Final Project Report

Giannios Panagiotis, Gkolias Theodoros, Makris Fragkiskos

January 2024

1 Abstract

This report presents a comprehensive analysis of user-item interactions within a digital gaming platform, leveraging two distinct recommendation methodologies: Singular Value Decomposition (SVD) and Neural Networks. The motivation behind this study stems from the growing importance of personalized recommendation systems in enhancing user experience and engagement within online gaming platforms. As the volume of user-generated data continues to surge, the need for effective algorithms to uncover patterns in user preferences becomes paramount.

To achieve this, we employed the SVD method to model user-game interactions and predict playtime. The sparsity-aware SVD method offers insights into user preferences by capturing latent factors contributing to user-game interactions. Additionally, a neural network model was trained to predict playtime, providing an alternative approach to personalized recommendations. The comparative analysis of the SVD and neural network models sheds light on their respective strengths and weaknesses in capturing user preferences within the gaming platform.

Results from both methodologies are presented and evaluated, showcasing their efficacy in generating personalized game recommendations. The recommendations generated by the SVD method are compared with those from the neural network model, providing a nuanced understanding of the diverse approaches to recommendation systems. The insights derived from this study contribute to the broader field of recommender systems, offering valuable perspectives for the design and enhancement of personalized gaming experiences.

2 Introduction

In the contemporary landscape of digital entertainment, online gaming platforms play a pivotal role in providing immersive experiences to millions of users globally. The sheer volume of available games, coupled with diverse user preferences, underscores the need for robust recommendation systems to enhance user engagement and satisfaction. This study addresses the challenge of personalized game recommendations within a digital gaming platform, aiming to decipher complex user-game interactions and deliver tailored suggestions.

The significance of personalized recommendations in online gaming lies in their potential to not only enrich user experiences but also drive user retention and platform loyalty. As users navigate through extensive catalogs of games, an effective recommendation system serves as a guiding force, connecting users with titles that align with their preferences and play styles. Beyond enhancing user satisfaction, such systems contribute to the overall success and competitiveness of gaming platforms in an industry characterized by dynamic and rapidly evolving content.

Our motivation for undertaking this study stems from the broader context of recommender systems and their impact on user engagement across various online platforms. Leveraging two distinct methodologies, namely Singular Value Decomposition (SVD) and Neural Networks, we seek to explore and compare their effectiveness in modeling and predicting user-game interactions. The input to our algorithms comprises user-game interaction data, specifically the 'game_play.dat' dataset, which includes user IDs, game IDs, and corresponding playtime hours. By employing SVD and Neural Networks, our objective is to output personalized game recommendations tailored to individual user preferences. The comparative analysis of these methodologies will not only contribute insights to the field of recommender systems but will also provide actionable recommendations for optimizing personalized gaming experiences.

3 Literature Review

The realm of recommender systems has witnessed significant research efforts in the context of online platforms, encompassing various domains such as e-commerce, streaming services, and digital gaming. In the specific domain of personalized gaming recommendations, several approaches have been explored, revealing a diverse landscape of methodologies.

One prominent category of recommendation systems for gaming platforms involves collaborative filtering techniques. Research by Bell et al. (2007) and Koren et al. (2009) has highlighted the efficacy of matrix factorization methods, such as Singular Value Decomposition (SVD), in capturing latent factors underlying user preferences. These approaches excel in providing accurate recommendations by uncovering hidden patterns within user-item interaction matrices. However, their performance can be hindered by data sparsity and scalability issues, especially in platforms with rapidly growing user bases and expansive game catalogs.

Another approach gaining traction is deep learning, with neural networks emerging as powerful tools for personalized recommendations. He et al. (2017) introduced the Neural Collaborative Filtering (NCF) model, which combines matrix factorization with neural networks, demonstrating improved performance over traditional methods. While neural networks showcase robustness in handling complex feature interactions, they may require substantial computational resources and data for effective training, posing challenges in resource-constrained environments.

In our work, we draw inspiration from these collaborative filtering and neural network-based approaches. Our utilization of SVD reflects the influence of matrix factorization methods, enabling us to uncover latent features contributing to user-game interactions. Simultaneously, the employment of a neural network for predicting playtime aligns with the evolving landscape of deep learning in recommender systems. By comparing the strengths and weaknesses of these approaches, our study aims to contribute insights into the state-of-the-art methodologies for personalized gaming recommendations.

In summary, the literature review underscores the significance of collaborative filtering and neural network approaches in the field of recommender systems, offering valuable insights into their strengths and limitations. Our work aims to bridge the gap between these methodologies, providing a comparative analysis to inform the design and enhancement of personalized gaming experiences.

