Book Recommendation System for Amazon Books

Name: Yixiao Wang, NetID: yw676

1 Background and Challenges in Book Recommendation Systems

Recommendation systems have evolved significantly over the last three decades. The first personalized recommendation algorithms appeared in 1994, as discussed in [6]. In 1998, Amazon launched item-based collaborative filtering, revolutionizing personalized recommendations by scaling to millions of customers and items. This method, detailed in [4], became foundational for most widely used recommendation in various companies. Further insights into the development and 20 years of recommendation history at Amazon can be found in [7]. Later advancements, such as Probabilistic Matrix Factorization (PMF), proposed in [5], addressed sparsity using low-dimensional factor models. In recent years, despite innovations in deep learning, item-to-item collaborative filtering has remained a robust benchmark at Amazon, often outperforming early neural models as highlighted in [2]. More recently, advanced techniques like Amazon DSSTNE (Deep Scalable Sparse Tensor Network Engine) [1] and attention-based mechanisms have emerged, offering deeper personalization and improved scalability.

In this project, we choose book recommendation system as our research object, which present unique challenges. First, books exhibit significant variability in user preferences based on demographics such as age, gender, nationality, and religion. Second, the vast catalog and sparse interactions with specific titles hinder meaningful pattern discovery. Third, the temporal nature of reading introduces delays in feedback, extending purchasing cycles compared to products like movies or games. Fourth, books are heavily influenced by macro-level factors such as political changes, global events, and economic trends, creating era-driven demand distinct from seasonal patterns. Lastly, bundling and versioning present complexities: series like "Harry Potter" or textbooks are often purchased in bulk, and traditional similarity-based algorithms struggle to differentiate between editions or identify sequels effectively.

This final project explores Amazon's latest pre-trained recommender system [3], focusing on leveraging demographic insights, addressing temporal dynamics, and refining book relationships to enhance recommendation quality and increase revenue.

2 Improvement of Pre-trained Recommender Systems

2.1 Pre-trained Recommender Model Overview

Pre-trained recommender system[3] developed by Ziqian L, Hao D et al. in Amazon leverage large-scale datasets to integrate domain-specific and user-specific knowledge, addressing challenges like data sparsity, domain heterogeneity, and cold-start problems. These models combine pre-training on existing data with fine-tuning for specific tasks.

$D_k \in \mathbb{R}^B$ represents the latent properties of domain k, acting as a confounder that influences user embeddings U_i, item embeddings V_j, and interactions R_{ijk}. $U_i \in \mathbb{R}^B$ captures the interests of user i, while $H_i \in \mathbb{R}^{N_u \times B}$ represents the interaction history of user i in domain k. Item j's popularity is encapsulated by $F_j \in \mathbb{R}^C$, and $X_j \in \mathbb{R}^B$ provides the description of item j. The popularity representation of item j, denoted by $Z_j \in \mathbb{R}^B$, acts as a confounder affecting both $V_j \in \mathbb{R}^B$, the overall properties of item j, and interactions R_{ijk}.

The conditional probability of the observed interactions is defined as:

$$P(R_{ijk} \mid U_i, V_j, D_k, Z_j) = f_{\text{softmax}}(U_i^\top V_j + D_k W_d + Z_j W_z)$$

The optimization objective minimizes the negative log-likelihood (NLL) of observed interactions, alongside regularization terms for latent variables:

$$\begin{aligned}\mathcal{L} = &- \sum_{k=1}^{K_s} \sum_{i=1}^{I_k} \sum_{j=1}^{J_k} R_{ijk} \log \left(f_{\text{softmax}}(U_i^\top V_j + D_k W_d + Z_j W_z) \right) \\ &+ \frac{\lambda_z}{2} \sum_{j=1}^{J_k} \|Z_j - f_{\text{pop}}(F_j)\|^2 + \frac{\lambda_d}{2} \sum_{k=1}^{K_s} \|D_k\|^2 \\ &+ \frac{\lambda_v}{2} \sum_{k=1}^{K_s} \sum_{j=1}^{J_k} \|V_j - f_{\text{e}}(D_k, Z_j)\|^2 + \frac{\lambda_u}{2} \sum_{k=1}^{K_s} \sum_{i=1}^{I_k} \|U_i - f_{\text{seq}}(D_k, H_i)\|^2\end{aligned}$$

