SR-PredictAO: Session-based Recommendation with High-Capability Predictor Add-On

Ruida WANG, Raymond Chi-Wing WONG*, Weile TAN

The Hong Kong University of Science and Technology Kowloon

Hong Kong SAR

Code: https://github.com/RickySkywalker/SR-PredictAO-official.git

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

AGENDA

Background

1. Next-item recommender systems:
   - Predict the user’s **subsequent behavior**
   - Valued in both industry and academic for its commercial value
2. Session-based recommendation (SR):
   - Hot topic of the next-item recommender
   - Make recommendations based on the previous item clicks **within a single session**

1 Plot thanks to Pan et al. [2]

Motivation

1. Recommendation system:
   - Direct commercial value
   - Provide valuable information about human behavior
2. Session-based recommendation:
   - Inter-session dependency is strong [6]
   - Additional user information may not be available (privacy protection)
   - Friendly to cold-start scenario

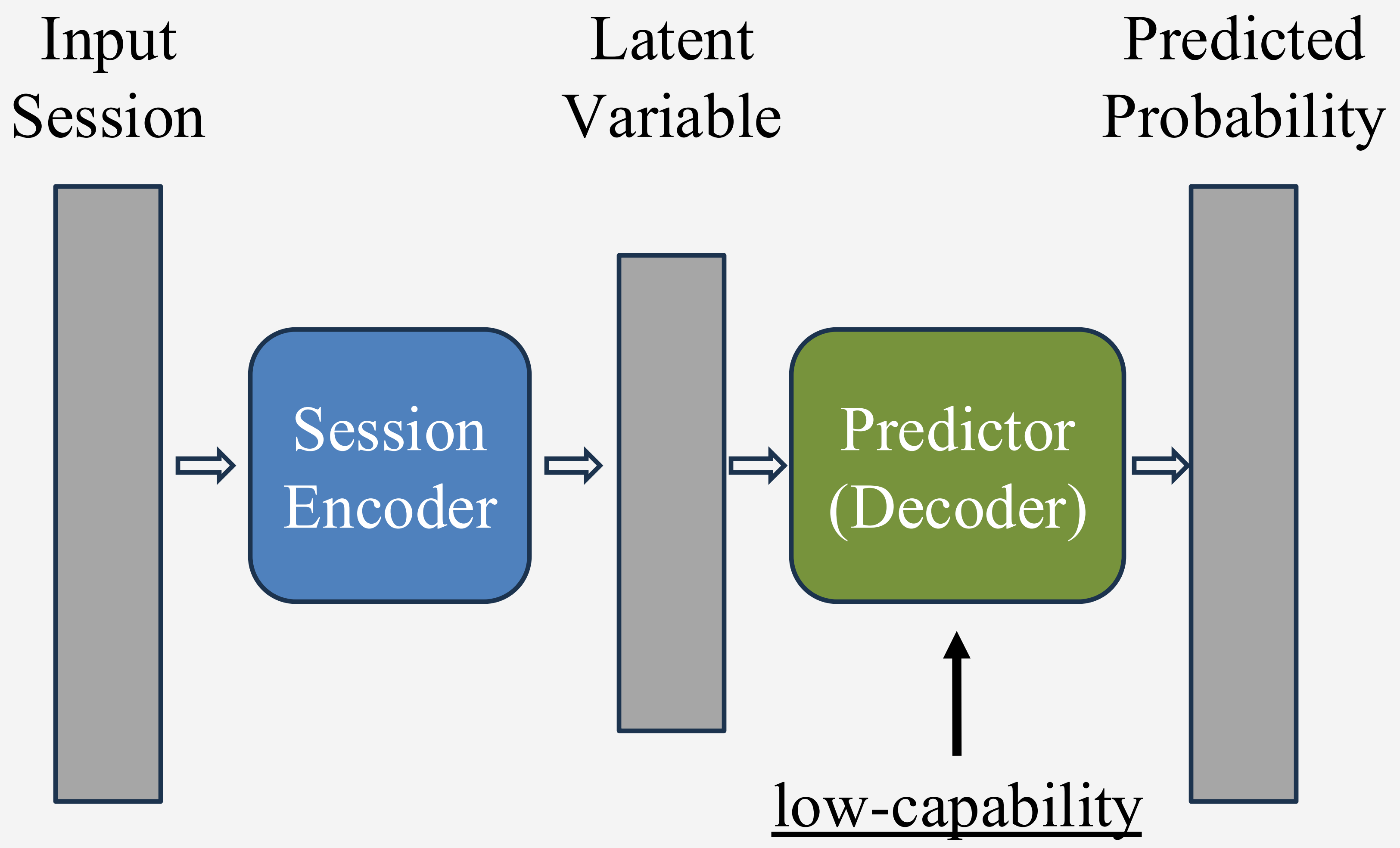
1

1 Plot thanks to Pan et al. [2]

Outstanding Issues

Issue 1: **Low-capability predictor** module

