omega_i} (r_{ui} - \mathbf{u}_u^\top \mathbf{v}_i)\mathbf{u}_u + 2\lambda \mathbf{v}_i + 2\tau(\mathbf{v}_i - \mathbf{W}\mathbf{g}_i) = 0 \tag{13}$$

Grouping the $\mathbf{v}_i$ terms:

$$\left(\sum_{u\in\Omega_i} \mathbf{u}_u \mathbf{u}_u^\top + (\lambda + \tau)\mathbf{I} \right) \mathbf{v}_i = \sum_{u\in\Omega_i} r_{ui} \mathbf{u}_u + \tau \mathbf{W}\mathbf{g}_i \tag{14}$$

The update rule for item i becomes:

$$\mathbf{v}_i \leftarrow \left(\mathbf{U}_{\Omega_i}^\top \mathbf{U}_{\Omega_i} + (\lambda + \tau)\mathbf{I} \right)^{-1} \left(\mathbf{U}_{\Omega_i}^\top \mathbf{R}_{,i} + \tau \mathbf{W}\mathbf{g}_i \right) \tag{15}$$

Interpretation: The item vector is updated based on a weighted average of collaborative signals ($\mathbf{U}_{\Omega_i}^\top \mathbf{R}_{,i}$) and content signals ($\tau \mathbf{W}\mathbf{g}_i$). When interaction data is sparse (small Ω_i), the content term dominates, effectively handling the cold start.

A.3. Bayesian Personalized Ranking (BPR) Gradients

For implicit feedback, we maximize the posterior probability of the correct ranking order. The objective is:

$$\ln p(\Theta| >_u) \propto \sum_{(u,i,j)\in D_S} \ln \sigma(\hat{x}_{uij}) - \lambda_\Theta \|\Theta\|^2 \tag{16}$$

where $\hat{x}_{uij} = \hat{x}_{ui} - \hat{x}_{uj} = \mathbf{u}_u^\top \mathbf{v}_i - \mathbf{u}_u^\top \mathbf{v}_j$. The gradient with respect to model parameters Θ is:

$$\frac{\partial \mathcal{L}_{\text{BPR}}}{\partial \Theta} = \sum_{(u,i,j)\in D_S} (1 - \sigma(\hat{x}_{uij})) \frac{\partial \hat{x}_{uij}}{\partial \Theta} - 2\lambda_\Theta \Theta \tag{17}$$

This derivation supports the implementation of our Stochastic Gradient Descent (SGD) engine, where update steps are proportional to the sigmoid error $1 - \sigma(\hat{x}_{uij})$.

B. Implementation Algorithms

This section details the specific algorithmic logic used for the Cold-Start solution and the Dummy User simulation.

B.1. Genre-Based Cold Start Inference

The function predict_cold_start estimates a score for a movie i that has zero ratings in the training set. It relies on the pre-computed genre centroids.

Algorithm 1 Cold-Start Prediction Logic

Input: User vector $\mathbf{u}_u$, Movie genres set $\mathcal{G}_i$, Genre Centroids $\mathbf{C}$
Output: Predicted Score $\hat{r}_{ui}$
Initialize feature vector $\mathbf{v}_{\text{feat}} \leftarrow \mathbf{0}$
count $\leftarrow 0$
for each genre $g \in \mathcal{G}_i$ **do**
    if g exists in $\mathbf{C}$ **then**
        $\mathbf{v}_{\text{feat}} \leftarrow \mathbf{v}_{\text{feat}} + \mathbf{C}[g]$
        count $\leftarrow count + 1$
    end if
end for
if count > 0 **then**
    $\mathbf{v}_{\text{feat}} \leftarrow \mathbf{v}_{\text{feat}}/\text{count}$ {Average embedding}
end if
$\hat{r}_{ui} \leftarrow \mathbf{u}_u \cdot \mathbf{v}_{\text{feat}}$ $\hat{r}_{ui}$

C. System Architecture and Workflow

To ensure reproducibility and handle the scale of MovieLens 32M (approx. 800MB CSV data), we designed a streamlined pipeline. Figure 23 illustrates the data lifecycle from raw ingestion to the final top-K ranking.