Here, W_d and W_z are trainable weights, and $\lambda_z, \lambda_d, \lambda_v, \lambda_u$ are regularization parameters. Functions $f_{\text{softmax}}, f_{\text{seq}}, f_{\text{pop}}$, and f_{e} are learnable components that compute interaction probabilities, user histories, item popularity, and embeddings, respectively. This formulation integrates user-item interactions, domain knowledge, and item popularity for robust recommendations.

2.2 Improvement of Pre-trained Recommender Systems

This section introduces two novel modules to enhance the Pre-trained Recommender Systems, The causal graphical model is in Figure 1.

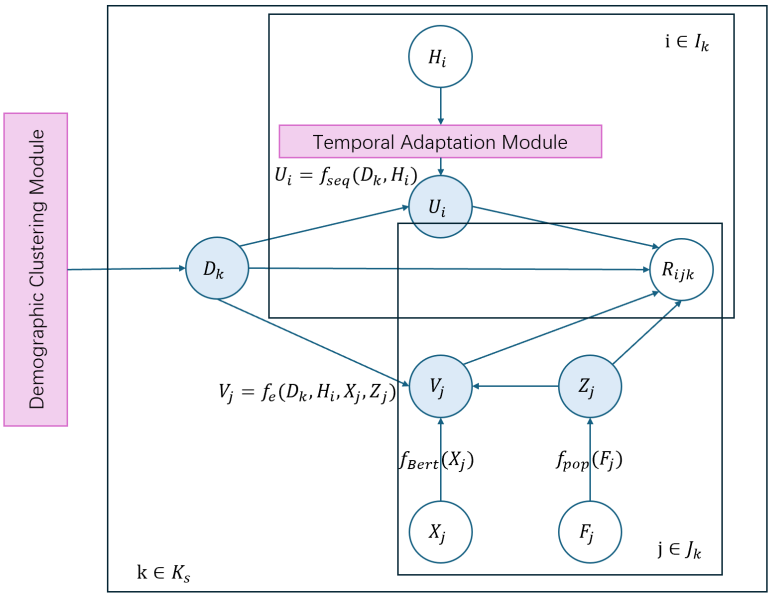

Figure 1: The pink modules represent the newly added components, the transparent circles indicate observed variables, and the blue circles denote latent variables.

2.2.1 Demographic Clustering Module

To replace the static domain knowledge used in the original model, we introduce a demographic clustering module. By applying clustering techniques to user demographic features such as region, age, or income level, this module captures nuanced patterns in user preferences. Compared to static domain-specific rules, this dynamic grouping improves the model's ability to personalize recommendations and adapt to diverse user behaviors.

2.2.2 Temporal Adaptation Module

To address the evolving nature of user interests, we incorporate a temporal adaptation module. This module leverages historical truncation, discarding user interaction data beyond a predefined timeframe (e.g., 12 months). The truncation ensures recommendations are based on recent and relevant behaviors, avoiding influence from outdated interactions.

Inspired by natural language processing, where word probability in sequences diminishes with distance, historical truncation reflects the decreasing relevance of older interactions. Alternatively, an adaptive weighting mechanism, such as Exponential Decay, dynamically prioritizes recent interactions by reducing the importance of older data over time. This ensures a balanced consideration of both recent and historical preferences, enhancing the system's responsiveness to changing user interests.

2.3 Details of Demographic Clustering Module

This section outlines the challenges and steps involved in implementing the demographic clustering module to enhance recommendation systems.

Steps for Implementation:

1. Data Preparation: Proper data preprocessing, including handling missing data, is crucial. Feature engineering tailored to the dataset enhances clustering outcomes.

2. Outlier Detection: Employ the Isolation Forest algorithm for efficient anomaly detection especially for multi-dimensional data.

