Self-Guided Learning to Denoise for Robust Recommendation

Yunjun Gao[1], **Yuntao Du**[1], Yujia Hu[1], Lu Chen[1],
Xinjun Zhu[1], Ziquan Fang[1], Baihua Zheng[2]

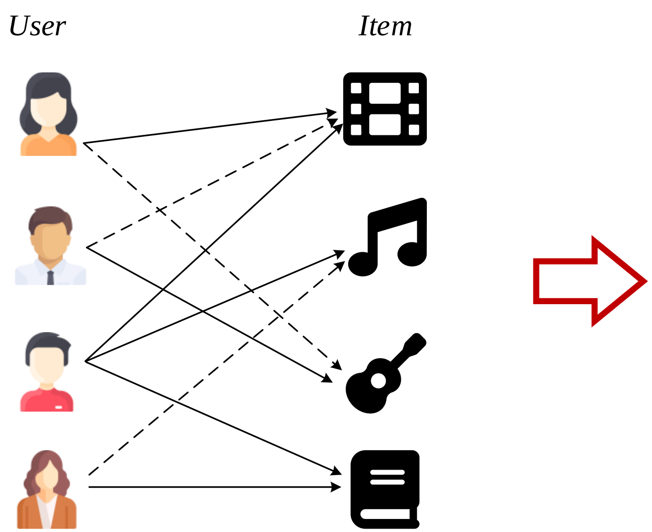
[1]*College of Computer Science, Zhejiang University, China*
[2]*School of Computing and Information Systems, Singapore Management University, Singapore*

Outline

- **Motivation**
- Related Work
- Preliminaries
- Methods
- Experimental Evaluation
- Conclusions

Motivation

- **Implicit feedback is the default choice of modern RS**
  - Large volume and high availability
- **Implicit feedback is inherently noisy**
  - Cannot directly indicate the users’ true preferences
  - Ubiquitous presence of noisy-positive and noisy-negative samples



1	0	1	0
1	0	1	0
1	1	0	1
0	1	0	1

Denoising matters!

Motivation (Cont.)

Solutions of existing methods

a) Design a score function to measure the “cleanness” of interactions (e.g., loss values)

b) Assign different weight to each interaction

c) Train the model with the re-weighting samples

Challenges

- Abandon of hard yet clean samples
- Lack of adaptivity and universality

SGDL: Self-Guided Learning to Denoise for Robust Recommendation

Outline

- Motivation
- **Related Work**
- Preliminaries
- Methods
- Experimental Evaluation
- Conclusions

Related Work

Sample Selection

- Select clean and informative samples through different sampling probabilities
- Drawbacks: high variance of denoising performance.

Sample Re-weighting

- Focus on the learning process of models (e.g., loss values) to assign different weights to clean and noisy samples
- Drawbacks: handcraft functions and poor generalization.

Other Directions

- Use auxiliary information or design model-specific structures
- Drawbacks: lack of adaptivity and universality.

Outline

- Motivation
- Related Work
- **Preliminaries**
- Methods
- Experimental Evaluation
- Conclusions