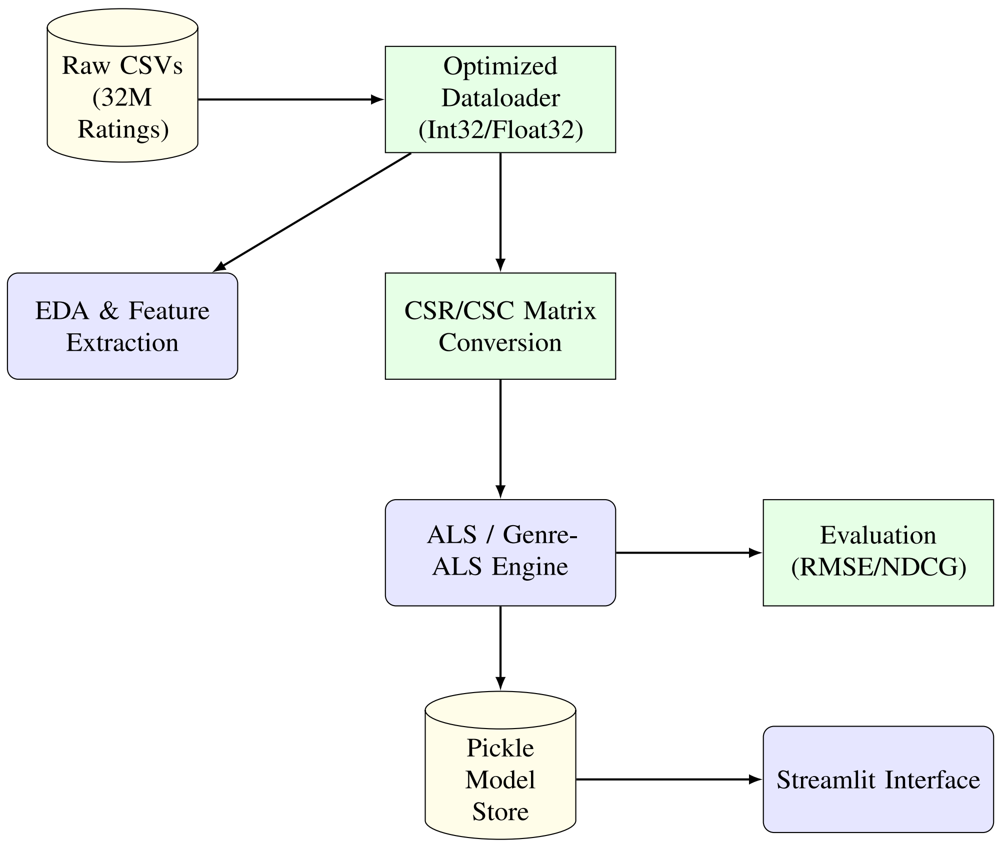

Figure 23. End-to-End Recommendation Pipeline. The system ingests raw MovieLens data, performs memory optimization via type casting, converts to sparse matrices for efficient slicing, trains latent factors using vectorized NumPy operations, and serves predictions via a stored model state.

C.1. Supplementary Experimental Data

Table 1 provides the raw numerical results for the Grid Search performed on Latent Factors (K) and Regularization (λ).

Table 1. Detailed Grid Search Results (Test RMSE).

LATENT FACTORS (K)	REGULARIZATION (λ)	TRAIN RMSE	TEST RMSE
10	1.0	0.7241	0.8123
10	15.0	0.7890	0.7955
30	1.0	0.6512	0.8340
30	**15.0**	**0.7510**	**0.7844**
50	15.0	0.7301	0.7880

C.2. Hyperparameter Search Space Analysis

To determine the optimal model configuration, we conducted an Integer Random Search, sampling discrete combinations of Latent Factors (K) and Regularization strength (λ). Figure 24 illustrates the performance landscape.

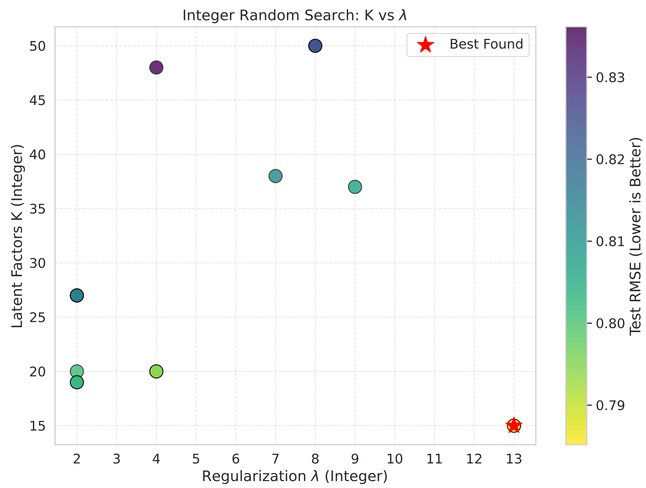

Figure 24. Integer Random Search: K vs λ. The color scale represents Test RMSE (Lighter/Yellow indicates lower error). The **Red Star** marks the global minimum found during the search ($K = 13, \lambda \approx 13$), which sits in the sweet spot balancing model capacity and regularization constraints.

