EXPLORING STATE SPACE MODELS' APPLICATION TO RECOMMENDER SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recommender systems aim to estimate the dynamically changing user preferences and sequential dependencies between historical user behaviours and metadata. Although transformer-based models have proven to be effective in sequential recommendations, their state growth is proportional to the length of the sequence that is being processed, which makes them expensive in terms of memory and inference costs. Also, as proposed by Chengkai Liu et al. (2024), state space models (SSMs) can achieve SOTA results in the sequential recommendations domain with lower latency, memory, and inference costs. In our research, we explored various SSM applications in the domain of recommender systems, such as Mamba blocks and Mamba-based backbones, including the recently introduced Mamba2 architecture, and compared them with algorithms that are widely used in the field.

1 INTRODUCTION

1.1 RELATED STUDY

1.2 SEQUENTIAL RECOMMENDATION TASK DEFINITION

Consider user set $U = \{u_1, u_2, ..., u_N\}$, item set $V = \{v_1, v_2, ..., v_K\}$ and $S_u = \{v_1, v_2, ..., v_{nu}\}$ the chronologically ordered interaction sequence for user $u \in U$, where n_u is the the length of the sequence. Given S_u the task is to predict the next interacted item, v_{n_u+1}.

1.2.1 TRANSFORMERS

Recently, transformer models have been shown to be effective in sequential recommendation tasks as the backbone of larger models Wang-Cheng Kang (2018) **?** and as individual LLMs Jinming Li (2023) Zhenrui Yue (2023). Despite their success, attention-based methods face inference inefficiencies due to the quadratic computational complexity inherent in attention operators and their rapid state growth, that is proportional to sequence length. Also, without special mechanisms **?**, transformers can't handle long contexts and, consequently, long user histories.

1.2.2 STATE SPACE MODELS

State Space Model (SSM) **?** is a recent framework for sequence modelling defined by linear ordinary differential equations:

$$h'(t) = Ah(t) + Bx(t)$$
$$y(t) = Ch(t)$$

Where A, B, C are learnable matrices, $h(t)$ is the latent space, $x(t)$ is the input sequence and $y(t)$ is the output sequence. To compute sequence-to-sequence transformations efficiently, the matrix A must be *structured*, so structured SSMs have been introduced **?**. A general form of a structured SSM is defined by the equations:

$$h_t = Ah_{t-1} + Bx_t$$
$$y_t = Ch_t$$

Where $A \in \mathbb{R}^{(N,N)}, B \in \mathbb{R}^{(N,1)}, C \in \mathbb{R}^{(N,1)}$ They map a 1-dimensional sequence $x \in \mathbb{R}^T \rightarrow y \in \mathbb{R}^T$ through an implicit latent state $h \in \mathbb{R}^{(T,N)}$. To operate directly on sequences, the discretisation rule is applied - (f_A, f_B) to continuous parameters $(\Delta, \mathring{A}, \mathring{B})$, by $A = f_A(\Delta, \mathring{A})$ and $B = f_B(\Delta, \mathring{B})$, where Δ is the parameterised step size.

1.2.3 MAMBA BLOCK

In order to adaptively focus on relevant information while filtering out noise, the Mamba block Albert Gu (2024) introduces an extension to structured SSMs by adding a data-dependent selection mechanism. An important feature of Selective SSMs is their ability to be computed efficiently on the GPU using kernel fusion, parallel scanning and recomputation mechanisms.

In contrast to transformers' $O(n^2 \cdot d)$, SSMs provide $O(n \cdot d^2)$ complexity, which makes them a more efficient alternative, especially when operating on long sequences (Dao & Gu (2024)) .

1.2.4 MAMBA APPLICATIONS TO SEQUENTIAL RECOMMENDATIONS

Several applications of Mambas selective SSMs to recommender systems have been introduced. Chengkai Liu et al. (2024), **?**, **?**. But in previous works Mamba was only considered to be a part of a larger architecture. As Mamba has already shown efficient results in different areas **? ? ?**, we propose a more complex exploration of Mamba applications to recommender systems, including the use of Mambas as separate LLMs.

2 OUR RESULTS

2.1 HYDRA LAYER

To apply the sequence-to-sequence potential of the Hydra block, we provide a custom Hydra layer that combines the Hydra block with a standard feed-forward network. The main part of our standard architecture consists of Hydra layers. We have found that Hydra layer works more effectively than Mamba layers. Then we use the same PFFN that was introduced in Mamba4Rec.

$$PFFN(H) = GELU(HW^{(1)} + h^{(1)})W^{(2)} + b^{(2)}$$

Where $W^{(1)} \in \mathbb{R}^{D \times 4D}$, $W^{(2)} \in \mathbb{R}^{D \times 4D}$, $b^{(1)} \in \mathbb{R}^D$ are parameters of two dense layers and we use GELU **?** activation.

2.2 PREDICTION LAYER

Prediction layer is adopted from SASRec and Mamba4Rec, last item embedding is used to generate final prediction scores:

$$\hat{y} = Softmax(hE^T) \in \mathbb{R}^{|V|}$$

where $h \in \mathbb{R}^D$ is the last item embedding from the Hydra layer. $\hat{y} \in \mathbb{R}^{|V|}$ represents the probability distribution over the next item in the item set V.

2.3 BENCHMARKS

We tested our models on 3 benchmarks: Amazon Reviews '23 Beauty and Personal care, Amazon Reviews '23 Video Games and MovieLens-1M.

