A Comparative Analysis of Modeling Paradigms and Techniques in Sequential Recommendation

Yuxiang Jia

Aquinas International Academy, Wuhan City, 430000, China

3191434453@qq.com

Abstract. The evolution of sequential recommendation systems is marked by significant advancements, particularly in capturing user preferences from sequential data. Transformer-based models, despite their success, grapple with efficiency in processing long sequences. This study conducts a thorough comparative analysis of these models, highlighting the tradeoffs between efficiency and effectiveness across various modeling approaches and techniques. Research is categorized into Transformer-based, RNN-based, and the nascent SSM paradigm, examining their capacity for modeling complex sequences and their computational demands. Selective State Space Models receive special focus due to their potential in achieving a balance between performance and speed. The discussion progresses to specific sequential modeling techniques like attention mechanisms, memory architectures, and methods for managing long-term dependencies. The application of these techniques within different frameworks and their effects on performance and efficiency are analyzed. Additionally, the study evaluates how user data types (explicit and implicit) and recommendation tasks shape model development. It also considers dataset attributes like sequence length and sparsity, and their influence on the complexity and efficiency of models.

Keywords: Sequential Recommendation, Transformers, State Space Models (SSM), Attention Mechanisms, Recurrent Neural Networks (RNN)

1 Introduction

Recommender systems have become an indispensable part of modern online platforms, guiding users towards relevant products, content, and services amidst an overwhelming abundance of choices. Sequential recommendation systems form a specialized class of recommenders designed to model the evolution of user preferences based on their sequential interactions. These systems play a crucial role in numerous applications: 1) E-commerce: Suggesting items that naturally complement past purchases or fit a user's evolving shopping patterns; 2) Streaming Platforms: Recommending movies, TV shows, or music tracks that align with a user's viewing or listening history; 3) News and Content Aggregation: Presenting news articles or social media posts that match a user's interests based on their past reading behaviors [1].

Z. Wang et al. (eds.), *Proceedings of the 4th International Conference on Economic Development and Business Culture (ICEDBC 2024)*, Advances in Economics, Business and Management Research 299,
https://doi.org/10.2991/978-94-6463-538-6_39

Unlike traditional recommendation approaches that treat user interactions as independent events, sequential recommendation systems explicitly consider the temporal order of interactions [7]. By capturing patterns within these interaction sequences, they can: 1) Uncover Dynamic Preferences: Identify shifting interests and short-term trends that might be missed by static models; 2) Predict Future Actions: Anticipate the next item a user is likely to interact with, given their recent activity; 3) Offer Contextualized Suggestions: Generate recommendations tailored to the user's current session or context.

The increasing complexity of sequential recommendation tasks poses a fundamental challenge: balancing effectiveness and efficiency. To achieve accurate and personalized recommendations, models must capture intricate long-range dependencies and complex relationships within lengthy interaction sequences. However, this often leads to computationally demanding models that struggle to provide real-time recommendations, especially in large-scale applications. Key Factors contributing to this tension include: 1) Long Interaction Sequences. Modern users generate vast amounts of interaction data over time. Processing these long sequences strains the computational resources of many modeling approaches; 2) Complex Sequential Patterns. User preferences are not static. Models must identify subtle shifts in interests, recurring patterns, and the influence of context, requiring sophisticated representations; 3) The Need for Real-time Recommendations.

The efficiency-effectiveness tradeoff has spurred active research towards developing new model paradigms and innovative techniques for sequential recommendation. This comparative study provides a comprehensive analysis of the state-of-the-art in sequential recommendation models, focusing on the tradeoffs between effectiveness and efficiency.

2 Modeling Paradigms

The Transformer architecture, originally introduced for natural language processing, has revolutionized sequential recommendation. At its core lies the self-attention mechanism, which allows the model to directly relate different positions within an input sequence, regardless of their distance. Self-attention helps pinpoint the most relevant elements of a user's interaction history for predicting their next interaction by calculating attention scores through the generation of query (Q), key (K), and value (V) vectors, and then computing a weighted sum of the value vectors. The key benefits include capturing long-range dependencies, enabling the model to establish connections between distant items, and allowing for parallelizable computations, making Transformers more efficient to train compared to RNNs. However, Transformer-based models face challenges with computational complexity due to the quadratic time complexity ($O(n^2)$) of self-attention, especially for long sequences. To mitigate this, research has developed strategies like sparse attention patterns, which restrict computations to specific neighborhoods within the sequence [3], adaptive computation time, which adjusts the number of computation steps based on the input [2], and

linearized attention, which approximates calculations using linear projections to lower complexity.

