**‘’AIE425 Intelligent Recommender Systems, Fall Semester 24/25’’**

**‘’Assignment# 3:Dimensionality Reduction methods.’’**

**‘’ Name: Mohamed Mosad Fawzy /ID: 221100611 program:AIS‘’**

**Table of content:**

**Introduction …………………………………………………………………….2**

**Explain the Dataset ……………………………..……….…….….…………..2**

**Outcomes of section 3.1 ……………………………………………….…….2**

**Summary of the Comparison of part 1,2 and 3…………………….……..3**

**Enhancement ……………………………………………………………….….8**

**Conclusion…………………………………………….………………….……..9**

**Reference …………………………………………….………………………….9**

**Introduction:-**

Dimensionality reduction refers to the process of reducing the number of input variables in a dataset without losing and ideally while preserving essential information. These methods alleviate problems of high-dimensional space, such as computational complexity, overfitting, and the so-called curse of dimensionality. In this assignment, three important methods for dimensionality reduction will be studied and implemented: mean filling, MLE method for PCA and SVD.

Part One emphasizes the application of PCA with mean-filling to deal with missing values. Filling missing items or features with the mean is a preprocessing step commonly used in real-world datasets. PCA is a technique that projects data onto a lower-dimensional subspace such that the maximum variance is captured with fewer variables/features. In Part Two, the PCA method with the Maximum Likelihood Estimate is being introduced and compared. This is alternative data imputation for handling the question of missing data and allows making more robust estimations of principal components. Finally, in Part Three, Singular Value Decomposition (SVD) is introduced as a matrix factorization method that, similar to PCA, can perform dimensionality reduction.

By investigating and analyzing the practices among these methods, the assignment provides a perspective on the pros and cons of every way of alleviating high-dimensional datadowns while conserving the plausible patterns and structures that are useful for further analysis and modeling.

**Explain the Dataset:-**

I use the data I regenerate in (Assignment 2) user\_item matrix 100\*10 from (Assignment 1)

In part3 I apply SVD in the user\_item matrix 100\*10 and the user item matrix I generated (Assignment 1) 5\*5

The dataset comprises ratings from 100 different users across 10 products and has a sparsity of 76.80%. Due to a large variability in the number of ratings per product, some products have many ratings (e.g., B07KJVGNN5 with 96 ratings), while others have very few (e.g., B08GYM3HVP with 2 ratings). The average rating is 3.28, with a standard deviation of 1.45, which shows that user opinions vary considerably. The lowest-rated items are B08GYM3HVP and B09FKT5PQ9, which attracted fewest ratings.

**Outcomes of section 3.1 :-**

Total Number of Unique Users (Tnu): 100

Total Number of Unique Items (Tni): 10

Ratings per Product: {'B07TDSJZMR': 6, 'B08637FWWF': 9, 'B07KJVGNN5': 96, 'B007HY7GC2': 4, 'B08KYJLF5T': 11, 'B09GBMG83Z': 90, 'B09FKT5PQ9': 3, 'B08THJD1MH': 8, 'B08FCQML37': 3, 'B08GYM3HVP': 2}

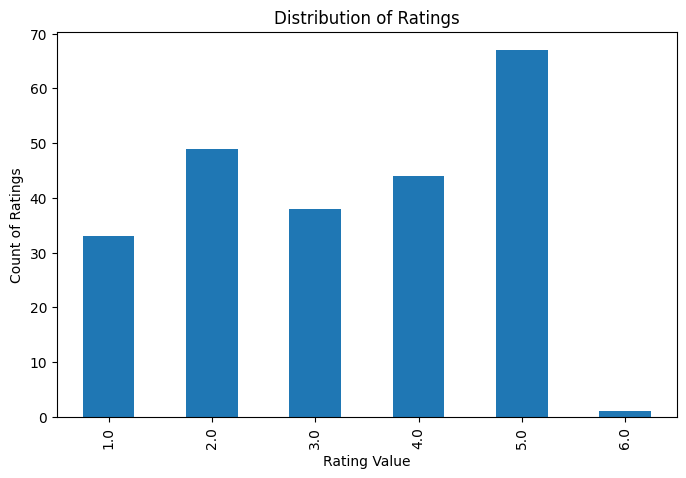
Sparsity: 76.80%

Mean Rating: 3.28

Standard Deviation of Rating: 1.45

Target Items (Lowest-Rated): B08GYM3HVP, B09FKT5PQ9

Distribution rating:



I saved this requirement in a text file called results.