The scatter plot reveals two key insights:

1. **Complexity Cost:** Models with high K (> 40) generally perform worse (darker dots) unless paired with extremely high regularization, indicating that the 32M dataset does not require massive latent dimensions to capture user preference.

2. **Optimal Region:** The cluster of yellow/green points suggests that a moderate K (between 10 and 20) yields the most stable generalization performance.

C.3. Analysis of Model Capacity and Overfitting

To justify our choice of latent dimension (K), we compared the training dynamics of a low-capacity model ($K = 10$) versus a higher-capacity model ($K = 20$). Figure 25 illustrates the generalization gap.

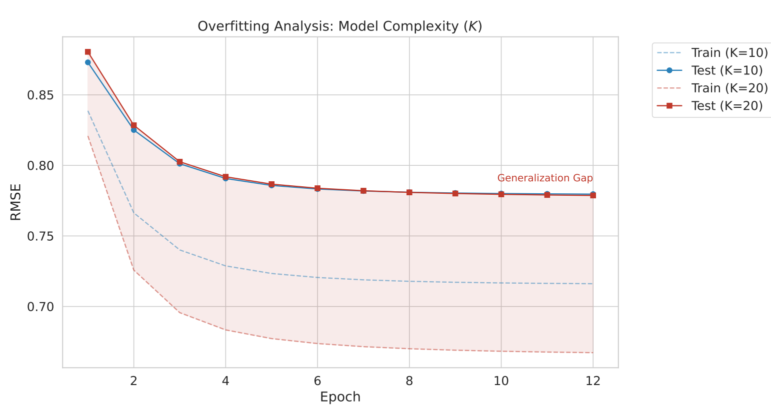

Figure 25. Overfitting Analysis: Model Complexity (K). The red dashed line ($K = 20$ Train) drops significantly lower than the blue dashed line ($K = 10$ Train), indicating the model is memorizing the data. However, the solid lines (Test set) are nearly identical. The shaded area represents the **Generalization Gap**, confirming that without increased regularization, additional latent factors contribute to overfitting rather than true predictive signal.

C.4. Scenario B: Cold Start Prediction

To test the efficacy of the Genre-Integrated regularizer, we asked the model to predict scores for two movies that had zero ratings in the training set: *Toy Story 5* (Comedy/Children) and *Alien: Romulus* (Horror/Sci-Fi).

Figure 26 details the model's decision-making process:

- **Toy Story 5:** The model assigns a **negative score** (-0.02). The breakdown shows that the user vector has a strong negative dot product (yellow bars) with the *Children* and *Comedy* genre centroids.
- **Alien: Romulus:** The model assigns a **positive score** ($+0.006$). The breakdown shows that the positive alignment with *Sci-Fi* and *Action* (red bars) outweighs the slight negative alignment with *Horror*.

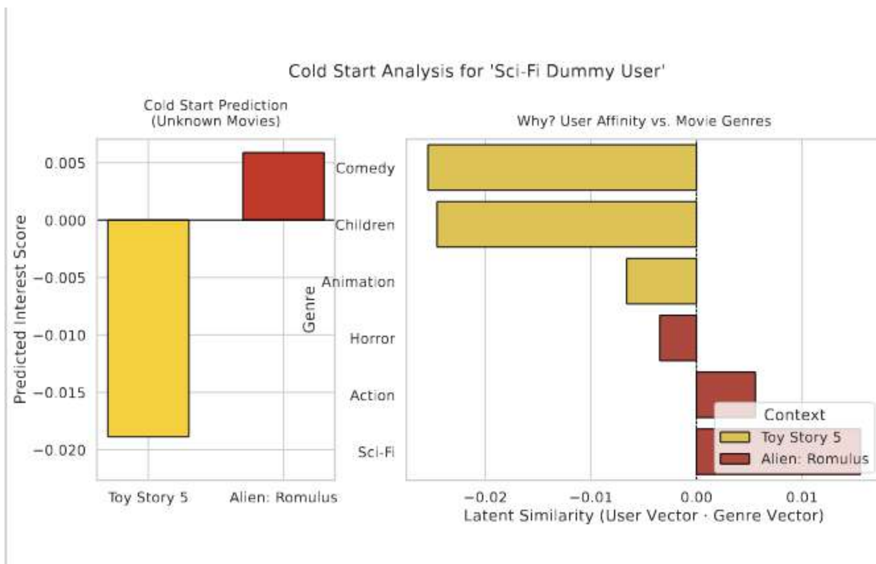

Figure 26. Explainable Cold Start Prediction. The left panel shows the final predicted scores for unrated movies. The right panel decomposes these scores into contributions from specific genres, revealing that the model correctly penalizes 'Comedy' while rewarding 'Sci-Fi' for this specific user profile.