VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY

UNIVERSITY OF SCIENCE

FACULTY OF INFORMATION TECHNOLOGY

REPORT

Topic: RECOMMENDER SYSTEM

Course: Computational Thinking

Students:

Nguyễn Thanh Trúc (24127137)

Đào Minh Khoa (24127422)

Trần Lưu Gia Bảo (24127018)

Nguyễn Tiến Cường (24127337)

Nguyễn Khánh Toàn (24127252)

Phạm Trần Anh Quân (24127226)

Supervisors:

PhD. Nguyễn Tiến Huy

PhD. Lê Thanh Tùng

Mr. Trần Hoàng Quân

Ngày 10 tháng 11 năm 2025

Mục lục

Danh sách hình vẽ

Danh sách bảng

1 Abstract

This article presents a comprehensive analysis of recommendation systems, examining their fundamental components, algorithmic approaches, and practical applications. We provide detailed comparisons of different techniques, focusing on their mathematical foundations, input-output characteristics, and real-world performance. The analysis covers traditional approaches, modern deep learning methods, and emerging hybrid systems, offering insights into their strengths, limitations, and suitable application domains.

2 Introduction

2.1 Fundamentals of Recommendation Systems

A recommendation system is an information filtering technology that predicts users' preferences and suggests relevant items from large item spaces. These systems are fundamental to modern digital services, addressing information overload by personalizing content discovery across e-commerce, entertainment, social media, and professional networks.

3 Analysis of Personalized Recommender Systems

[?] [?]

3.1 Content-Based Filtering

This method recommends items based on a user's explicit preferences and the features/attributes of the items the user liked in the past.

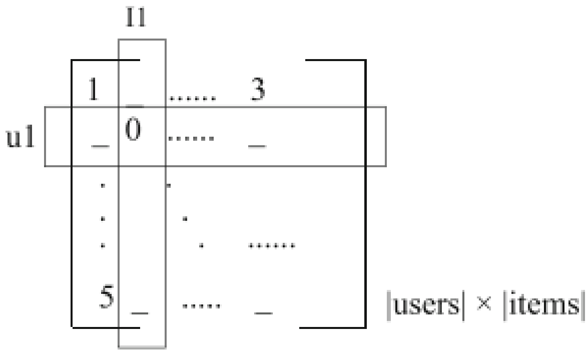
Advantage: It does not require data from other users, is not prone to data sparsity or cold-start problems, and can recommend new items not yet rated by many users.

3.2 Collaborative Filtering (CF)

Driven by user profiles or historical interactions

3.2.1 Memory-Based CF

This method get input is a matrix of user-item interactions (e.g., ratings, clicks, purchases).



Hình 1: Memory-Based Collaborative Filtering

Figure 1 illustrates the user-item interaction matrix, where rows represent users and columns represent items. The values in the matrix indicate the interactions (e.g., ratings) between users and items. Memory-based collaborative filtering utilizes this matrix to find similar users or items based on their interactions.

In the memory-based approaches, the closest (similar) users or items are calculated by cosine similarity or pearson correlation coefficient.

This method is best for small-to-medium-sized datasets and is highly valued when explainability is necessary, as the prediction is easily traceable to the neighbors' ratings.

Drawback: Explicitly contrast the complexity: The computational cost is $O(M^2)$ for User-Based or $O(N^2)$ for Item-Based in the worst case (M users, N items), making it unscalable for massive, real-time catalogs.

A general solution to sparsity and high-dimensionality involves transforming high-dimensional sparse vectors into low-dimensional dense ones, a process known as feature embedding or representation learning.

- **User-based:** Measures similarity between the target user and all the other users.
- **Item-based:** Measures the similarity between the item that a target user is interested in and all the other items.

In both case, if similarity above a threshold, it then categorized as neighbor. The output of memory-based CF is a prediction, $\hat{r}_{u,i}$, for an unrated item i by a target user u, typically derived as a weighted average of neighbor ratings.

3.2.2 Model-Based CF

Uses machine learning, deep learning, or data mining algorithms to build a predictive model for recommendations. Able to capture complex relationships because the model predicts the user's rating for unrated items in the model-based collaborative filtering framework.
Output: Latent factor vectors for users and items. The final score is the dot product of these vectors.