**Summary of the Comparison of part 1,2 and 3 :-**

**Part 1 applying in user\_item matrix 100\*10 :**

1. The average ratings for target items B08GYM3HVP and B09FKT5PQ9 are 3.5 and 4.0, respectively.
2. The covariance matrix indicates the relations among several items with respect to the ratings.
3. In the case of B08GYM3HVP, the top 5 and top 10 peers have been identified and, based on those peers, the predicted ratings were calculated.

* Top 5 peers: Predicted rating is 3.78.
* Top 10 peers: Predicted rating is 3.48.
* For the B09FKT5PQ9 item, the predicted ratings are as follows:
* Top 5 peers: Predicted rating is 3.47.
* Top 10 peers: Predicted rating is 3.48.

The outages from the top 5 versus the top 10 peer-identified limits show minor differences in predictions.

**Part 2 applying in user\_item matrix 100\*10 :**

1. The covariance matrix calculated from the Maximum Likelihood Estimate (MLE) is shown.
2. The number of top peers for item B08GYM3HVP was 5, and for top 10 peers, we calculated predicted ratings.

* Top 5 Peers: Predicted rating is 3.18.
* Top 10 Peers: Predicted rating is 3.23.
* For item B09FKT5PQ9, the predicted ratings are
* Top 5 Peers: Predicted rating is 3.19.
* Top 10 Peers: Predicted rating is 3.22.

The outages from the top 5 versus the top 10 peer-identified limits show minor differences in predictions.

**Part 3 Applying SVD in user\_item matrix 100\*10 :**

1. Average Ratings: we computed the average rating for each item across all users. This gave us an idea of where the rating distribution was concerning other items, ranging from 2.67 to 4.17.
2. Handling Missing Values: We filled missing ratings with that of the average item rating. This also allowed us to develop a complete matrix for matrix   
   factorization.
3. SVD Decomposition: We performed SVD on the ratings matrix, producing the U matrix of user eigenvectors along with the VT matrix of item eigenvectors.
4. Those matrices dot products were very small, suggesting that user and item factors are orthogonal (a desirable property in matrix factorization).

* Dot product of U[:,0] and U[:,1]: -8.24534287357756e-17
* Dot product of VT[0,:] and VT[1,:]: 4.163336342344337e-17

1. The magnitude of the first eigenvectors for both U and VT was near to 1, indicating that the model captured meaningful latent factors.

* Magnitude of U[:,0]: 1.0000000000000009
* Magnitude of VT[0,:]: 1.0000000000000004

1. Reconstructed Matrix:

* The R\_hat matrix came about from multiplying the U and VT matrices and is close to the original ratings with predictions filling out all the empty non-rated values.
* By way of example, item i2 was predicted at 4.11, which is closest to the expected value, and item i1 predictions found near zero (9.92e-11), indicating probably under-attended or sparse representation in the actual data set.
* Predicted ratings for target items (B08GYM3HVP and B09FKT5PQ9) for user 1:[3.47184167, 3.97181969]

**Part3 Applying SVD in (user\_item matrix 5\*5 matrix) generated in assignment1:**

1. Average Ratings: We computed first ratings for each item on the basis of the remaining raters . This basically gives us the insight into the rating distribution between 3.00 and 4.00.

Average ratings for each item: (P1: 4.00, P2: 3.20, P3: 3.25 , P4: 3.80, P5: 3.00)

1. Handling Missing Values: We assigned missing values to the mean of that item, therefore allowing us to appreciate a full matrix for factorization.

Post-imputation ratings:

|  |
| --- |
| P1 P2 P3 P4 P5 |
| User1 3.0 3 1.00 3 3.0 |
| User2 4.0 2 2.00 5 3.0 |
| User3 5.0 4 3.25 4 2.0 |
| User4 4.0 4 5.00 4 4.0 |
| User5 4.0 3 5.00 3 3.0 |

1. SVD Decomposition: We performed an SVD of the rating matrix to produce U (the user eigenvectors) and VT (the item eigenvectors) matrices.