Datasets

Table 1: Amazon Reviews '23 Beauty and Personal care

Model	HT@10	NDCG@10	MRR@10	Latency	# Parameters	# Parameters w/o Embeddings
SASRec	0.048	0.028	0.022	0.117	13,636,224	100,096
Mamba4Rec	0.046	0.027	0.028	0.078	13,605,184	78,624
MamRec	0.031	0.020	0.017	2.51	130M	
GPT4Rec	0.030	0.025	0.015	2.32	117M	
2Mamba4Rec	0.048	0.030	0.025	0.096	13,605,184	78,624
Hydra4Rec	0.070	**0.043**	0.035	**0.0005**	842694	
LlamaRec	**0.093**	0.040	**0.040**	-	7B	

Table 2: Amazon Reviews '23 Video Games

Model	HT@10	NDCG@10	MRR@10	Latency	# Parameters	# Parameters w/o Embeddings
SASRec	0.119	0.073	0.059	0.129	1,790,016	100,096
Mamba4Rec	0.107	0.061	0.048	0.077	1,765,344	78,624
MamRec	0.083	0.033	0.025	2.51	130M	
GPT4Rec	0.080	0.042	0.026	2.32	117M	
2Mamba4Rec	0.110	0.061	0.047	0.127	1,765,344	78,624
LlamaRec	**0.150**	**0.098**	**0.064**	-	7B	
Hydra4Rec	0.112	0.059	0.044	**0.0005**	751238	

Author Contributions

If you'd like to, you may include a section for author contributions as is done in many journals. This is optional and at the discretion of the authors.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

References

Tri Dao Albert Gu. Mamba: Linear-time sequence modeling with selective state spaces. *arxiv*, 2024.

Jianghao Lin Chengkai Liu, Hanzhou Liu Jianling Wang, and James Caverlee. Mamba4rec: Towards efficient sequential recommendation with selective state space models. *arxiv*, 2024.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. *arxiv*, 2024.

Tian Wang Guanglei Xiong Alan Lu Gerard Medioni Jinming Li, Wentao Zhang. Gpt4rec: A generative framework for personalized recommendation and user interests interpretation. *arxiv*, 2023.

Julian McAuley Wang-Cheng Kang. Self-attentive sequential recommendation. *arxiv*, 2018.

Gabriel de Souza Pereira Moreira Dong Wang Even Oldridge Zhenrui Yue, Sara Rabhi. Llamarec: Two-stage recommendation using large language models for ranking. *arxiv*, 2023.

Table 3: MovieLens-1M

Model	HT@10	NDCG@10	MRR@10	Latency	# Parameters	# Parameters w/o Embeddings
SASRec	0.224	0.117	0.084	0.128	321,984	100,096
Mamba4Rec	0.306	0.178	0.138	0.016	297,312	78,624
MamRec	0.201	0.072	0.064	2.51	130M	
GPT4Rec	0.212	0.074	0.060	3.23	117M	
2Mamba4Rec	**0.340**	**0.193**	**0.148**	0.021	297,312	78,624
Hydra4Rec	0.308	0.179	0.140	**0.0005**	286854	
LlamaRec	0.148	0.067	-	-	7B	

Table 4: Datasets

Datasets	Users	Items	Reviews
Beauty and P.C.	750,835	211,452	6,860,059
MovieLens-1M	6,041	3,417	999,611
Video Games	98,907	26,355	857,505

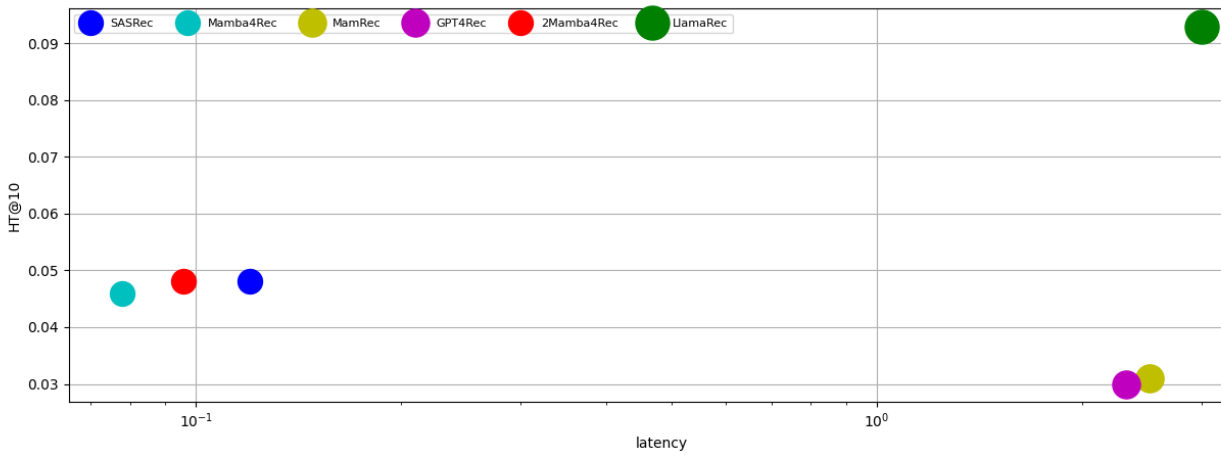

Figure 1: Beauty