4 Dataset and Features

The dataset employed in this study comprises user-game interaction data obtained from a popular digital gaming platform. Specifically, two datasets, namely 'game_play.dat' and 'item_info.dat,' are utilized. The 'game_play.dat' dataset encapsulates 70,490 interactions by 11,350 users on 3,600 games, capturing user IDs, game IDs, and corresponding playtime hours. Additionally, 'item_info.dat' provides essential information about the games, including Game_ID and Game Name.

Preprocessing was conducted to ensure the quality and integrity of the data. Duplicate entries in the 'game_play.dat' dataset, identified based on the combination of User_ID and Game_ID, were removed, preserving the first occurrence. This step mitigates potential biases introduced by duplicate entries and ensures the accuracy of subsequent analyses.

Normalization techniques were applied to the playtime values to bring them within a standardized range. The sparsity of the user-game interaction matrix was considered, guiding the decision to employ Singular Value Decomposition (SVD) for matrix factorization. Additionally, data splitting was performed to create distinct training, validation, and test sets for model training and evaluation.

For the neural network model, the 'game_play.dat' dataset was further processed to create input features. A 2D array, initialized with zeroes and representing user-game interactions, was constructed. The array dimensions were determined by the maximum User_ID and Game_ID, with playtime values placed in the corresponding cells.

The features used in both the SVD and neural network models include User_ID, Game_ID, and playtime hours. These features are crucial for capturing user preferences and interactions within the gaming platform. The extracted features were employed directly as input for the SVD model, while for the neural network model, the 2D array representation of user-game interactions served as the input.

No image data is involved in this study; instead, the focus is on time-series data representing user-game interactions. Therefore, techniques such as Fourier transforms, word2vec, HOG, or PCA were not applicable in this context.

The datasets were obtained from the Steam platform, a leading digital distribution platform for PC gaming. Steam provides extensive datasets for research purposes, allowing for a comprehensive analysis of user behavior within the gaming ecosystem.

Examples from the 'game_play.dat' dataset, showcasing user-game interactions, are as follows:

User_ID	Game_ID	Hours
1	1	273.0
2	22	2.3.0
3	41	238.0
3	1	58.0
17	22	1784.0

These examples illustrate the essential components of the dataset, forming the foundation for our analyses and personalized recommendation models.

5 Methods

5.1 Singular Value Decomposition (SVD)

Singular Value Decomposition is a matrix factorization technique employed in collaborative filtering-based recommendation systems. Given the user-item interaction matrix R, where R_{ij} represents the playtime hours of user i for game j, SVD decomposes R into three matrices U, Σ, and V^T such that $R \approx U\Sigma V^T$. Here, U represents the user matrix, Σ is a diagonal matrix containing singular values, and V^T is the transposed item matrix.

Mathematically, the objective is to minimize the Frobenius norm of the difference between R and the reconstructed matrix $U\Sigma V^T$:

$$\min_{U,\Sigma,V} \|R - U\Sigma V^T\|_F^2$$

The optimization is achieved through iterative methods, where gradient descent is commonly employed. The learned matrices U and V capture latent factors, providing insights into user and item preferences. Recommendations are then generated by multiplying the user matrix with the product of singular values and the transposed item matrix.

5.2 Neural Network for Playtime Prediction

The neural network employed for playtime prediction consists of three layers: two hidden layers with ReLU activation functions and an output layer with a linear activation function. The architecture is defined as follows:

$$\text{Input Layer} \rightarrow \text{Hidden Layer 1 (ReLU)} \rightarrow \text{Hidden Layer 2 (ReLU)} \rightarrow \text{Output Layer (Linear)}$$

The input to the network comprises user and game pairs, and the output is the predicted playtime. The neural network is trained using mean squared error as the loss function, and the Adam optimizer is employed for weight updates. The network learns to capture complex relationships between user and game features, enabling it to predict playtime for unseen user-game pairs.

Through backpropagation, the network adjusts its weights to minimize the difference between predicted and actual playtime values. The ReLU activation functions introduce non-linearity, allowing the model to capture intricate patterns within the data.

The training process involves iterating through epochs, with model checkpointing to save the best-performing weights. The neural network thus becomes a powerful tool for personalized playtime predictions, leveraging its ability to uncover intricate feature interactions within the user-game interaction matrix.

6 Experiments/Results/Discussion

6.1 Experimental Setup

Our experiments focus on evaluating the performance of two recommendation methodologies: Singular Value Decomposition (SVD) and a Neural Network for playtime prediction. The datasets utilized for training and evaluation are 'game_play.dat' and 'item_info.dat,' obtained from the Steam platform.

For the SVD model, the hyperparameters include the number of latent factors (k) and the learning rate (α). The neural network's hyperparameters encompass the learning rate, batch size, and the number of hidden layers and units.