Preliminaries

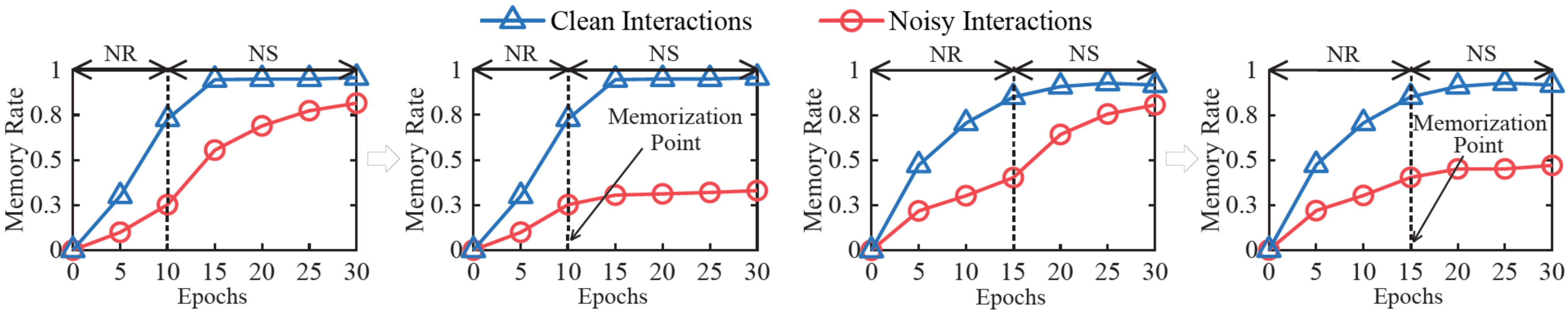
Problem Statement

- ***Implicit Feedback*** $\mathcal{D} = \{u, i, y_{ui} | u \in \mathcal{U}, i \in \mathcal{I}\}$
  - $\mathcal{U}$ is the set of users, $\mathcal{I}$ is the set of items
  - $y_{ui} \in \{0,1\}$ is the interaction that indicates whether user u has interacted with item i
- ***Denoising Learning Task***
  - Given the noisy implicit feedback $\mathcal{D}$, infer users' true preference with optimal model parameter θ^*

Outline

- Motivation
- Related Work
- Preliminaries
- **Methods**
- Experimental Evaluation
- Conclusions

Memorization Effect



(a) NeuMF (b) NeuMF with SGDL (c) LightGCN (d) LightGCN with SGDL

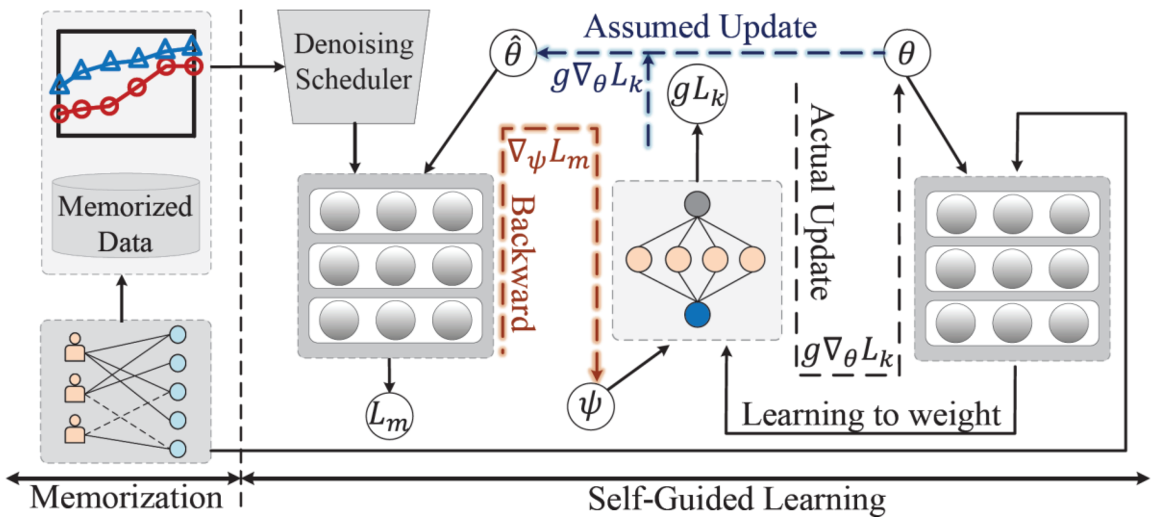
Memorization effect of recommendation models

- Models focus on learning easy and clean patterns at their early stage of training;
- And eventually memorize all the implicit feedback at the later stage.