- Most SR models follows **encoder-predictor design** past research focus on different designs of the **encoder** modules.
- The **predictor** modules of most (if not all) existing SR models are **linear models**, which are **low-capability**.
- The **predictor** module should simulate the **complicated decision process** of a user.



Outstanding Issues

Issue 2: ***Overfit problem*** of a high-capability model

- Extremely high-capability model suffers from the serious overfit problem
- The predictor module needs to be an appropriate high-capability model

Issue 3: **Random user's behavior** in the input session

- A session can include any random behavior of the user
- e.g., Multi-intention problems: the user can be distracted from her/his original intention
- We model the general randomness as noise in the input.

Our solutions

香港科技大學 THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Issue 1:
low-capability predictor module

Issue 2:
overfitting problem

Issue 3:
random user's behavior

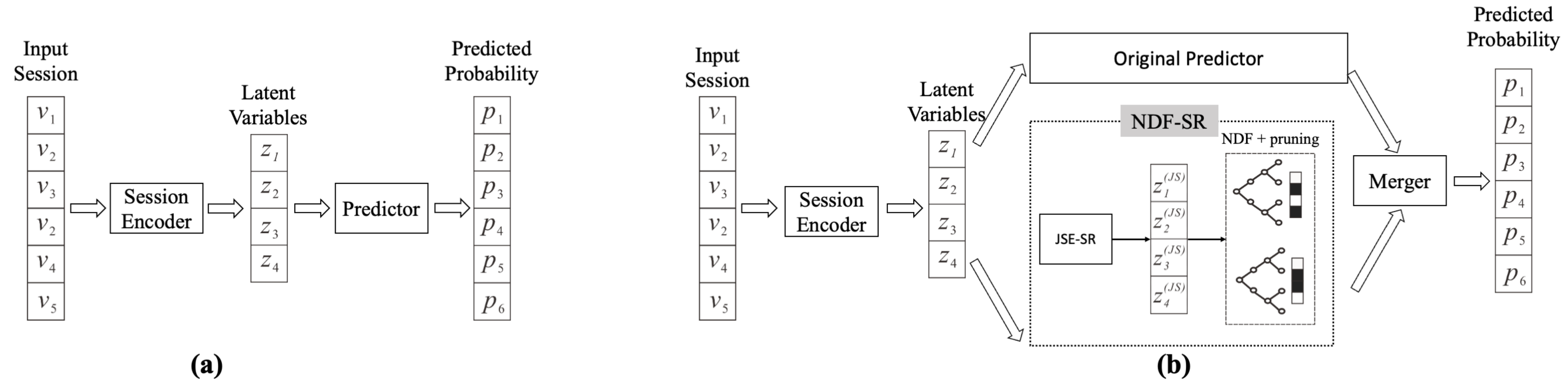
SR-PredictAO
(*Session-based Recommendation with Predictor Add-On*)