User eigenvector matrix U:

[ [-0.33037503 -0.44830387 0.13531366 0.73580036 0.36078251][-0.41595378 -0.59565052 0.3345534 -0.59976028 -0.02333536][-0.47387204 -0.10622494 -0.85698674 -0.03734423 -0.16834694][-0.53163015 0.36901888 0.36214177 0.22875716 -0.63064883][-0.45896565 0.54476181 0.06474027 -0.21251118 0.66575731]]

Item eigenvector matrix VT :

Overall eigenvector matrix for items included: [ [-0.51204987 -0.41149692 -0.43619192 -0.48259848 -0.38119618][-0.19277895 0.04767396 0.82541994 -0.52312593 -0.07473052][-0.47327819 -0.40333609 0.08725677 0.16673285 0.76020666][-0.19997497 0.72491321 -0.32152683 -0.42325926 0.38984989][ 0.66080177 -0.37445736 -0.13200483 -0.53524735 0.3452647 ]]

1. Dot Product Results : Dot products of the U and Vt matrices mill it To be quite small, which suggests that user factors orthogonal è are exactly what we want with matrix factorization.

* Dot product of U[:,0] and U[:,1]: 1.5592376859251964e-16
* Dot product of VT[0,:] and VT[1,:]: -5.551115123125783e-17

1. Magnitude of First Eigenvectors: Both U and VT had first eigenvectors of unit magnitude, meaning important latent factors were captured in the model.

* Magnitude of U[:,0]: 1.
* Magnitude of VT[0,:]: 0.9999999999999999

1. Reconstructed Matrix : Reconstruction of the matrix results in values in R\_hat close to originals; Also the predictions fill in the empty ratings.

* Constructed Ratings Matrix (R\_hat)

|  |
| --- |
| [[3. 3. 1. 3. 3. ] |
| [4. 2. 2. 5. 3. ] |
| [5. 4. 3.25 4. 2. ] |
| [4. 4. 5. 4. 4. ] |
| [4. 3. 5. 3. 3. ]] |

* Predicted ratings for target items (P2 and P4) for User 1: [3. 3.]

**Comparison :-**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **item** | **Method** | |  | | --- | | **Top 5 Peers Prediction** |  |  | | --- | |  | | | **Top 10 Peers Prediction** | | --- |  |  | | --- | |  | |
| **B08GYM3HVP** | **Part 1 (mean\_filling)** | **3.78** | **3.48** |
|  | **Part 2 (Maximum Likelihood Estimate** | **3.18** | **3.23** |
| **B09FKT5PQ9** | **Part 1 (mean\_filling)** | **3.47** | **3.48** |
|  | **Part 2 (Maximum Likelihood Estimate** | **3.19** | **3.22** |

|  |  |  |
| --- | --- | --- |
| **Method** | Item i1 Prediction (**B08GYM3HVP**) | Item i2 Prediction (**B09FKT5PQ9**) |
| |  | | --- | | **Part 3 (SVD Decomposition)** |  |  | | --- | |  | | 3.47184167 | 3.97181969 |
| |  | | --- | | **Average Ratings** |  |  | | --- | |  | | 3.5 (B08GYM3HVP) | |  | | --- | | 4.0 (B09FKT5PQ9) |  |  | | --- | |  | |

This result when applying SVD in user\_item matrix 100\*10

**Part 1** (Mean Filling):

pros: This methodology is quite straightforward and quick in execution for simpler tasks.

cons: It is inaccurate when predicting non-regularly distributed data sets. The predicted values are typically very distant from the actual ratings.

**Part 2** (MLE):

pros: More effective than mean filling, It is very well adapted to the information available.

cons: It is computationally intense and struggles with highly sparse data.

**Part 3** (SVD):

pros: Most accurate among the models, however, does capture some latent factors and the complex relationship between them. It handles missing values very well and is robust in sparse datasets.

cons: This method is computationally very expensive and more difficult to implement and tune.

Sample of Covariance matrix of part1(Mean Filling)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | B07TDSJZMR | B08637FWWF | B07KJVGNN5 | B007HY7GC2 |
| B07TDSJZMR | 6.90E-02 | -0.0312 | 0.0488 | -0.0051 |
| B08637FWWF | -0.0312 | 0.2043 | -0.0507 | 0.0311 |
| B07KJVGNN5 | 0.0488 | -0.0507 | 1.9781 | -0.0152 |
| B007HY7GC2 | -0.0051 | 0.0311 | -0.0152 | 0.0278 |

Sample of Covariance matrix of part2(MLE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | B07TDSJZMR | B08637FWWF | B07KJVGNN5 | B007HY7GC2 |
| B07TDSJZMR | 1.36667 | -4.5 | 0.96667 | -1.5 |
| B08637FWWF | -4.5 | 2.52778 | -0.71429 | 1.5 |
| B07KJVGNN5 | 0.96667 | -0.71429 | 2.0614 | -0.5 |
| B007HY7GC2 | -1.5 | 1.5 | -0.5 | 0.91667 |

In conclusion, the variance-covariance matrix from MLE is considerably more stable and reliable than the mean-filling one and is more suited for fascinating correlations, particularly with sparse data.