Overview

SGDL Framework

- **Phase I:** Memorization
  - Most memorized interactions are clean until memorization point
  - Collect memorized data as denoising signals for training in Phase II
- **Phase II:** Self-Guided Learning
  - Leverage memorized data as clean signals to guide the training process
  - Use a novel adaptive denoising scheduler to improve robustness



Phase I: Memorization

Memorized Interactions

- An interaction (u, i) is memorized if item i is in the top-N ranking list of user u, where N is the length of u's all observed interactions
- Consider results of recent h epochs to improve stableness

$$m_t^h(u,i) = \frac{1}{\left|\mathcal{P}_t^h(u,i)\right|} \sum_{m_j(u,i)\in\mathcal{P}_t^h(u,i)} m_i(u,i)$$

Phase I: Memorization(Cont.)

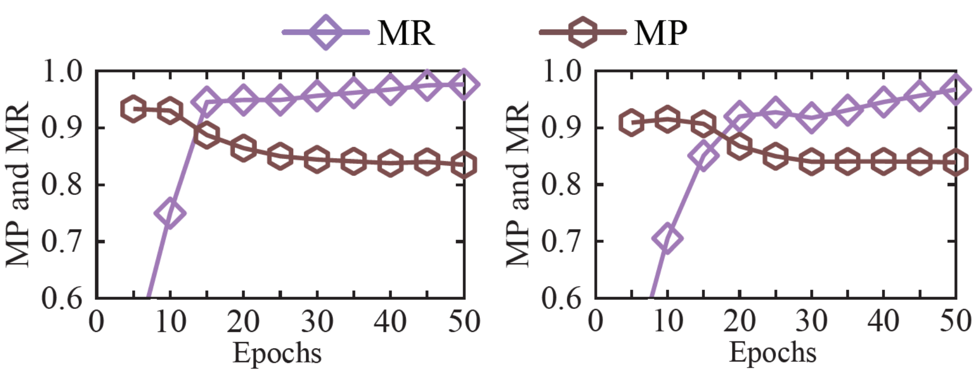
Memorization Point Estimation

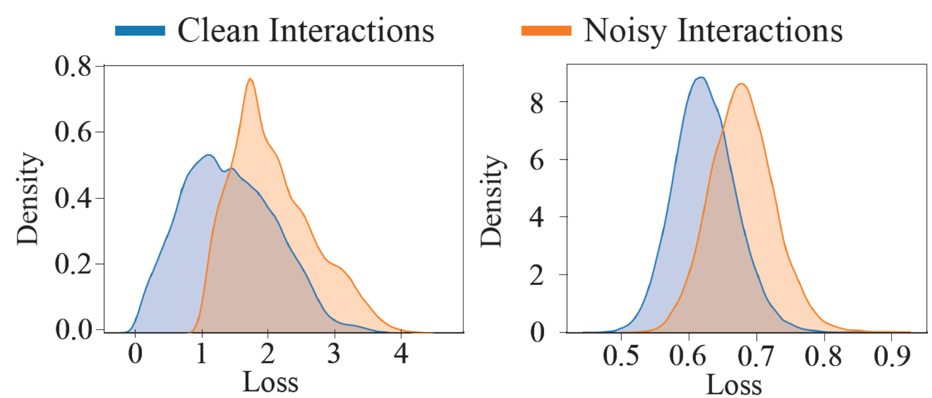
- The best memorization point should be the best trade-off point between memorization Precision and memorization Recall (i.e. MP=MR)

$$MP_t = \frac{|\mathcal{R}_t|}{|\mathcal{M}_t|} \qquad MR_t = \frac{|\mathcal{R}_t|}{|\mathcal{M}_t|} \qquad \mathcal{M}_{t_m} = |\{(u,i\} \in \mathcal{D}_t : y_{ui} = y^*_{ui})| = (1-\sigma)|\mathcal{D}|$$