Neural Decision Forest (NDF) model

Pruning methods in the NDF model

Random user's Behavior Alleviator

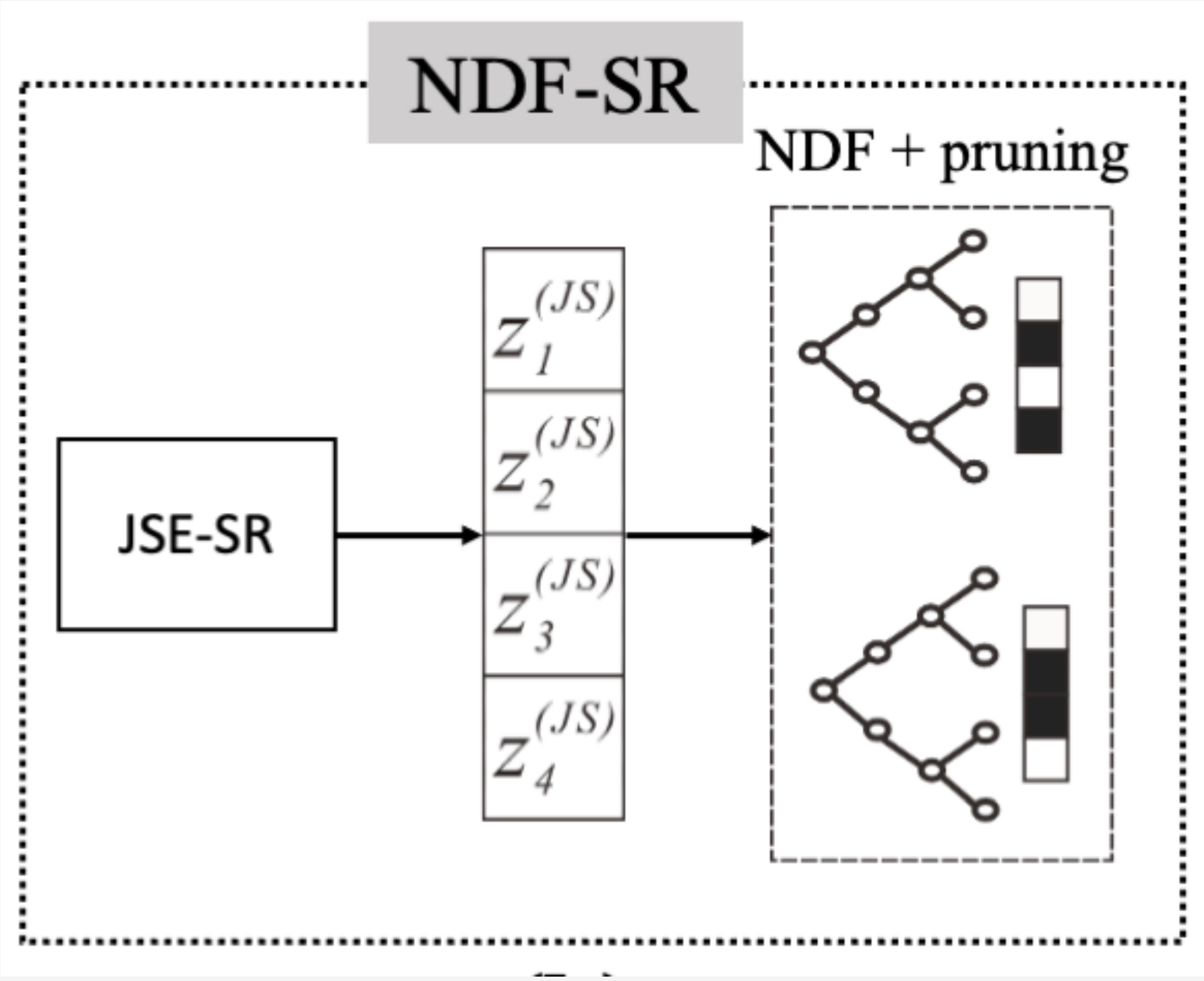
Methodology



NDF-SR

Neural Decision Forest for Session-based Recommendation (NDF-SR) involves:

1. Random User's Behavior Alleviator based on James-Stein Estimator (JSE-SR)
2. Neural Decision Forest (NDF) model
3. Pruning method for NDF



