Comparative Analysis of Movie Recommendation Systems

Vikash Singh John, Darshan Jadhav
vxs465, dxj294

April 29, 2024

Introduction

This report provides a comprehensive evaluation of three distinct collaborative filtering techniques for movie recommendations: Singular Value Decomposition (SVD), Non-negative Matrix Factorization (NMF), and K-Nearest Neighbors (KNNBasic). These techniques are assessed based on metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), precision, recall, and computational efficiency.

Methodology

Data Preprocessing

The user-item rating dataset contains four attributes: user ID, item ID, rating, and timestamp. The preprocessing steps are as follows:

1. Load the data into a pandas DataFrame.
2. Normalize the timestamp attribute to a 0-1 scale using the MinMaxScaler:

$$t' = \frac{t - \min(t)}{\max(t) - \min(t)}$$

3. Transform the DataFrame into a format compatible with the Surprise library, specifying the rating scale from 1 to 5.

Algorithms

Singular Value Decomposition (SVD)

Mathematical Foundation: SVD decomposes the user-item interaction matrix R into three matrices U, Σ, and V^T:

$$R \approx U\Sigma V^T$$

where U (user features matrix) and V^T (item features matrix) contain the left and right singular vectors, and Σ is a diagonal matrix of singular values.

Algorithm Flow:

1. Initialize U, Σ, and V^T with random values.
2. Use stochastic gradient descent (SGD) to minimize the reconstruction error between R and $U\Sigma V^T$.
3. Regularize the factors to prevent overfitting.
4. Iterate until convergence or a specified number of epochs is reached.

Non-negative Matrix Factorization (NMF)

Mathematical Foundation: NMF factorizes R into two non-negative matrices W (user matrix) and H (item matrix) that approximate the original matrix:

$$R \approx WH$$

Algorithm Flow:

1. Initialize W and H with non-negative values.
2. Apply an iterative update rule to minimize the cost function, typically the Frobenius norm of $R - WH$.
3. Enforce non-negativity constraints on W and H throughout the optimization.
4. Continue until the change in the cost function is below a threshold or a maximum number of iterations is reached.

K-Nearest Neighbors (KNNBasic)

Mathematical Foundation: KNN predicts a user's rating for an item based on the weighted average of ratings of nearest neighbor items or users.

Algorithm Flow:

1. Calculate the distance or similarity between items or users based on their ratings. Common metrics include cosine similarity and Pearson correlation.
2. Identify the k nearest neighbors with the highest similarity scores.
3. Compute the predicted rating as a weighted average of the neighbors' ratings.
4. Optionally, adjust for the mean ratings of each user to normalize the ratings across different scales.

Parameter Tuning

We employ grid search to optimize the parameters for each algorithm. This technique systematically explores a range of values for each parameter, conducting cross-validation to determine which combination of parameters yields the lowest RMSE and MAE.

Results and Discussion

SVD only Performance

The model was evaluated using 5-fold cross-validation. The following table summarizes the RMSE for each fold, along with the mean and standard deviation of the RMSE across all folds:

Metric	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Mean (Std)**
RMSE (testset)	0.2364	0.2358	0.2336	0.2361	0.2358	0.2355 (0.0010)
Fit time (s)	1.09	1.14	1.14	1.25	1.08	1.14 (0.06)
Test time (s)	0.14	0.09	0.09	0.15	0.14	0.12 (0.03)

Table 1: Cross-validation results of SVD

The best RMSE achieved was 0.2362, and the mean RMSE across all folds was 0.2356 with a standard deviation of 0.0013. Similarly, the mean MAE was 0.1867 with a standard deviation of 0.0011. These metrics indicate a consistent performance across different data splits.

Inference

These results demonstrate that the SVD model achieves consistent and reliable prediction accuracy on the ML-100k dataset. The low standard deviation in both RMSE and MAE across the folds highlights the model's stability and robustness. The mean RMSE and MAE values close to the best RMSE suggest that the model not only performs well on average but also maintains this performance reliably across multiple data splits. The differences in fit and test times indicate variability in computational performance, possibly due to system load or hardware specifics, yet the overall average times show that the model is relatively efficient. This performance balance makes SVD a compelling choice for recommender systems requiring both accuracy and efficiency.

