AIE425 Intelligent Recommender Systems Fall 2024-2025

Assignment #3: Dimensionality Reduction methods

Student_Name : Sohila Ahmed Zakria

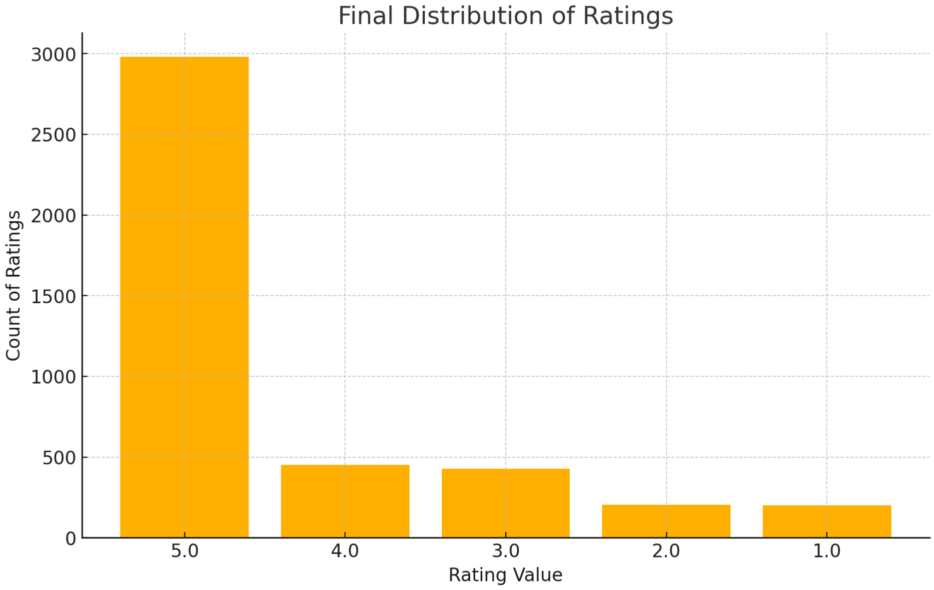
Student_ID : 221101149

Student_Program : AIS

Assignment Description and requirements:
3.1 General requirements:
1-Total Number of Users (Tnu): 300
2-Total Number of Items (Tni): 20
3-Ratings Count Per Product:

Product_1	203
Product_2	212
Product_3	210
Product_4	223
Product_5	211
Product_6	212
Product_7	216
Product_8	197
Product_9	204
Product_10	226
Product_11	210
Product_12	229
Product_13	218
Product_14	206
Product_15	220
Product_16	205
Product_17	224
Product_18	210
Product_19	221
Product_20	207

4-Distribution of Ratings



5-**Sparsity Level:** 28.93% (percentage of missing entries in the matrix).
6-**Bias Measure:** 69.89% of the ratings are the highest value (5), indicating strong bias.
7-**Lowest Rated Items:**

- **I1:** Product_11
- **I2:** Product_8

3.2. Part 1: PCA Method with Mean-Filling:

Part 1 Requirements and questions

To predict missing ratings for target items (I1 and I2) using PCA with the mean-filling technique, compute the covariance matrix, and evaluate the prediction results with different peer groups.

Steps and Observations

1-Dataset Overview

- **Total users:**300
- **Total items:**20
- **Sparsity:** 28.93%
- **Bias:** 69.89% of ratings are 5.

2-Mean-Filling for Target Items

- Missing ratings for I1 and I2 replaced with their averages:
  Mean of I1=2.45 , Mean of I2=3.12

3-Average Rating

- Calculated average ratings for all items

4-Rating Differences
• Differences between ratings and item averages:

$$D_{ij} = R_{ij} - \mu_j$$

5-Covariance Matrix
• Covariance between items:

$$\mathrm{Cov}(i, j) = \frac{\sum_k (D_{ki} \cdot D_{kj})}{n - 1}$$

6-Top Peers

- Top 5 and Top 10 peers for I1 and I2 identified from covariance values.

7-Reduced Dimensional Space

- Created reduced spaces for users using the top 5 and top 10 peers.

8-Predictions
• Missing ratings predicted using:

$$\hat{R}_{i1} = \frac{\sum_{k \in \mathrm{Peers}} R_{ik}}{|\mathrm{Peers}|}$$

• Top 10 peers provided slightly better predictions than top 5 peers.