Random User's Behavior Alleviator

1. As mentioned, due to limited background information within the session, there are many random user behaviors that hard to remove, which are typically represented as "noise" in the data. (normally, $z \sim \mathcal{N}(\mu, \sigma^2)$)
2. We identified that widely used Most Likelihood Estimators (MLEs) has a high irreducible noise
3. To reduce such noise, we propose using a more noise-resistant James-Stein Estimator (JSE) in the SR problem.
4. We have JSE-SR, to alleviate general random user behavior in the session. (Detailed proof in paper)

Data we observed

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}_1^T \\ \boldsymbol{z}_2^T \\ \vdots \\ \boldsymbol{z}_m^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{\xi}_1 \\ \boldsymbol{\xi}_2 \\ \vdots \\ \boldsymbol{\xi}_{n'} \end{bmatrix}^T = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1n'} \\ z_{21} & z_{22} & \cdots & z_{2n'} \\ \vdots & \vdots & \ddots & \vdots \\ z_{m1} & z_{m2} & \cdots & z_{mn'} \end{bmatrix}$$

Underlying true value

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1n'} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2n'} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{m1} & \mu_{m2} & \cdots & \mu_{mn'} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_1^T \\ \boldsymbol{\mu}_2^T \\ \vdots \\ \boldsymbol{\mu}_m^T \end{bmatrix}$$

JS-Estimator

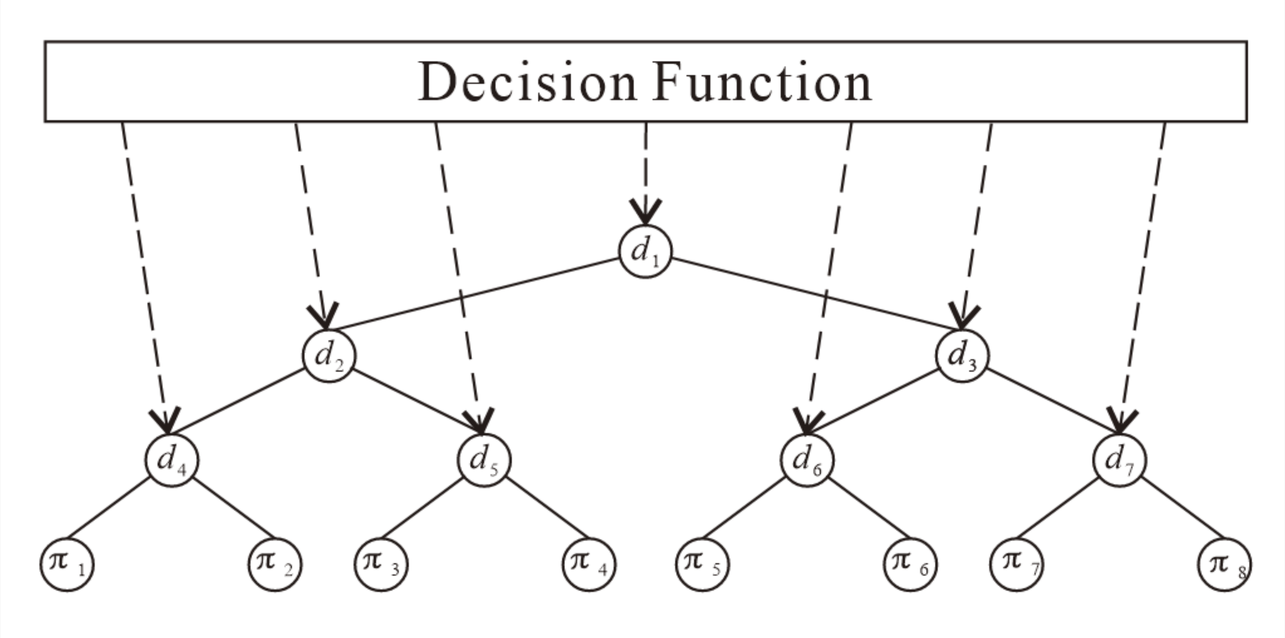
$$\hat{\mu}_{ij}^{(JS)} = \left(1 - \frac{m-2}{\|\boldsymbol{\xi}_j\|^2}\right) z_{ij}$$