Comparative Results of Multiple algorithms

The performance of each algorithm is tabulated in two parts to clearly display all metrics:

Accuracy and Efficiency

Algorithm	RMSE	MAE	Fit Time (s)	Test Time (s)
SVD	0.924 ± 0.005	0.729 ± 0.003	0.677	0.113
NMF	0.944 ± 0.005	0.750 ± 0.003	1.395	0.093
KNNBasic	0.975 ± 0.005	0.770 ± 0.003	0.553	2.469

Table 2: Error metrics and computational times

Discussion of Accuracy and Efficiency: The Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are critical metrics for assessing the accuracy of predictions made by each algorithm. Lower values indicate more accurate predictions. As shown, SVD achieves the lowest RMSE and MAE, suggesting that it provides the most accurate predictions for this dataset.

Regarding computational efficiency, the fit time reflects the training duration while the test time represents the processing time needed to evaluate the model on the test set. SVD and NMF demonstrate reasonable fit times, but NMF's slightly longer fit time may impact scalability. KNNBasic, although faster to train (lower fit time), shows significantly longer test times due to its computationally intensive distance calculations between users, especially in larger datasets.

Precision and Recall

Algorithm	Precision	Recall
SVD	1.000 ± 0.000	0.006 ± 0.000
NMF	1.000 ± 0.000	0.006 ± 0.000
KNNBasic	1.000 ± 0.000	0.006 ± 0.000

Table 3: Precision and recall metrics

Discussion of Precision and Recall: Precision measures the proportion of positive identifications that were actually correct, while recall measures the proportion of actual positives that were correctly identified. The perfect precision scores indicate that when the models do make a recommendation, it is almost always relevant to the user. However, the extremely low recall scores reveal a significant limitation: the models are not capturing the vast majority of relevant items. This could suggest that the models are too conservative or that the threshold set for deciding what constitutes a 'relevant' recommendation is too strict. Enhancing recall without sacrificing precision could involve techniques like adjusting the recommendation threshold, incorporating richer feature sets into the model, or employing more sophisticated model tuning strategies.

Comparative Analysis

Accuracy and Error Metrics: SVD exhibits the lowest RMSE and MAE, indicating it is the most accurate at predicting ratings within this dataset. NMF, while slightly less accurate, provides the benefit of interpretable components due to its non-negative constraints. KNNBasic shows higher errors and the longest test times, suggesting less suitability for larger datasets or real-time scenarios.

Computational Efficiency: SVD and NMF demonstrate reasonable fit times, but NMF's slightly longer times may affect scalability. KNNBasic, despite a shorter fit time, has significantly longer test times due to its computational complexity when calculating distances between large sets of users.

Precision and Recall Implications: The perfect precision across all models indicates that items rated highly by the model are generally well-received by users. However, the near-zero recall suggests that these models are failing to identify the majority of relevant items. This could be due to the sparsity of the data or limitations in how the neighborhood or latent features are being utilized and modeled.

Conclusion

The comparative analysis underscores SVD as the most effective model for this dataset based on the balance of accuracy and efficiency. The universal low recall across models indicates a potential area for improvement in future studies, possibly by integrating richer contextual data or exploring hybrid models. Further investigation into parameter tuning and threshold adjustments might also help enhance the recall rate, thus improving the overall utility of the recommendation systems.

Acknowledgment

We extend our heartfelt gratitude to Dr. Jing Ma for her invaluable instruction and mentorship in the field of Data Mining. Her expertise and insightful guidance have been pivotal in the successful completion of this project. Her dedication to fostering a deep understanding of complex concepts has profoundly enhanced my learning experience.

References

[1] Yehuda Koren, Robert Bell, Chris Volinsky. *Matrix Factorization Techniques for Recommender Systems*. Computer, 42(8):30–37, IEEE, 2009.

[2] Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor. *Recommender Systems Handbook*. Springer, 2011.

[3] Surprise: A Python scikit for building and analyzing recommender systems. Available online: `http://surpriselib.com`. Accessed: 2024-04-28.