On the Evaluation of Unlearning in Session-Based Recommendation

Liu Yang[1], Zhaochun Ren[2], Ziqi Zhao[1], Pengjie Ren[1], Zhumin Chen[1], Jun Ma[1], Xin Xin[1]

[1]Shandong University, Qingdao, China

[2]Leiden University, Leiden, The Netherlands

Outline

- Introduction
- Methods
- Experiments
- Conclusion & Future Work

Our task: session-based recommendation unlearning

- Session-based recommendation

- Unlearning

Legitimacy & User privacy

- Session-based recommendation unlearning (item-level & session-level)

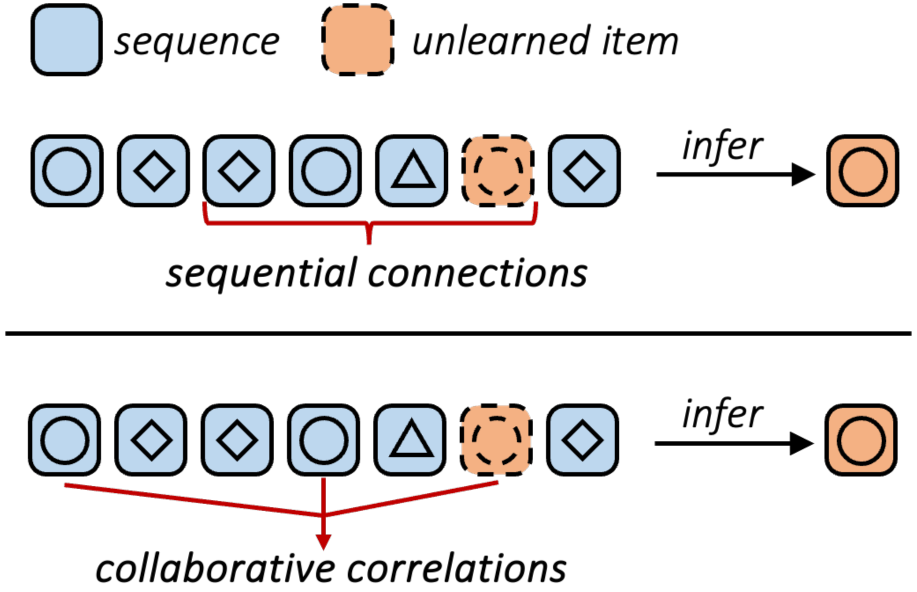
Challenges

- Exact unlearning is hard to achieve.

- Existing recommendation unlearning methods do not evaluate the unlearning effectiveness.

Our contributions

- Exact unlearning is hard to achieve.

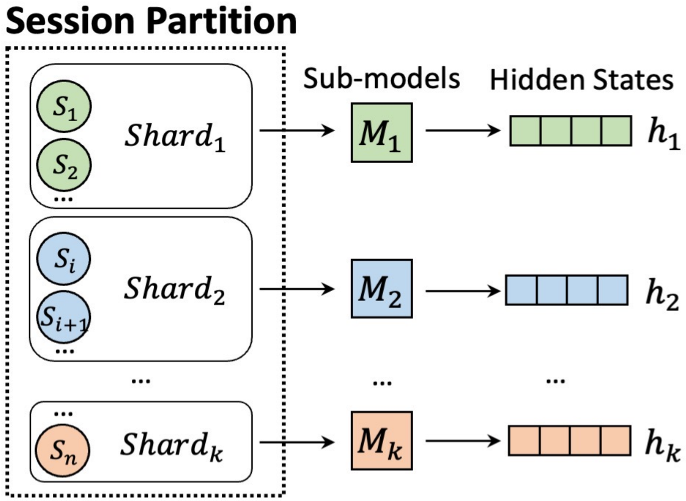


We propose an unlearning framework SRU and three extra deletion strategies.