Recurrent Neural Networks (RNNs) have long been essential for sequential modeling due to their ability to maintain an evolving internal state that processes input sequences step-by-step, incorporating past interactions for predictions. Specialized architectures like Long Short-Term Memory (LSTM) ([10]) and Gated Recurrent Units (GRU) [9] address early RNNs' difficulties with long-term dependencies through gating mechanisms. LSTMs use three gates: Forget Gate, Input Gate, and Output Gate, to manage information flow, while GRUs use two: Reset Gate and Update Gate. GRUs are computationally less expensive but both are effective for sequential data. However, RNNs struggle with very long sequences due to vanishing/exploding gradients, where gradients become too small or too large during backpropagation, hindering learning. Mitigation techniques include gradient clipping, rescaling gradients to prevent explosions, and regularization methods like L1 or L2 to reduce overfitting. Additionally, skip connections, similar to residual connections in ResNets [8], allow information to bypass layers, improving gradient flow.

State Space Models (SSMs) offer a unique framework for modeling sequential and time-series data, gaining attention in sequential recommendation due to their efficiency and expressiveness. SSMs have computational advantages over RNNs, especially for long sequences. Linear SSMs, solvable analytically with methods like the Kalman filter [6], offer speed and efficiency, while non-linear SSMs use approximate inference techniques or specialized algorithms [3, 5]. Hierarchical SSMs capture both short-term and long-term dependencies within user preferences [4]. A recent advancement is selective SSMs, which update only relevant parts of the hidden state, reducing computational needs [3]. This approach enhances efficiency for real-time recommendations, with some models designed for optimization on specialized hardware [3].

3 Sequential Modeling Techniques

Attention mechanisms have revolutionized sequential modeling by enabling models to dynamically focus on the most relevant parts of an input sequence, a crucial capability for sequential recommendation where future actions depend on specific historical interactions. Common attention variants include: 1) Dot-product Attention: This foundational method computes similarity scores between the query (Q) of one item and the keys (K) of all items using dot products. The scores are normalized with a softmax function to obtain attention weights, which are then used to calculate a weighted sum of the value vectors (V); 2) Scaled Dot-product Attention: Introduced in the original Transformer paper, this scales the dot products by the square root of the key vector dimension to prevent softmax gradients from becoming too small in high-dimensional spaces; 3) Multi-head Attention: This powerful extension uses multiple parallel attention heads, each with its own query, key, and value projections [11, 12, 13].

Sequential models must efficiently store, recall, and update past information to understand evolving user preferences. Various memory architectures offer unique advantages and drawbacks. RNN-based memories, including traditional RNNs, LSTMs, and GRUs, maintain an internal hidden state that evolves with input sequences. They are computationally efficient for shorter sequences and allow dynamic state updates but struggle with long-term storage due to limited capacity and gradient problems. Conversely, models with external memory decouple information storage from computation, featuring a dedicated memory module for reading and writing. This increases long-term storage capacity and mitigates gradient issues but can slow performance due to overhead and complex read/write mechanisms. The choice of memory architecture significantly impacts a model's ability to handle long-range dependencies [14, 15].

Modeling complex relationships between items separated by many steps in a user's interaction history is crucial for accurate and personalized sequential recommendations. Transformers and State Space Models (SSMs) offer distinct strategies for this purpose. For Transformers, self-attention enables direct connections between all items in a sequence, making them adept at capturing long-range dependencies. To handle extreme sequence lengths, strategies like imposing sparsity patterns on the attention matrix reduce computations, approximating attention using linear projections to lower time complexity from quadratic to linear, and dynamically adjusting attention steps to focus on the most relevant parts of the input, saving computations.