JSE-SR lower noise

$$\mathbb{E}\Big[\sum_{i=1}^{m}(\mu_{ij} - \hat{\mu}_{ij}^{(JS)})^2\Big] \leqslant \mathbb{E}\Big[\sum_{i=1}^{m}(\mu_{ij} - \hat{\mu}_{ij}^{(MLE)})^2\Big]$$

Neural Decision Forest (NDF) Model

1. The motivation to use a tree-based model is because of its famous high-capability (or high Degrees-of-Freedom).
2. However, the traditional tree-based model has no gradient and cannot be optimized with an encoder.
3. Thus, inspired by [7], we propose an NDF-enhanced predictor for the session-based recommendation system.
4. We control the overfitting problem by developing a pruning method that randomly drops some leaves.



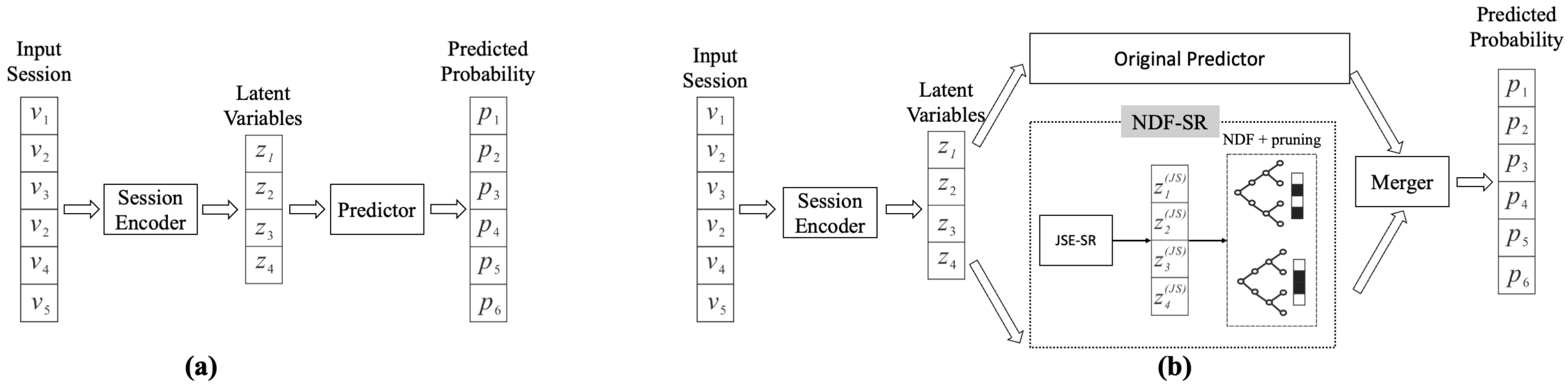
Theoretical foundation

1. Generally, a tree-based function can be understood as a simple function
2. By *Approximation of Measurable Functions by Simple Functions Theorem:* Consider the measurable function $f: X \rightarrow \mathbb{R}$ on a measure space $(X, \mathcal{M}, \mu)$. Then $\forall\, \epsilon > 0, \exists$ simple function ϕ s.t.:

$$\mu(\{x \in X: |f(x) - \phi(x)| > \epsilon\}) < \epsilon$$

   The convergence of such sequence can be uniform under $X - S_0$ under condition for $\mu(S_0) = 0$
3. However, the linear function with neural networks often struggles with highly discontinuous functions that have sharp changes

Methodology



Experiment

Base Models: LESSR [1], SGNN-HN [2], DIDN [3]

- We denote SR-PredictAO with base model *M* as "SR-PredAO(*M*)"

Datasets: Diginetica[1], Yoochoose[2]

- For the Yoochoose dataset, we use the most recent 1/64 for experiments, denoted as "Yoochoose 1/64"

Evaluation Metrics: commonly used HR@20 (Hit Rate) and MRR@20 (Mean Reciprocal Rank)

- HR@20 is denoted as Recall@20 or P@20 in some related works.