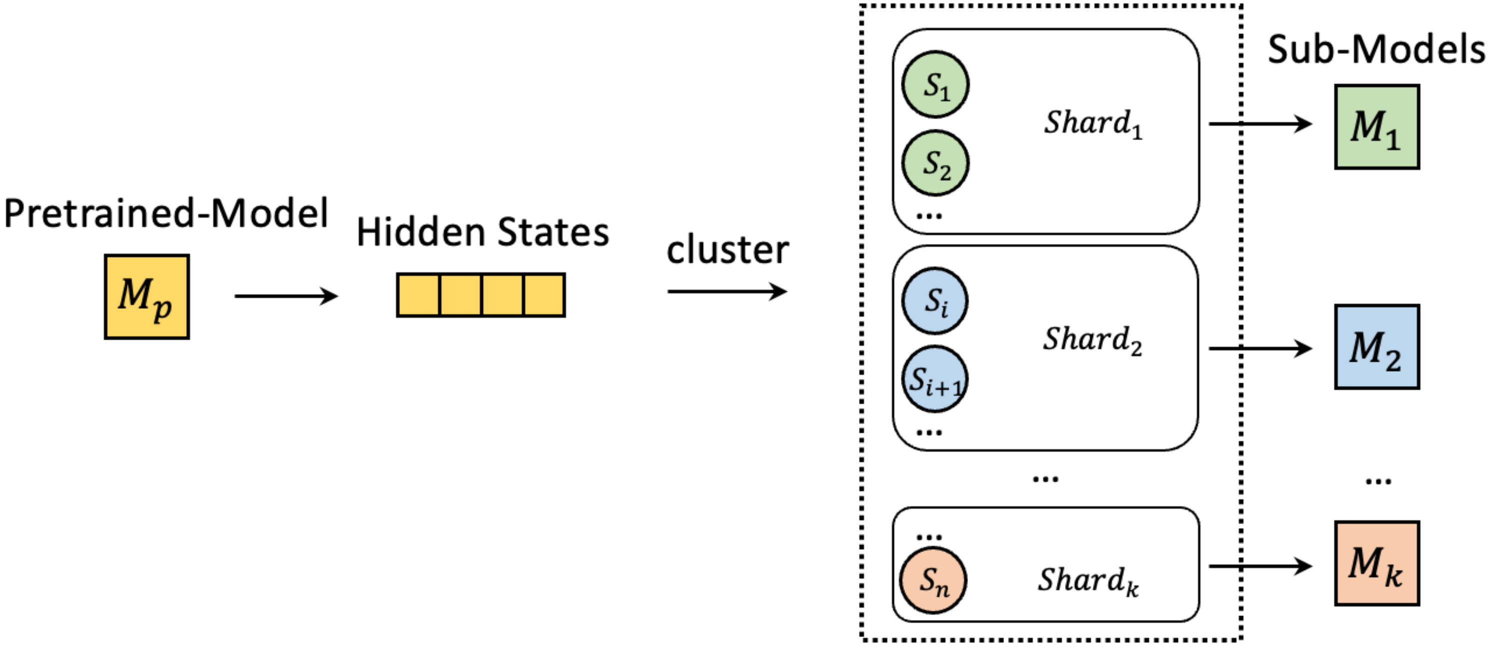
- Existing recommendation unlearning methods do not evaluate unlearning effectiveness.

We propose an evaluation metric.