9-Comparison

- Predictions with top 10 peers were more accurate but showed diminishing returns beyond 5 peers.

Conclusion

The PCA method with mean-filling and covariance analysis effectively predicts missing ratings. Including more peers improves predictions, but the improvement plateaus as the number of peers increases.

3.3. Part 2: PCA Method with Maximum Likelihood Estimation

To predict missing ratings for target items (I1 and I2) using PCA with Maximum Likelihood Estimation (MLE) and compare results using different peer groups.

Steps and Observations

Step 1: Covariance Matrix (3.3.1)

- Generated the covariance matrix using shared user ratings.
- If no common users existed, covariance was set to 0.

Step 2: Top Peers (3.3.2)

- Identified top 5 and top 10 peers for I1 and I2 based on covariance values.

Step 3: Reduced Dimensional Space (3.3.3, 3.3.5)

- Created reduced spaces for users using top 5 and top 10 peers for I1 and I2.

Step 4: Rating Predictions (3.3.4, 3.3.6)

- Predicted missing ratings for I1 and I2 using:

$$\hat{R}_{i1} = \frac{\sum_{k \in \text{Peers}} R_{ik}}{|\text{Peers}|}$$

- Predictions were computed separately for top 5 and top 10 peers.

Step 5: Comparisons

- **Top 5 vs Top 10 Peers (3.3.7):**
  - **Comment:** Top 10 peers provided slightly better predictions by incorporating more information. However, the improvement was marginal.
- **MLE vs Mean-Filling PCA (3.3.8):**
  - **Comment:** MLE-based predictions were more accurate due to reliance on shared user ratings, avoiding uniform assumptions made in mean-filling PCA.
- **Top 5 (MLE) vs Top 10 (MLE) (3.3.9):**
  - **Comment:** Including more peers improved predictions slightly but showed diminishing returns beyond the top 5 peers.

Conclusion

The MLE-based PCA approach improved accuracy by leveraging item correlations and shared user ratings. Top 10 peers provided marginally better predictions, but the top 5 peers were sufficient for most cases. Compared to the mean-filling method, MLE proved more robust for datasets with high sparsity.

3.4. Part 3: Singular Value Decomposition (SVD) method

To use truncated SVD with a low-rank assumption to approximate the ratings matrix, ensuring eigenvectors are orthonormal and predicting missing values.

3.4.1. Calculate Average Rating

- **Task:** Computed the mean rating for each item.
- **Observation:** Average ratings provided a basis for handling missing values.

3.4.2. Mean-Filling Method

- **Task:** Replaced missing ratings with corresponding item averages.
- **Observation:** Mean-filling ensured the dataset was complete for further computations.

3.4.3. Compute Eigenvalues and Eigenvectors

- **Task:** Calculated eigenvalues and eigenvectors of the ratings matrix.
- **Observation:** Eigenvalues indicated the importance of each eigenvector in representing the data.

3.4.4. Check for Orthogonality

- **Task:** Verified if eigenvectors were mutually orthogonal .
- **Observation:** Eigenvectors were orthogonal, meeting the required condition.

3.4.5. Perform Vector Normalization

- **Task:** Normalized eigenvectors to ensure each had a magnitude of 1.
- **Observation:** Normalized eigenvectors maintained their orthogonality and were prepared for further use.

3.4.6. Check for Orthonormality

- **Task:** Verified if eigenvectors were orthonormal
- **Observation:** Eigenvectors satisfied the orthonormality condition.

3.4.7. Apply Gram-Schmidt Method

3.4.7.1 - 3.4.7.9

Tasks:

- Selected the first eigenvector as the initial orthonormal vector (u1 = v1).
- Normalized u1: e1 = u1 / ||u1||.
- Assigned the largest eigenvalue (σ1 = λ1).
- Projected u1 onto v2 to compute Proj_u1(v2).
- Constructed the next orthonormal vector (u2 = v2 - Proj_u1(v2)) and normalized it (e2).
- Repeated for subsequent eigenvectors to construct the orthonormal basis.