Noise ratio

- Use GMM to estimate noise ratio





Phase II: Self-Guided Learning

Denoising Learning with Memorized Data

- Formulate weighting function as a simple MLP
- Solve the bi-level optimization problem in a meta-learning manner

$$\theta^*(\psi) = \mathrm{argmin}_\theta \frac{1}{|\mathcal{D}_T|} \sum_k^{|\mathcal{D}_T|} g(L_k(\theta); \psi) L_k(\theta) \qquad \psi^* = \mathrm{argmin}_w \frac{1}{|\mathcal{M}_{t_m}|} \sum_m^{|\mathcal{M}_{t_m}|} L_m(\theta^*(\psi))$$

Adaptive Denoising Scheduler

- Leverage intermediate outputs to quantify the contribution of each memorized data
- Assign different sampling probability according to their contributions

$$o_m = s\left(L_m(\theta), \cos\left(\nabla_{\hat{\theta}} L_m(\hat{\theta}), \nabla_\theta L_m(\theta)\right); \phi\right) \qquad \pi_m = \frac{\exp(o_m; \phi)}{\sum_{i \in \mathcal{M}_{t_m}} \exp(o_i; \phi)}$$

Outline

- Motivation
- Related Work
- Preliminaries
- Methods
- **Experimental Evaluation**
- Conclusions

Experimental Evaluation

Datasets

- Adressa, Yelp and MovieLens are three widely-used real world datasets for state-of-the-art recommender systems.

Dataset	#Users	#Items	#Interactions	Sparsity
Adressa	212,231	6,596	419,491	99.97%
MovieLens	943	1,683	100,000	93.70%
Yelp	45,548	57,396	1,672,520	99.94%

Evaluation

- Equipped with 4 representative base models and 2 different loss functions.
- Compared with 6 state-of-the-art denoising learning schemes, including 2 graph-specific robust learning methods.

Metrics

- Recall@N , NDCG@N (N=5, 20).

Experimental Evaluation (Cont.)