Baselines:

- Traditional methods: Item-KNN; GRU-based method: GRU4REC; Transformer-based method: STAMP, and SR-IEM; GNN-based method: SR-GNN, and base models

1 http://cikm2016.cs.iupui.edu/cikm-cup
2 http://2015.recsyschallenge.com/challenge.html

Dataset Statistics

Statistic	Yoochoose 1/64	Diginetica
# of Clicks	565,332	982,961
# of Training Sessions	375,625	647,523
# of Test Sessions	55,896	71,947
# of Items	17,792	43,097
Average length	6.14	5.12

TABLE I: Statistics of datasets

Experiment

Method	Diginetica		Yoochoose 1/64	
	HR@20	**MRR@20**	**HR@20**	**MRR@20**
Item-KNN	35.75	11.57	51.60	21.81
GRU4REC	29.45	8.33	60.64	22.89
STAMP	45.64	14.32	68.74	29.67
SR-IEM	52.35	17.64	71.15	31.71
SR-GNN	50.73	17.59	70.57	30.94
LESSR	51.71	18.15	70.94	31.16
SR-PredAO(LESSR)	**53.10**	**18.38**	**71.73**	**31.70**
Improvement (%)	**2.7**	**1.3**	**1.1**	**1.7**
p-value	$< 10^{-5}$	-	1.8×10^{-3}	-
SGNN-HN	55.67	**19.12**	72.06	**32.61**
SR-PredAO(SGNN-HN)	**55.91**	19.06	**72.62**	32.47
Improvement (%)	**0.4**	-0.3	**0.8**	-0.4
p-value	0.179	-	1.8×10^{-2}	-
DIDN	56.22	20.03	68.95	31.27
SR-PredAO(DIDN)	**57.86**	**20.49**	**69.50**	**31.44**
Improvement (%)	**2.9**	**2.3**	**0.8**	**0.5**
p-value	$< 10^{-5}$	-	2.3×10^{-2}	-