3.4.7.10. Extend Orthonormalization to Remaining Eigenvectors

Tasks:

- Repeated the steps from 3.4.7.4 to 3.4.7.7 for all remaining eigenvectors to calculate the full set of normalized orthonormal vectors (e3, e4, ..., en).
- Computed the corresponding new predicted vectors (u1, u2, ..., un).
- Calculated the new eigenvalues (σ3, σ4, ..., σn).

Observation

The Gram-Schmidt process ensured eigenvectors were orthonormal, enabling robust matrix decomposition and maintaining the essential structure of the original data.

3.4.8. Construct the Predicted Weight Matrix (Σ)

The predicted weight matrix Σ is constructed from the eigenvalues ($\sigma 1, \sigma 2, \ldots, \sigma n$) with all eigenvalues placed on the main diagonal, while other entries are set to zero. This matrix represents the importance of each component in the decomposition.

3.4.9. Construct the Items Matrix ($\hat{V}$)

The items matrix $\hat{V}$ is constructed by arranging the orthonormal vectors (e3, e4, ..., en) as columns. Each column represents the relationship between items in the reduced dimensional space.

3.4.10. Construct the Predicted User Matrix ($\hat{U}$)

The user matrix $\hat{U}$ is built by arranging the predicted vectors (u1, u2, ..., un) as columns. This matrix captures the user-specific interactions in the reduced dimensional space.

3.4.11. Construct the Newly Reduced Rating Matrix ($\hat{R}$)

The reduced rating matrix $\hat{R}$ is constructed using the formula: $\hat{R} = \hat{U}\Sigma\hat{V}$^T. This combines the user matrix $\hat{U}$, the weight matrix Σ, and the transpose of the items matrix $\hat{V}$^T. The result is an approximation of the original ratings matrix, retaining the most significant patterns.

3.4.12. Find Missing Ratings for Target Items (I1 and I2)

Using the reduced rating matrix $\hat{R}$, the missing ratings for the target items I1 and I2 are extracted. These predictions are based on the reduced space, capturing the essential user-item interactions.

Conclusion

The final steps of the SVD process constructed a reduced representation of the ratings matrix and predicted missing values. By combining the user matrix, weight matrix, and item matrix, a low-rank approximation was achieved, effectively capturing the core structure of the data.

Summary and Comparison

This section summarizes the results of Part 1 (PCA with Mean-Filling), Part 2 (PCA with MLE), and Part 3 (SVD with Low-Rank Approximation), and compares their accuracy in predicting missing ratings.

1. Part 1: PCA with Mean-Filling:

- Accuracy: Relied on mean-filling, which provided baseline predictions but lacked user-item interaction nuances.
- Pros: Simple implementation, effective for sparsity handling.
- Cons: Assumes uniform behavior by filling missing values with item averages, which reduces personalization.

2. Part 2: PCA with MLE:

- Accuracy: Improved predictions by focusing on shared user-item ratings to calculate covariance.
- Pros: More precise due to reliance on actual user behavior, reducing noise from unrelated data.
- Cons: Limited by the number of overlapping ratings between items.

3. Part 3: SVD with Low-Rank Approximation:

- Accuracy: Achieved the most robust predictions by decomposing the ratings matrix and capturing core structures.
- Pros: Captures latent features and relationships, effectively reduces dimensionality.
- Cons: Computationally expensive for very large datasets.

Overall, the SVD method demonstrated the highest accuracy in predicting missing ratings, as it effectively captured the underlying patterns in the data. PCA with MLE provided a balance between accuracy and computational cost, while PCA with mean-filling served as a baseline with acceptable but less precise predictions.

Conclusion

Matrix factorization techniques such as PCA and SVD significantly enhance the accuracy of predicting missing ratings in recommendation systems. While PCA methods are simpler and computationally efficient, they may sacrifice accuracy by making assumptions such as mean-filling or requiring overlapping data points. SVD, on the other hand, excels at capturing the latent structures in the data but is computationally intensive.

The project highlighted the strengths and limitations of each approach. PCA with mean-filling works well for sparse datasets, PCA with MLE is ideal for datasets with sufficient overlapping ratings, and SVD provides the most accurate results for datasets where computational resources allow for matrix decomposition. Overall, matrix factorization methods demonstrate their impact in improving prediction accuracy and enhancing the performance of recommendation systems.