3. Feature Scaling: Normalize or standardize features to ensure consistency. Apply log transformations for features with large ranges firstly (e.g., spending records).

4. Correlation Analysis: Check for multicollinearity to prevent redundant or misleading patterns in clustering. Use Principal Component Analysis (PCA) to reduce dimensionality and neutralize correlations while retaining essential variance.

5. Clustering Algorithm: Apply KMeans clustering with an optimal number of clusters (K) determined through techniques such as the elbow method or silhouette analysis. PCA-reduced features improve computational efficiency and clustering quality. Followed by the collaborative filter model, we suggest Cosine Similarity as the distance measure.

6. Evaluation Metrics: Use metrics such as the Silhouette Score to evaluate cluster quality. Subsample the dataset for approximate assessments, employing stratified sampling to ensure representative clusters.

By addressing these challenges and following this structured approach, demographic clustering provides a robust foundation for personalized recommendation systems. After finish the trained model. We can deal with the cold-start problems for new users, also we can update the module periodically.

3 Is the Change Worth It? And Some Broader Considerations

Although the development of new approaches, especially in deep learning and attention mechanisms, has gained momentum, Amazon's recommendation algorithms are still primarily rooted in traditional Collaborative Filtering (CF).

For a company with such a vast user base and catalog—over 310 million users and 40 million books (ebooks and print)—computational complexity is a critical consideration. Transitioning from CF to an attention mechanism-based model would involve significant deployment costs. However, the demographic clustering strategy and temporal adaptation techniques proposed in our work can be seamlessly integrated into CF, ensuring their continued relevance.

From a business perspective, increasing revenue remains the ultimate goal. This raises important questions:

- How often will the recommended items differ between algorithms like FPGrowth and Amazon's current recommendation system?
- How much additional revenue can be gained through algorithmic improvements?
- How about investing in logistics and warehouse management instead of improving the recommendation algorithm?

An alternate perspective involves leveraging recommendations as an advertising platform. By enabling publishers or authors to pay for prioritized recommendations, Amazon could integrate investment mechanisms into its recommendation framework. This could provide a dual revenue stream—enhancing user satisfaction and monetizing the recommendation process.

To mitigate potential echo chamber effects, introducing controlled randomness into recommendations may also prove beneficial. Suggesting items that differ significantly from a user's typical preferences could help users explore new interests and enhance engagement. These broader considerations highlight the balance between algorithmic sophistication, business goals, and user experience in large-scale recommendation systems.

References

[1] AMAZON, 2019. Amazon dsstne: Deep scalable sparse tensor network engine [EB/OL]. https://github.com/amazon-archives/amazon-dsstne.

[2] HARDESTY L, 2019. The history of amazon's recommendation algorithm: Collaborative filtering and beyond[J/OL]. Amazon Science. https://www.amazon.science/the-history-of-amazons-recommendation-algorithm.

[3] LIN Z, DING H, HOANG N T, et al., 2023. Pre-trained recommender systems: A causal debiasing perspective[A/OL]. https://doi.org/10.48550/arXiv.2310.19251.

[4] LINDEN G, SMITH B, YORK J, 2003. Amazon.com recommendations: item-to-item collaborative filtering[J/OL]. IEEE Internet Computing, 7(1): 76-80. DOI: 10.1109/MIC.2003.1167344.

[5] MNIH A, SALAKHUTDINOV R R, 2007. Probabilistic matrix factorization[C/OL]// PLATT J, KOLLER D, SINGER Y, et al. Advances in Neural Information Processing Systems: Vol. 20. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2007/file/d7322ed717dedf1eb4e6e52a37ea7bcd-Paper.pdf.

[6] RESNICK P, IACOVOU N, SUCHAK M, et al., 1994. Grouplens: an open architecture for collaborative filtering of netnews[C/OL]//175-186. DOI: 10.1145/192844.192905.

[7] SMITH B, LINDEN G, 2017. Two decades of recommender systems at amazon.com [J/OL]. IEEE Internet Computing, 21(3): 12-18. DOI: 10.1109/MIC.2017.72.