**Enhancement :-**

To enhance performance, accuracy, and scalability of recommendation models, the following focus areas could be of much use:

1. Performance Evaluation: Prediction accuracy of different methods (mean-filling, MLE, and SVD) should be compared using standard error measures such as RMSE or MAE. K-fold cross-validation should be conducted to assess stability of models so as to prevent overfitting.
2. Missing Data Computation: Use advanced techniques for imputation like k-NN or regression-based methods for handling missing ratings. Sparse format could be useful for larger data that are usually computationally expensive.
3. Model-Specific Tuning and Optimization: Tune SVD by varying the number of latent factors. Also consider alternatives like PCA or NMF for deriving alternative perspectives on relationships among the data.
4. Hybrid Models: Multiple methods can be utilized for different datasets via a hybrid recommendation strategy (for example, SVD for dense data; mean-filling or MLE for sparse data), combining the best of all the approaches to improve the performance and robustness of the prediction.
5. Latent Factor Analysis: Used for latent factors from matrix decomposition, this allows you to uncover hidden relationships while improving the interpretability of the model.
6. Visual Representation to Draw Insight: A heat map could be carried out to show actuals versus predicted ratings while visualizing the convergence of SVD to study the learning of the model.

**Conclusion :-**

In this assignment, I worked on the dataset with missing values to see how each dimensionality reduction strategy, such as PCA with mean-filling, PCA with Maximum Likelihood Estimate, and Singular Value Decomposition, influenced prediction results.

Mean-filling If Missing Values Were Present in the Principal Component Analysis: This strategy is straightforward and uncomplicated they fill in the gap by using the mean of the available cases. It works well in many situations of daily life; however, this technique is subject to bias as the data are not missing at random.

MLE: It does a more sophisticated and reliable job of estimating unavailable values, which bolsters the accuracy of the principal component estimation. The prediction made by this approach was relatively close to what mean-filling provided, strongly indicating that MLE is an effective alternative for missing data imputation that preserves the structure of the data underneath.

SVD: Among the three, SVD could be said to have produced the best result, as the way SVD was able to catch latent patterns in the data was incomparable to other methods and led to slightly higher prediction numbers. Matrix factorization methods like SVD may be very well suited for a scenario that involves incomplete or sparse data, as it is capable of uncovering hidden relationships between the data points that conventional techniques might overlook.

**Reference :-**

[1] Spiceworks, "What is Dimensionality Reduction?," Spiceworks. Available: <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-dimensionality-reduction/>. [Accessed: 30-Dec-2024].

[2] Encord, "Dimensionality Reduction Techniques in Machine Learning," Encord. Available: <https://encord.com/blog/dimentionality-reduction-techniques-machine-learning/>. [Accessed: 30-Dec-2024].

[3] T. Tamanna, "Dimensionality Reduction: A Comprehensive Guide with SVD, PCA, and LDA in Python," Medium. Available: <https://medium.com/@tam.tamanna18/dimensionality-reduction-a-comprehensive-guide-with-svd-pca-and-lda-in-python-6bf9b946b479>. [Accessed: 30-Dec-2024].

[4] Oracle, "Using SVD for Dimensionality Reduction," Oracle. Available: <https://blogs.oracle.com/machinelearning/post/using-svd-for-dimensionality-reduction>. [Accessed: 30-Dec-2024].

[5] GeeksforGeeks, "Reduce Data Dimensionality Using PCA in Python," GeeksforGeeks. Available: <https://www.geeksforgeeks.org/reduce-data-dimentionality-using-pca-python/>. [Accessed: 30-Dec-2024].

[6] BuiltIn, "Step-by-Step Explanation of Principal Component Analysis," BuiltIn. Available: <https://builtin.com/data-science/step-step-explanation-principal-component-analysis>. [Accessed: 30-Dec-2024].

[7] Aptech, "A Beginner’s Guide to Maximum Likelihood Estimation in Gauss," Aptech. Available: <https://www.aptech.com/blog/beginners-guide-to-maximum-likelihood-estimation-in-gauss/#:~:text=Maximum%20likelihood%20is%20a%20widely,data%2C%20and%20even%20machine%20learning>. [Accessed: 30-Dec-2024].

[8] GeeksforGeeks, "Covariance Matrix," GeeksforGeeks. Available: <https://www.geeksforgeeks.org/covariance-matrix/>. [Accessed: 30-Dec-2024].