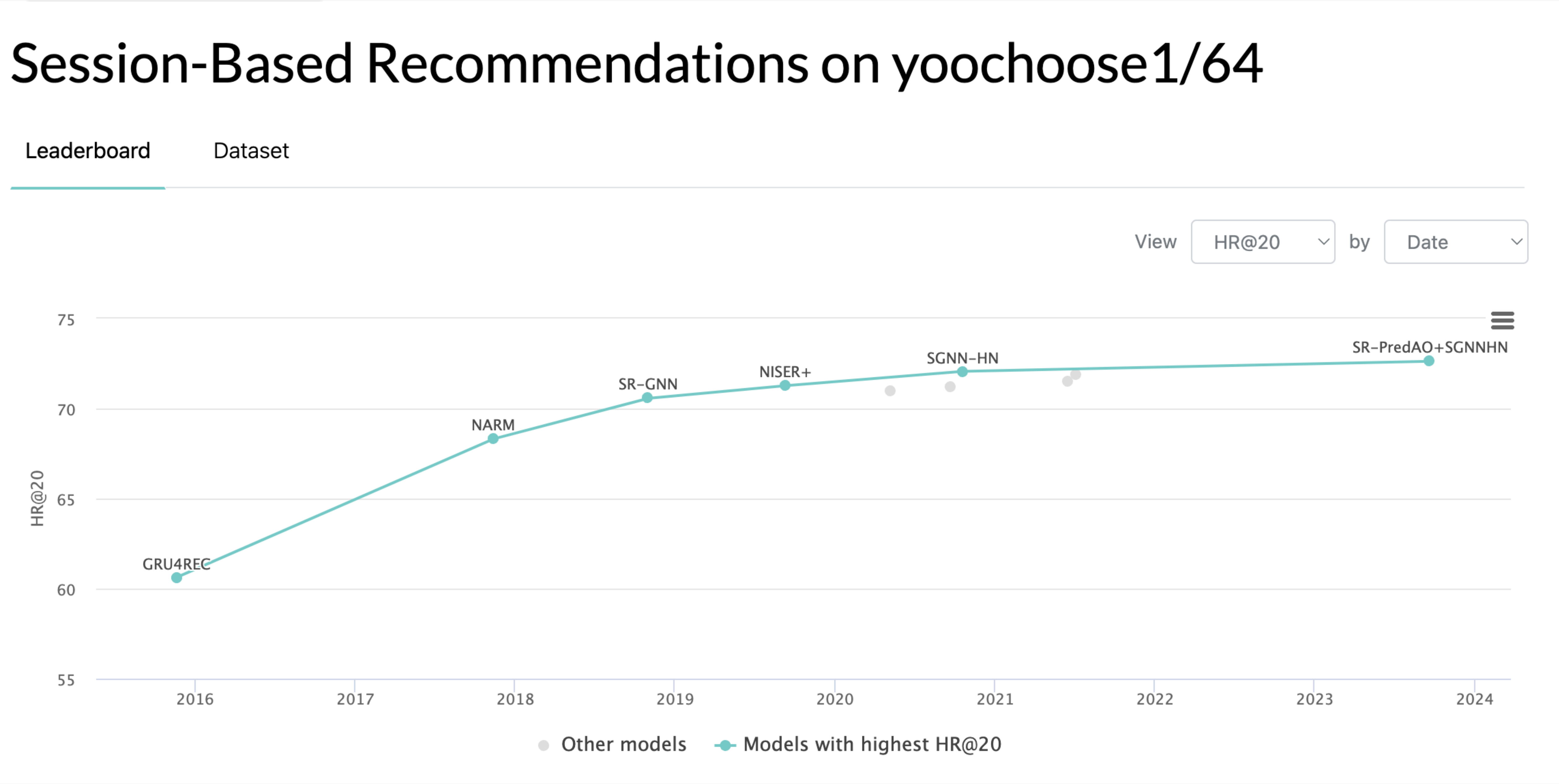
SOTA Performance

Session-Based Recommendations on Diginetica

Leaderboard   Dataset

Data source: https://paperswithcode.com/sota/session-based-recommendations-on-diginetica

SOTA Performance



Data source: https://paperswithcode.com/sota/session-based-recommendations-on-yoochoose1-1

Ablation Study

SR-PredictAO with three different base models

Ablation Study:

- Full model
- Without Random User's Behavior Alleviator
- Without NDT-pruning

Dataset	type	LESSR	SGNNHN	DIDN
Diginetica	full model	**53.15**	**55.91**	**57.86**
	w/o Alleviator	52.99	55.79	57.33
	w/o Pruning	53.06	55.78	57.26
YC 1/64	full model	**71.73**	**72.62**	**69.50**
	w/o Alleviator	71.67	72.58	69.26
	w/o Pruning	71.66	72.58	69.20

Quantify model capability

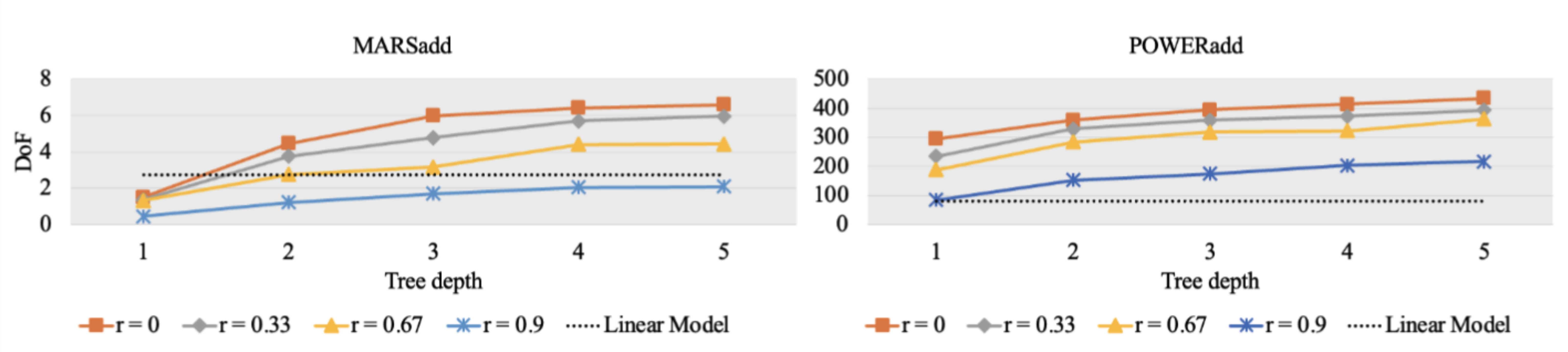
- We use **Degrees-of-Freedom (DoF)** to define the **capability** of a model formally.
- Given a dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{M}$, and $y_i = f(\boldsymbol{x}_i) + \varepsilon_i$.