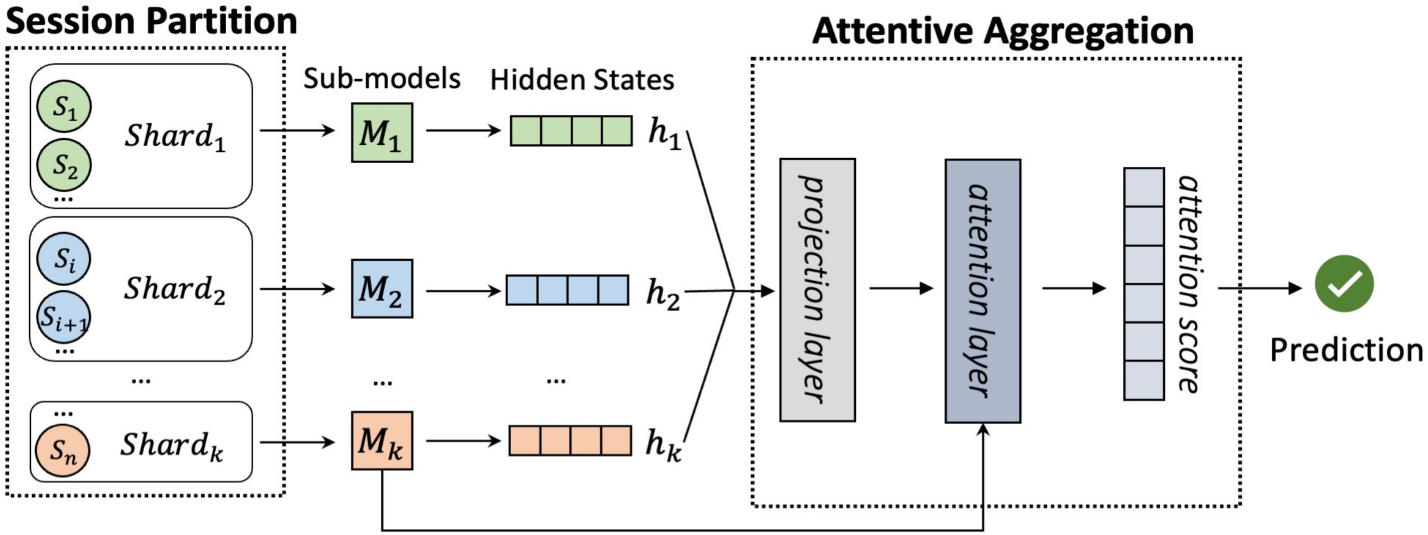
Our method: SRU——Training



Task: divide the training sessions into disjoint data shards and then sub-models are trained on each shard.