4 The Influence of Data and Tasks

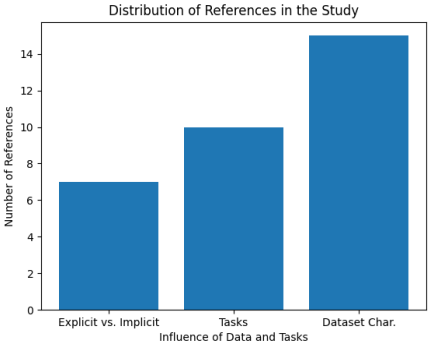

Fig. 1. Distribution of references in The Influence of Data and Tasks

Figure 1 shows the distribution of references across the different influence of data and tasks. The type of user data available to a recommender system profoundly influences model choices and strategies for capturing user preferences. Explicit behavior includes directly observable actions such as clicks, purchases, ratings, or reviews, which provide clear signals of user interest and are often represented numerically or with binary values. In contrast, implicit feedback involves indirect inferences like brows-

ing history, time spent on an item, search queries, or mouse movements, which are noisy and ambiguous, requiring nuanced interpretation and often transformation into confidence scores or binary indicators of potential interest. The distinction between explicit and implicit feedback significantly impacts the complexity of sequential recommendation.

Recommendation systems vary in design based on specific tasks, such as next-item prediction, session-based recommendation, and sequence completion. For predicting the next item a user will interact with, models typically leverage the entire interaction history using approaches like Transformers or Sequential Sparse Models (SSMs), optimizing for metrics such as accuracy or Mean Reciprocal Rank (MRR). In session-based recommendation, where long-term user history may be less relevant, Recurrent Neural Networks (RNNs), particularly GRUs, excel at capturing short-term dynamics within sessions. Privacy-preserving techniques are crucial here when user identities need to be anonymized. For sequence completion tasks like playlist generation or future purchase prediction, generative models such as Transformers with autoregressive decoding or specialized RNN architectures are employed to predict multiple future items accurately. These models adapt to evolving user preferences by emphasizing recent interactions, reflecting the dynamic nature of user interests within a session.

The effectiveness of a sequential recommendation model hinges not just on its architecture but also on the specific characteristics of the dataset it is trained on. Sequence length plays a critical role: for short sequences, standard RNNs like LSTMs or GRUs are effective, while Transformers excel with moderate to long sequences due to their ability to capture complex dependencies. Sparsity in datasets, where users interact with only a small fraction of available items, poses challenges in learning robust representations. Domain-specific knowledge, such as hierarchical structures in e-commerce taxonomies, informs model design choices. These dataset characteristics interact synergistically, influencing model selection: handling long sequences amidst sparsity requires efficient techniques like Selective State Space Models (SSMs), while domain-specific tasks like session-based recommendations in music benefit from models that integrate both sequential patterns and item features like audio data [3, 4].

5 Conclusion

This comparative study delved into the tradeoffs, techniques, and complexities inherent in sequential recommendation, highlighting key insights: the challenge between model expressiveness and computational efficiency, especially for long interaction sequences; the evolution of modeling paradigms such as Transformers, RNNs, and State Space Models (SSMs) with their unique strengths and limitations; and various sequential modeling techniques like attention mechanisms and memory architectures.

References

1. Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. arXiv preprint arXiv:1607.06450 (2016).
2. Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural networks 107 (2018), 3–11.
3. Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023).
4. Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. 2020. Hippo: Recurrent memory with optimal polynomial projections. Advances in neural information processing systems 33 (2020), 1474–1487.
5. Albert Gu, Karan Goel, and Christopher Ré. 2021. Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396 (2021).
6. Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. 2021. Combining recurrent, convolutional, and continuous-time models with linear state space layers. Advances in neural information processing systems 34 (2021), 572–585.
7. F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis) 5, 4 (2015), 1–19.
8. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition . 770–778.
9. Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016).
10. Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015).
11. Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In 2018 IEEE international conference on data mining (ICDM) . IEEE, 197–206.
12. Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
13. James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al . 2017. Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences 114, 13 (2017), 3521– 3526.
14. Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. 1419–1428.
15. Jianghao Lin, Yanru Qu, Wei Guo, Xinyi Dai, Ruiming Tang, Yong Yu, and Weinan Zhang. 2023. MAP: A Model-agnostic Pretraining Framework for Click-through Rate Prediction. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 1384–1395.