To avoid overfitting, the data is split into training and test sets. Cross-validation is employed for model tuning, with five folds utilized to assess generalizability. We monitor mean squared error (MSE) as the primary evaluation metric for both models, given its suitability for regression tasks.

6.2 Results

6.2.1 Singular Value Decomposition (SVD)

The SVD model demonstrates strong performance with a root mean squared error of 270 on the test set. By capturing latent factors within the user-game interaction matrix, SVD successfully predicts playtime hours, providing accurate and personalized recommendations.

6.2.2 Neural Network for Playtime Prediction

The neural network achieves a root mean squared error of 243 on the test set. The ReLU activation functions in the hidden layers enable the model to capture intricate patterns within the data, contributing to its predictive accuracy.

6.3 Performance Metrics

Our primary metric, Mean Squared Error (MSE), is calculated as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where N is the number of test samples, y_i is the actual playtime, and $\hat{y}_i$ is the predicted playtime.

6.4 Discussion

6.4.1 Strengths and Limitations

SVD excels in capturing latent factors and provides robust recommendations, particularly in scenarios with sparse data. However, scalability can be an issue as the user and item base expands. The neural network, while scalable, demands substantial computational resources and data for effective training.

6.4.2 Overfitting

To mitigate overfitting, we incorporated regularization techniques, such as dropout in the neural network. Cross-validation and hyperparameter tuning were crucial in achieving models that generalize well to unseen data.

7 Conclusion/Future Work

7.1 Conclusion

In conclusion, our study delved into the evaluation of two recommendation methodologies: Singular Value Decomposition (SVD) and a Neural Network, for predicting playtime hours in a gaming platform. The findings highlight competitive performances from both models, with SVD excelling in capturing latent factors and providing robust recommendations. Meanwhile, the neural network demonstrated its prowess in capturing intricate patterns within the data. The comparative analysis revealed trade-offs between interpretability, scalability, and computational requirements.

The success of these models underscores the significance of collaborative filtering and deep learning approaches in delivering personalized gaming recommendations. SVD proved effective in scenarios with sparse data, offering accurate and interpretable predictions. The neural network, despite its computational demands, exhibited strong predictive accuracy.

7.2 Future Work

Looking ahead, increased computational resources could enable the exploration of more sophisticated neural network architectures and hyperparameter tuning. Ensemble methods that combine the strengths of collaborative filtering and deep learning could be investigated to further enhance recommendation accuracy. Moreover, the inclusion of user feedback and contextual information, such as user preferences and gaming genres, could contribute to a more holistic and personalized recommendation system.

8 Contributions

Each team member played a crucial role in the successful completion of the project, contributing expertise in various areas to achieve a comprehensive and effective outcome.

Makris Fragkiskos: Led the implementation of the Singular Value Decomposition (SVD) recommendation model, focusing on model tuning and optimization. Conducted experiments to assess the performance of SVD and contributed to the writing of the Methods and Experiments/Results/Discussion sections.

Giannios Panagiotis: Implemented and fine-tuned the Neural Network for playtime prediction, focusing on hyperparameter optimization and model evaluation. Conducted experiments to analyze the neural network's performance and contributed to the writing of the Methods and Experiments/Results/Discussion sections.

Gkolias Theodoros: Led the literature review, researching existing papers and methodologies related to recommendation systems. Contributed to the discussion on the strengths and weaknesses of different approaches. Contributed to the writing of the Literature Review and Experiments/Results/Discussion sections.

The collaborative efforts of each team member were instrumental in the project's success. Regular communication and coordination facilitated a synergistic workflow, where individual strengths were leveraged to address various aspects of the recommendation system implementation and evaluation.

9 References/Bibliography

1. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30-37.

2. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173-182).

3. Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence* (pp. 452-461).

4. Zhang, A., Lipton, Z., Li, M., & Smola, A. J. (2023). Dive into Deep Learning. `https://d2l.ai/`

5. Item_info.dat, Users_info.dat, Game_purchase.dat, Game_play.dat datasets from Steam platform. Retrieved from `https://github.com/caserec/Datasets-for-Recommender-Systems/tree/master/Processed%20Datasets/Steam`.

6. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct), 2825-2830. Retrieved from `https://scikit-learn.org/`

7. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from `tensorflow.org`.

8. Chollet, F. (2015). Keras. In *Dive into Deep Learning*. Retrieved from `https://keras.io/`

9. McKinney, W. (2010). Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference* (pp. 51-56). Retrieved from `https://pandas.pydata.org/`

10. Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357-362. Retrieved from `https://numpy.org/`

11. Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3), 90-95. Retrieved from `https://matplotlib.org/`

12. Jones, E., Oliphant, T., Peterson, P., & others. (2001). SciPy: Open source scientific tools for Python. Retrieved from `https://www.scipy.org/`