Comparison: MF solves the sparsity problem by inferring hidden preferences and improves scalability by reducing dimensionality. However, it often struggles to model non-linear relationships or integrate side features easily.

4 Classification of recommender systems

- **Clustering-Based Models:** Use unsupervised learning like K-Nearest Neighbor (KNN) or Locality-Sensitive Hashing (LSH) to group similar users or items, making the system more scalable.
- **Matrix Factorization (MF)-Based Models:** Decompose the user-item utility matrix into lower-dimensional matrices representing latent factors (e.g., SVD, PMF, NMF) to uncover dependencies and user-item structures.
- **Deep Learning-Based Models:** Utilize neural networks (e.g., Autoencoders, CNNs, RNNs) to learn complex patterns in user-item interactions, capturing non-linear relationships for improved recommendations.

5 Deep Learning-Based Recommendation Models

Deep Neural Networks (DNNs) have revolutionized recommendation systems by enabling the modeling of complex, non-linear user-item relationships. These models excel at:

- **Feature Learning:** Automatic extraction of high-level features from raw inputs
- **Multimodal Fusion:** Integration of heterogeneous data sources (text, images, user behavior)
- **Sequential Patterns:** Capturing temporal dynamics and evolution of user preferences
- **Cross-Domain Transfer:** Leveraging knowledge from related domains

5.1 Input Types and Preprocessing

Deep learning models typically accept:

- **Dense Features:** Normalized continuous variables (age, price)
- **Sparse Features:** One-hot or multi-hot encoded categorical variables
- **Sequential Data:** Session logs, interaction histories
- **Unstructured Data:** Text descriptions, images, audio

5.2 Output Formats

Models produce various outputs depending on the task:

- **Rating Prediction:** Scalar values indicating predicted user preference
- **Click-Through Rate (CTR):** Probability of user interaction
- **Ranking Scores:** Used for ordering candidate items
- **Embeddings:** Dense vector representations for retrieval

Deep learning-based recommendation models include:

6 Other Recommendation Approaches

6.1 Explainable Recommender Systems (XRS)

XRS focus on providing transparency and justification for recommendations, enabling users to understand why items are recommended.

- **Input:** User-item interactions plus auxiliary information for generating explanations
- **Output:** Recommendations with natural language or structured explanations
- **Best suited for:** High-stakes decisions, regulated industries, trust-sensitive domains
- **Trade-off:** Some predictive accuracy is typically sacrificed for interpretability

Model	Core mechanism	Input	Drawback
Multilayer perceptron-based recommendation	Feed-forward neural networks with multiple hidden layers	User and item features, interaction data	High computational cost, requires large training data
Autoencoders	Encoding and decoding input data to learn efficient representations	User-item interaction matrices	Information loss during compression, sensitive to noise
Convolutional Neural Networks (CNNs)	Feature extraction through convolution operations	Image data, user profiles	Limited to structured data, computationally intensive
Recurrent Neural Networks (RNNs)	Sequential data processing with memory cells	Temporal user behavior data	Gradient vanishing/exploding, slow training
Restricted Boltzmann Machines	Probability distribution learning over inputs	User-item interaction patterns	Complex training process, difficult to tune
Deep Reinforcement Learning	Optimization based on user feedback	User interactions and feedback over time	Delayed rewards, exploration vs exploitation trade-off
Adversarial Networks	Generative modeling with competing networks	User-item interactions	Training instability, mode collapse risk
Graph Neural Networks	Graph structure processing	User-item relationship graphs	Scalability issues with large graphs, computational complexity

Bảng 1: Deep Learning-Based Recommendation Models

6.2 Context-Aware Recommender Systems

These systems incorporate dynamic contextual signals to improve recommendation relevance.

- **Input:** Traditional user-item data plus contextual features (time, location, weather, device)
- **Output:** Context-specific recommendations or rankings
- **Best suited for:** Point-of-Interest (POI), mobile apps, real-time recommendations
- **Advantage:** Significantly improved relevance compared to static approaches

6.3 Knowledge-Based Systems

Particularly effective for high-consideration, infrequent purchases where historical data is sparse.