Our method: SRU ——Training



Projection layer

$$\mathbf{h}'_k = \mathbf{W}_k \mathbf{h}_k + \mathbf{b}_k$$

Attention layer

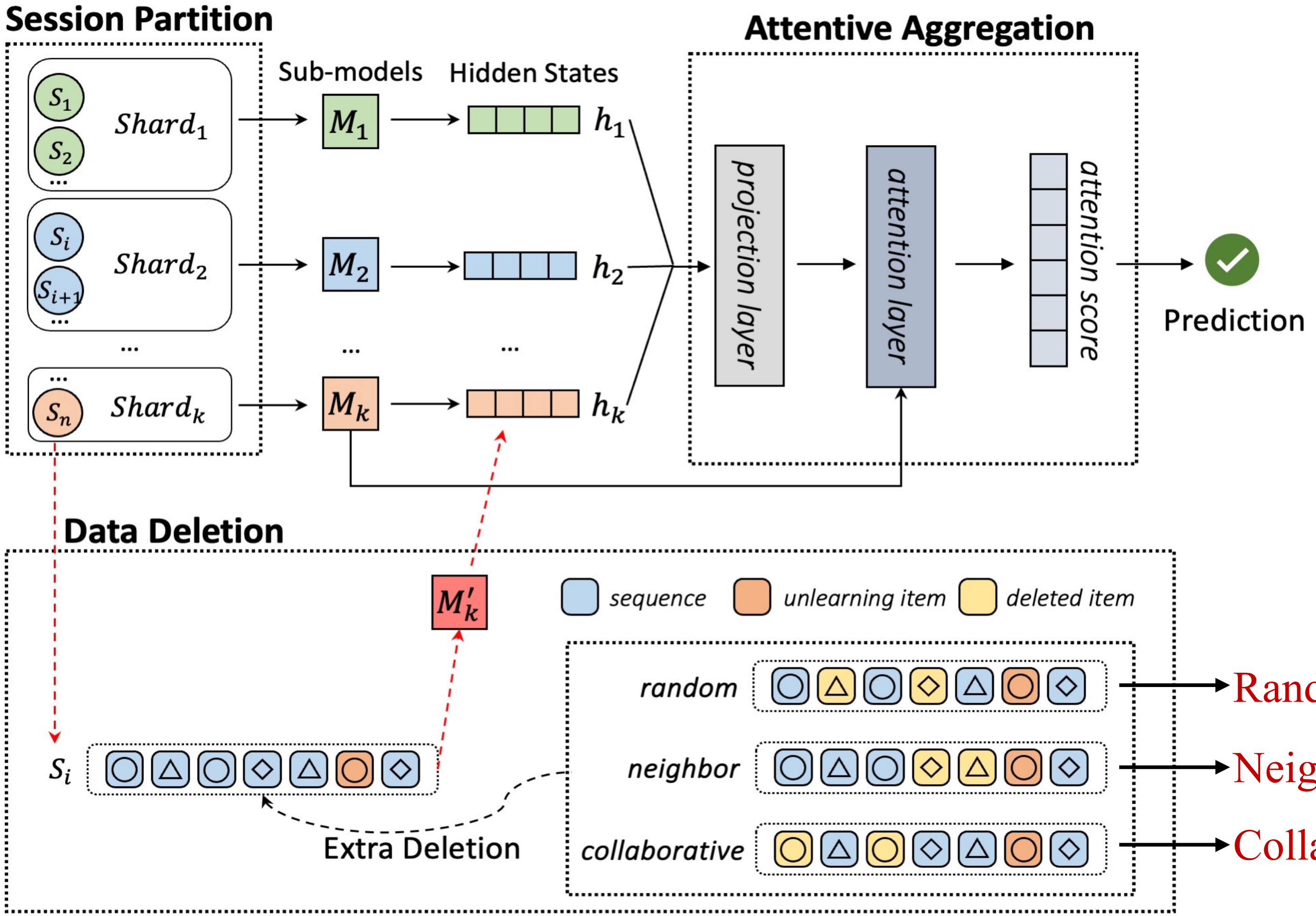
$$a_k = softmax(\mathbf{g} \cdot ReLU(\mathbf{W}' \odot (\mathbf{h}'_k \odot \mathbf{c}'_k) + \mathbf{b}'))$$

$$\mathbf{h}^f = \sum_{k=1}^{\mathcal{K}} a_k \mathbf{h}'_k$$

Output layer

Task: fuses the hidden states coming from different sub-models for the final prediction.

Our method: SRU ——Unlearning



Task: apply extra data deletion strategies to the corresponding session.

Random Extra Deletion(RED)

Neighbor Extra Deletion(NED)

Collaborative Extra Deletion(CED)

Evaluation

- The unlearned item should not be recommended to the user again in the near future.
- For item-level unlearning: We define one unlearning effectiveness evaluation metric as the hit ratio (i.e., HIT@K) which measures whether the unlearned item would occur in the top-K recommendation list.
- Lower scores denote better results.
- For session-level unlearning: We use membership inference attacks.

Experimental setups

Three real-world datasets:

- Amazon Beauty, Games, and Steam.
- 80% for training, 10% for validation, 10% for testing.
- Metrics

  For recommendation performance: Recall@k and NDCG@k, k=10, 20.

  For unlearning effectiveness: HIT@k, k=1, 5, 10, 20

Recommendation models:

- GRU4Rec, SASRec, and BERT4Rec.

Experimental questions

- **RQ1:** How is the recommendation performance of SRU when instantiated with different session-based recommendation models?
- **RQ2:** How is the unlearning effectiveness of SRU?
- **RQ3:** How is the unlearning efficiency of SRU?