  The model we fit to estimate f is denoted as $\hat{f}$, and the prediction is $\hat{y}_i = \hat{f}(\boldsymbol{x}_i)$.
- The Degrees-of-Freedom (DoF) is defined as $DoF(\hat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{N} \mathrm{Cov}(\hat{y}_i, y_i)$
- We perform DoF analysis of the **NDF-SR** module on two **generated** datasets with the following underlying functions:

MARSadd: $$y_i = 0.1e^{4x_{i1}} + \frac{4}{e^{-20(x_{i2}-0.5)}} + 3x_{i3} + 2x_{i4} + x_{i5} + \varepsilon_i$$

POWERadd: $$y_i = \sum_{j=1}^{5} x_{ij} + \sum_{j=6}^{10} x_{ij} + \varepsilon_i$$

Quantify model capability



1. From the above plots, we can see that by controlling the hyperparameters (depth & pruning rate) of the model, we can control the **DoF** of the **NDF-SR** module in a wide bandwidth.
2. The results proves that our framework can provide the predictor module with **sufficient** and **appropriate** capability

Takeaways

- We are the first to discover the important **low-capability** issue in the **predictor** module of most (if not all) existing models, such an issue lowers their prediction accuracy.
- We propose a framework called ***SR-PredictAO*** which could be applied to any existing models following the common encoder-predictor paradigm.
- Extensive **experimental results** on two public benchmark datasets show that when framework SR-PredictAO is applied to three existing state-of-the-art models, their performances are **consistently improved** by up to 2.9% on HR@20 and up to 2.1% on MRR@20.

Future Work

- The cost of the SR-PredictAO framework is relatively high when the number of leaf nodes is large. Thus, how to develop an **efficient tree-based method** is a potential direction.
- Explore the application of the **tree-based enhancement** in **other fields** such as **language** and **vision** modeling
- The **theoretical foundation** for neural-based trees also needs to be built up by future studies

References

[1] T. Chen and R. C.-W. Wong, “Handling information loss of graph neural networks for session-based recommendation,” in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 1172–1180.

[2] Z. Pan, F. Cai, W. Chen, H. Chen, and M. de Rijke, “Star graph neural networks for session-based recommendation,” in Proceedings of the 29th ACM international conference on information & knowledge management, 2020, pp. 1195–1204.

[3]X. Zhang, H. Lin, B. Xu, C. Li, Y. Lin, H. Liu, and F. Ma, “Dynamic intent-aware iterative denoising network for session-based recommendation,” Information Processing & Management, vol. 59, no. 3, p. 102936, 2022.

[4] Zheng, Y., Liu, S., Li, Z., & Wu, S. (2020, November). Dgtn: Dual-channel graph transition network for session-based recommendation. In 2020 international conference on data mining workshops (ICDMW) (pp. 236-242). IEEE.

[5] Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., & Tan, T. (2019, July). Session-based recommendation with graph neural networks. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 346-353).

[6] Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., ... & Sampath, D. (2010, September). The YouTube video recommendation system. In Proceedings of the fourth ACM conference on Recommender systems (pp. 293-296).

[7] Kontschieder, P., Fiterau, M., Criminisi, A., & Bulo, S. R. (2015). Deep neural decision forests. In Proceedings of the IEEE international conference on computer vision (pp. 1467-1475).

Code link:

Q & A

Ruida WANG

Raymond Chi-Wing WONG*

Weile TAN