- **Main approaches:**
  - **Knowledge Graph Embedding (KGE):** Encodes domain knowledge into dense vector representations
  - **Path-Based:** Uses meta-paths in knowledge graphs for semantic similarity
- **Best suited for:** Complex products (cars, real estate), professional services
- **Advantage:** Can handle cold-start cases effectively using domain knowledge

6.4 Specialized Domain Recommenders

- **Demographic RS:**
  - Uses population segment data for quick but generic recommendations
  - Effective for cold-start but less personalized
- **Reciprocal RS:**
  - For two-sided matching (dating, job matching)
  - Requires mutual satisfaction of both parties
  - Complex optimization to balance competing preferences
- **Group RS:**

- For shared experiences (travel planning, movie watching)
- Uses preference aggregation strategies
- Challenges: Fairness, satisfaction distribution

7 Challenges

7.1 Traditional Challenges

- **Cold Start:**
  - Problem: No historical data for new users/items
  - Solutions: Hybrid approaches, content-based features, quick surveys
  - Impact: Most severe in CF systems
- **Data Sparsity:**
  - Problem: Very few interactions per user/item
  - Solutions: Matrix factorization, transfer learning
  - Impact: Affects recommendation quality and coverage
- **Scalability:**
  - Problem: Large-scale real-time recommendations
  - Solutions: Two-stage (retrieval + ranking), ANN search
  - Impact: Trade-off between accuracy and latency

7.2 Current Challenges

- **System Security:**
  - Shilling attacks: Fake profiles/ratings
  - Profile injection attacks
  - Adversarial perturbations
  - Solutions: Robust optimization, anomaly detection
- **Fairness and Bias:**

  - Population bias in training data
  - Filter bubbles and echo chambers
  - Long-tail item exposure
  - Solutions: Diversity metrics, fairness constraints

- **Privacy and Trust:**
  - Data minimization requirements
  - Federated learning approaches
  - Explainable recommendations
  - Solutions: Privacy-preserving ML, local differential privacy

8 Similarity Measures in memory-based CF recommendation systems

Similarity measures are fundamental to certain recommendation approaches, particularly memory-based collaborative filtering, where they are used to identify neighboring users or items. The most fundamental of these are the Pearson Correlation Coefficient and Cosine Similarity. Pearson Correlation Coefficient (PCC): This measures the linear correlation between the ratings of two users (u and v) on their co-rated items (Iu,v), defined as:

$$\mathrm{PCC}(u,v) = \frac{\sum_{i\in I_{u,v}}(r_{u,i}-\bar{r}_u)(r_{v,i}-\bar{r}_v)}{\sqrt{\sum_{i\in I_{u,v}}(r_{u,i}-\bar{r}_u)^2}\sqrt{\sum_{i\in I_{u,v}}(r_{v,i}-\bar{r}_v)^2}}$$

Cosine Similarity (COS): This measures the cosine of the angle between two user rating vectors in the item space, defined

$$\cos(\mathbf{u},\mathbf{v}) = \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}}$$

9 Conclusion

This comprehensive analysis of recommendation systems has examined the theoretical foundations, practical implementations, and emerging trends in the field. We have:

- Detailed the mathematical frameworks and algorithms underlying different recommendation approaches
- Analyzed the specific inputs, outputs, and data requirements for each model family
- Compared the strengths and limitations of various techniques across different application domains
- Identified key challenges and promising future research directions

The field continues to evolve, with deep learning and knowledge-based approaches offering new capabilities while raising important questions about scalability, privacy, and fairness. Success in modern recommendation systems requires carefully balancing these competing concerns while selecting appropriate techniques for specific application requirements.

Tài liệu

[1] Chintoo Kumar, C. Ravindranath Chowdary, and Ashok Kumar Meena. Recent trends in recommender systems: a survey. *International Journal of Multimedia Information Retrieval*, 13, 2024.

[2] Yang Li, Kangbo Liu, Ranjan Satapathy, Suhang Wang, and Erik Cambria. Recent developments in recommender systems: A survey [review article]. *IEEE Computational Intelligence Magazine*, 19(2):78–95, 2024.