Overall results of SGDL

Database		Adressa				MovieLens				Yelp			
Base Model	**Method**	**R@5**	**R@20**	**N@5**	**N@20**	**R@5**	**R@20**	**N@5**	**N@20**	**R@5**	**R@20**	**N@5**	**N@20**
NeuMF	Normal	0.1533†	0.3208†	0.1224†	0.1808†	0.1023†	0.2687†	0.2890†	0.2765†	0.0129†	0.0393†	0.0129†	0.0215†
	WBPR	0.1538†	0.3207†	0.1225†	0.1809†	0.1025†	0.2689†	0.2891†	0.2769†	0.0128†	0.0392†	0.0127†	0.0214†
	IR	0.1541†	0.3212†	0.1229†	0.1830†	0.1054†	0.2704†	0.2928†	0.2758†	0.0132†	0.0407†	0.0131†	0.0229†
	T-CE	0.1537†	0.3220†	0.1267†	0.1839†	0.1025†	0.2821†	0.2923†	0.2845†	0.0119†	0.0396†	0.0119†	0.0211†
	DeCA	0.1597	0.3205†	0.1226†	0.1799†	0.1024†	0.2723†	0.2904†	0.2801†	0.0129†	0.0394†	0.0129†	0.0216†
	SGDL	**0.1598**	**0.3291**	**0.1272**	**0.1853**	**0.1135**	**0.2844**	**0.3279**	**0.3032**	**0.0155**	**0.0469**	**0.0158**	**0.0260**
CDAE	Normal	0.1445†	0.3159†	0.0987†	0.1886†	0.0904†	0.2185†	0.2617†	0.2356†	0.0145†	0.0436†	0.0149†	0.0277†
	WBPR	0.1443†	0.3158†	0.0987†	0.1890†	0.0908†	0.2184†	0.2619†	0.2346†	0.0148†	0.0437†	0.0151†	0.0278†
	IR	0.1444	0.3152†	0.0981†	0.1893†	0.0909†	0.2186†	0.2612†	0.2358†	0.0153†	0.0438	0.0152†	0.0278†
	T-CE	0.1415†	0.3106†	0.0991	0.1840†	0.0912†	0.2158†	0.2642	0.2386†	0.0147†	**0.0439**	0.0151†	0.0279†
	DeCA	0.1447†	0.3159†	0.0991	0.1888†	0.0917†	0.2189†	0.2641	0.2378†	0.0158†	0.0438	0.0154†	0.0292†
	SGDL	**0.1450**	**0.3181**	**0.0993**	**0.1956**	**0.0921**	**0.2220**	**0.2643**	**0.2404**	**0.0162**	**0.0439**	**0.0172**	**0.0296**
NGCF	Normal	0.0769†	0.1322†	0.0571†	0.0769†	0.1285†	0.3103†	0.3694†	0.3392†	0.0267†	0.0736†	0.0262†	0.0417†
	WBPR	0.0770†	0.1324†	0.0572†	0.0769†	0.1287†	0.3105†	0.3692†	0.3395†	0.0265†	0.0739†	0.0265†	0.0417†
	IR	0.0772†	0.1337†	0.0570†	0.0768†	0.1280†	0.3104†	0.3701†	0.3395†	0.0269†	0.0737†	0.0261†	0.0412†
	DeCA	0.0760†	0.1326†	0.0571†	0.0766†	0.1304†	0.3113†	0.3729†	0.3401	0.0277	0.0739†	0.0262	0.0418
	SGCN	0.0773†	0.1336†	0.0543†	0.0770	0.1288†	0.3112†	**0.3768**	0.3401	0.0267†	0.0734†	0.0265	**0.0443**
	SGL	0.0775†	0.1345	0.0576	0.0768†	0.1303†	0.3141†	0.3763†	0.3360†	**0.0279**	**0.0750**	0.0264†	0.0409†
	SGDL	**0.0788**	**0.1347**	**0.0579**	**0.0771**	**0.1309**	**0.3186**	0.3745	**0.3404**	0.0273	0.0746	**0.0267**	0.0420
LightGCN	Normal	0.0951†	0.1817†	0.0713†	0.0994†	0.1258†	0.3173†	0.3678†	0.3358†	0.0334†	0.0912†	0.0332†	0.0515†
	WBPR	0.0958†	0.1845†	0.0733†	0.1006†	0.1262†	0.3189†	0.3701†	0.3510	0.0333†	0.0911†	0.0331†	0.0512†
	IR	0.0953†	0.1822†	0.0726†	0.1003†	0.1285†	0.3194†	0.3681†	0.3361†	0.0305†	0.0909†	0.0326†	0.0510†
	DeCA	0.0974†	0.1855†	0.0758†	0.1162†	0.1293†	0.3076†	0.3575†	0.3270†	0.0337	0.0911†	0.0332†	0.0524
	SGCN	0.0941†	0.1899†	0.0765†	0.1160†	0.1282†	0.3210†	0.3602†	0.3318†	0.0335†	0.0916	**0.0346**	**0.0528**
	SGL	0.0980†	0.1770†	0.0741†	0.0999†	0.1299†	0.3156†	0.3638†	0.3343†	**0.0341**	0.0915	0.0344	0.0526
	SGDL	**0.1134**	**0.2105**	**0.0844**	**0.1178**	**0.1378**	**0.3335**	**0.3844**	**0.3513**	0.0339	**0.0918**	0.0341	0.0525

- SGDL can improve the performance of all base models in all datasets.
- SGDL outperforms all general denoising methods and achieves even better performance than the state-of-the-art graph-based robust learning methods.

Experimental Evaluation (Cont.)