Experimental results: overall recommendation performance(RQ1)

Beauty	GRU4Rec				SASRec				BERT4Rec			
	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20
Retrain	0.0327	0.0382	0.0550	0.0768	0.0399	0.0450	0.0632	0.0835	0.0314	0.0380	0.0558	0.0816
SISA	0.0289	0.0328	0.0460	0.0615	0.0271	0.0307	0.0428	0.0571	0.0259	0.0310	0.0464	0.0666
SRU-R	0.0304	**0.0347**	0.0489	0.0662	0.0280	**0.0323**	0.0448	**0.0617**	0.0292	0.0341	0.0509	0.0704
SRU-C	0.0286	0.0330	0.0468	0.0643	**0.0280**	0.0320	**0.0456**	0.0616	**0.0293**	**0.0348**	**0.0525**	**0.0743**
SRU-N	**0.0306**	0.0346	**0.0506**	**0.0668**	0.0274	0.0312	0.0440	0.0591	0.0291	0.0346	0.0507	0.0726

Steam	GRU4Rec				SASRec				BERT4Rec			
	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20
Retrain	0.0495	0.0631	0.0947	0.1489	0.0539	0.0679	0.1016	0.1574	0.0593	0.0742	0.1116	0.1711
SISA	0.0471	0.0601	0.0898	0.1412	0.0457	0.0581	0.0863	0.1357	0.0482	0.0615	0.0932	0.1460
SRU-R	**0.0490**	**0.0621**	**0.0924**	0.1444	**0.0485**	**0.0614**	**0.0914**	**0.1431**	**0.0577**	**0.0722**	**0.1077**	**0.1652**
SRU-C	0.0484	0.0616	0.0916	**0.1445**	0.0476	0.0604	0.0901	0.1411	0.0576	0.0720	0.1075	0.1648
SRU-N	0.0480	0.0612	0.0916	0.1442	0.0480	0.0608	0.0906	0.1414	0.0567	0.0710	0.1067	0.1636

Games	GRU4Rec				SASRec				BERT4Rec			
	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20
Retrain	0.0401	0.0495	0.0747	0.1122	0.0479	0.0580	0.0864	0.1268	0.0474	0.0596	0.0921	0.1406
SISA	0.0324	0.0377	0.0564	0.0776	0.0267	0.0318	0.0459	0.0661	0.0322	0.0402	0.0629	0.0948
SRU-R	**0.0357**	0.0424	**0.0621**	0.0887	**0.0333**	**0.0405**	**0.0596**	**0.0883**	**0.0395**	**0.0497**	**0.0752**	**0.1159**
SRU-C	0.0342	0.0410	0.0614	0.0887	0.0314	0.0378	0.0570	0.0824	0.0363	0.0462	0.0690	0.1084
SRU-N	0.0352	**0.0424**	0.0620	**0.0909**	0.0321	0.0393	0.0566	0.0851	0.0384	0.0488	0.0730	0.1146

SRU always performs better than SISA even though SRU has removed more training data.

Experimental results: unlearning effectiveness(RQ2)

Beauty	GRU4Rec				SASRec				BERT4Rec			
	HIT@1	HIT@5	HIT@10	HIT@20	HIT@1	HIT@5	HIT@10	HIT@20	HIT@1	HIT@5	HIT@10	HIT@20
Retrain	0.0764	0.1715	0.2294	0.3052	0.0619	0.1566	0.2123	0.2807	0.0700	0.1588	0.2080	0.2739
SISA	0.0685	0.1654	0.2244	0.3074	0.0681	0.1605	0.2222	0.3091	0.0763	0.1730	0.2321	0.3119
SRU-R	0.0675	0.1561	0.2122	0.2809	0.0625	0.1468	0.2042	0.2697	0.0720	0.1573	0.2131	0.2798
SRU-C	**0.0577**	**0.1335**	**0.1824**	**0.2510**	**0.0593**	**0.1429**	**0.1970**	**0.2666**	0.0661	**0.1516**	0.2058	**0.2689**
SRU-N	0.0643	0.1533	0.2028	0.2731	0.0605	0.1482	0.2039	0.2736	**0.0638**	0.1527	**0.2054**	0.2759
Steam	**GRU4Rec**				**SASRec**				**BERT4Rec**			
	HIT@1	HIT@5	HIT@10	HIT@20	HIT@1	HIT@5	HIT@10	HIT@20	HIT@1	HIT@5	HIT@10	HIT@20
Retrain	0.1581	0.3992	0.5372	0.6805	0.1411	0.3636	0.4975	0.6483	0.1159	0.3292	0.4701	0.6309
SISA	0.1582	0.3979	0.5349	0.6775	0.1410	0.3646	0.4959	0.6365	0.1166	0.3282	0.4668	0.6184
SRU-R	0.1545	0.3954	0.5319	0.6739	0.1412	0.3687	0.5020	0.6417	0.0992	0.2979	0.4282	0.5749
SRU-C	0.1499	0.3882	0.5241	0.6702	0.1389	0.3686	0.5041	0.6475	0.1036	0.3088	0.4407	0.5901
SRU-N	**0.1461**	**0.3799**	**0.5136**	**0.6568**	**0.1138**	**0.3186**	**0.4422**	**0.5812**	**0.0957**	**0.2897**	**0.4205**	**0.5713**

- The unlearned item still has a high probability of being inferred again from the remaining interactions in the session.
- SRU-R, SRU-C and SRU-N achieve better unlearning effectiveness.

Experimental results: unlearning effectiveness(RQ2)

Beauty	GRU4Rec		SASRec		BERT4Rec	
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
Retrain	0.7487	0.6048	0.7661	0.7357	0.9054	0.6977
SISA	0.7412	0.5821	0.7487	0.5421	0.8743	0.5419
SRU	0.7688	0.6959	0.794	**0.7552**	0.9196	0.7049
SRU-C	**0.8040**	**0.7181**	**0.8091**	0.7275	**0.9347**	**0.7689**

Steam	GRU4Rec		SASRec		BERT4Rec	
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
Retrain	0.4081	0.5757	0.5077	0.5535	0.4472	0.5447
SISA	0.4050	0.5757	0.4992	0.5348	0.4102	0.5662
SRU	0.5085	0.5751	0.5242	0.5771	0.5507	0.5038
SRU-C	**0.5314**	**0.5999**	**0.5371**	**0.5986**	**0.5662**	**0.5766**

Games	GRU4Rec		SASRec		BERT4Rec	
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
Retrain	0.6438	**0.6797**	0.7397	0.6476	0.7808	0.6123
SISA	0.5734	0.5718	0.6783	0.5633	0.7762	0.5536
SRU	0.6433	0.6091	0.7482	0.7026	0.8182	**0.6344**
SRU-C	**0.6853**	0.6526	**0.7692**	**0.7137**	**0.8741**	0.5798

- SRU-C has the highest Accuracy scores with a reasonable AUC score in all datasets and models which means that it has better unlearning effectiveness.

Experimental results: unlearning efficiency(RQ3)

Dataset		Beauty			Games			Steam		
Method		GRU4Rec	SASRec	BERT4Rec	GRU4Rec	SASRec	BERT4Rec	GRU4Rec	SASRec	BERT4Rec
Retrain		46.80m	55.60m	55.76m	31.22m	29.91m	31.14m	274.67m	368.99m	296.89m
SRU	Sub-model	5.80m	5.07m	7.44m	3.76m	4.75m	4.80m	33.67m	36.78m	34.07m
	Aggregation	0.72m	6.05m	5.53m	1.78m	4.40m	3.87m	25.30m	62.53m	64.30m
	Total	**6.52m**	**11.12m**	**12.97m**	**5.54m**	**9.15m**	**8.67m**	**58.97m**	**99.31m**	**98.37m**

SRU performs much more efficiently than Retrain.

Conclusions

- Due to plenty of collaborative correlations and sequential connections, simply removing the unlearning samples cannot achieve the exact unlearning effect.
- Unlearning effectiveness is also an important metric of session-based recommendation unlearning.
- We proposed SRU framework and three extra deletion strategies to tackle the above challenges.

Future Work

- Session-level unlearning.
- The trade-off between unlearning effectiveness, recommendation performance, and unlearning efficiency.

Liu Yang
Shandong University
Qingdao, China
E-mail: yangliushirry@gmail.com

Thanks for your attention!

Code: https://github.com/shirryliu/SRU-code