Ablation Study

Database		Adressa				MovieLens				Yelp			
Base Model	**Method**	**R@5**	**R@20**	**N@5**	**N@20**	**R@5**	**R@20**	**N@5**	**N@20**	**R@5**	**R@20**	**N@5**	**N@20**
NeuMF	w/o DLS	0.1528	0.3107	0.1211	0.1794	0.1055	0.2690	0.2911	0.2774	0.0136	0.0397	0.0131	0.0218
	w/o ADS	0.1576	0.3285	0.1255	0.1801	0.1097	0.2801	0.3210	0.3008	0.0146	0.0438	0.0146	0.0259
LightGCN	w/o DLS	0.0964	0.1810	0.0702	0.0985	0.1244	0.3159	0.3688	0.3349	0.0330	0.0909	0.0331	0.0513
	w/o ADS	0.1013	0.1995	0.0811	0.1007	0.1316	0.3328	0.3824	0.3502	0.0338	0.0914	0.0340	0.0521

- Removing denoising learning strategy or the adaptive denoising scheduler will both cause significant performance degradation.

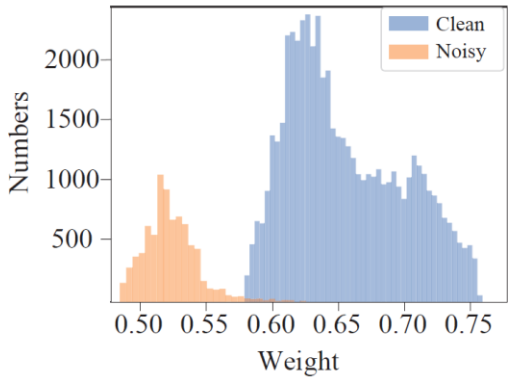
Estimation of Memorization Point

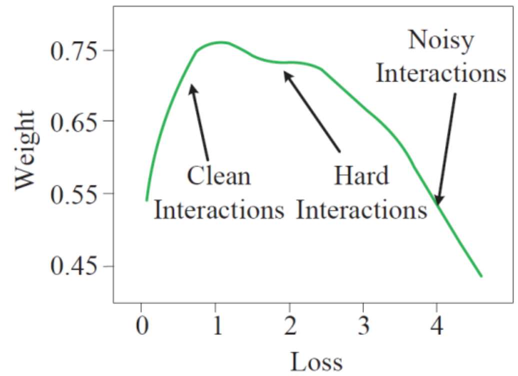
Memorization Point		Early		Est.	Late	
Base Model	**Database**	+10%	+5%	+0%	-5%	-10%
NeuMF	Adressa	0.3221	0.3275	**0.3291**	0.3256	0.3203
	MovieLens	0.2810	**0.2851**	0.2844	0.2757	0.2704
	Yelp	0.458	0.4420	**0.0469**	0.4430	0.4370
LightGCN	Adressa	0.2006	**0.2114**	0.2105	0.2017	0.1990
	MovieLens	0.3321	0.3325	**0.3335**	0.3262	0.3198
	Yelp	0.0895	0.0912	**0.0918**	0.0904	0.0887

- The best performance is achieved near the estimated memorization point.

Experimental Evaluation (Cont.)

Learned Weights of SGDL





- Almost all large weights belong to clean samples.
- The weighting function tends to highlight informative clean samples (including hard samples) and suppress noise interactions.

Outline

- Motivation
- Related Work
- Preliminaries
- Algorithms
- Experimental Evaluation
- **Conclusions**

Conclusions

- A New Perspective. We present a two-stage denoising paradigm which fully leverages the memorization effect of recommendation models
- Self-Guided Denoising Learning. Our proposed SGDL framework can collect memorized data and utilize them as guidance to denoise implicit feedback with a novel adaptive denoising scheduler.
- Adaptivity and Universality. Our method does not need any predefined weighting functions or auxiliary information, and is easy to be implement to any learning-based recommendation models.

Thank you !

Questions?

ytdu@zju.edu.cn

Self-Guided Learning to Denoise for Robust Recommendation

https://github.com/ZJU-DAILY/SGDL

SIGIR 2022