Denoising Implicit Feedback for Recommendation

Wenjie Wang[*], Fuli Feng[*], Xiangnan He[$], Liqiang Nie[#], Tat-Seng Chua[*].

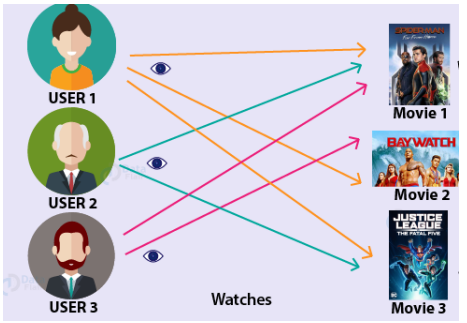
[*]National University of Singapore
[$]University of Science and Technology of China
[#]Shandong University

OUTLINE

- Background
- Related work
- Preliminary
- Method
- Experiment

Background: Noisy Implicit Feedback

- Input: User-Item Interactions
  1. Explicit Feedback (e.g., rating)
  2. Implicit Feedback (e.g., clicks)



- **Large volume** of implicit feedback alleviates the sparsity issue
- Downside is that they are **not as clean** in reflecting the actual user preference
  - e.g., negative reviews, click with quick quit
- **Gap** between implicit feedback and the actual satisfaction of users due to the common existence of noisy interactions

Background: Noisy Implicit Feedback

- Gap between implicit feedback and the actual satisfaction of users due to the common existence of noisy interactions

$\bar{y}_{ui}$: implicit feedback
y^*_{ui}: actual user preference

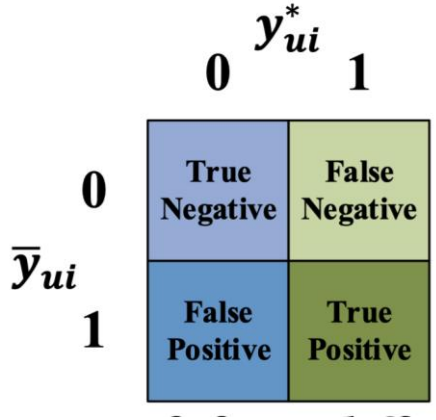


Figure 1. Illustration of four different types of implicit interactions according to the value of user satisfaction (y^*_{ui}) and implicit feedback ($\bar{y}_{ui}$).

Background: Noisy Implicit Feedback

- Negative effects of false-positive interactions
  - Identification of false-positive interactions: auxiliary information of post-interaction behaviors, e.g., dwell time, rating score
  - Normal training: training NeuMF **with** false-positive interactions
  - Clean training: training NeuMF **without** false-positive interactions

Table 1: Results of the clean training and normal training over NeuMF. #Drop denotes the relative performance drop of normal training as compared to clean training.

Dataset	Adressa		Amazon-book	
Metric	Recall@20	NDCG@20	Recall@20	NDCG@20
Clean training	0.4040	0.1963	0.0293	0.0159
Normal training	0.3159	0.1886	0.0265	0.0145
#Drop	21.81%	3.92%	9.56%	8.81%

Related work

1. Negative experience identification
   - Predict false-positive interactions with additional user behaviors (e.g., dwell time and gaze pattern) and auxiliary item features (e.g., textual description)
2. Incorporation of multi-behavior feedback
   - Directly incorporate multi-behavior data into recommenders
   - e.g., favorite and skip patterns

➢ Disadvantage
   1. Sparsity issue of additional user behaviors
   2. Need to collect user satisfaction for each interaction

➢ **This work explores denoising implicit feedback for recommendation without using any additional data**

Preliminary

- False-positive interactions are **harder** to fit in the early training stages

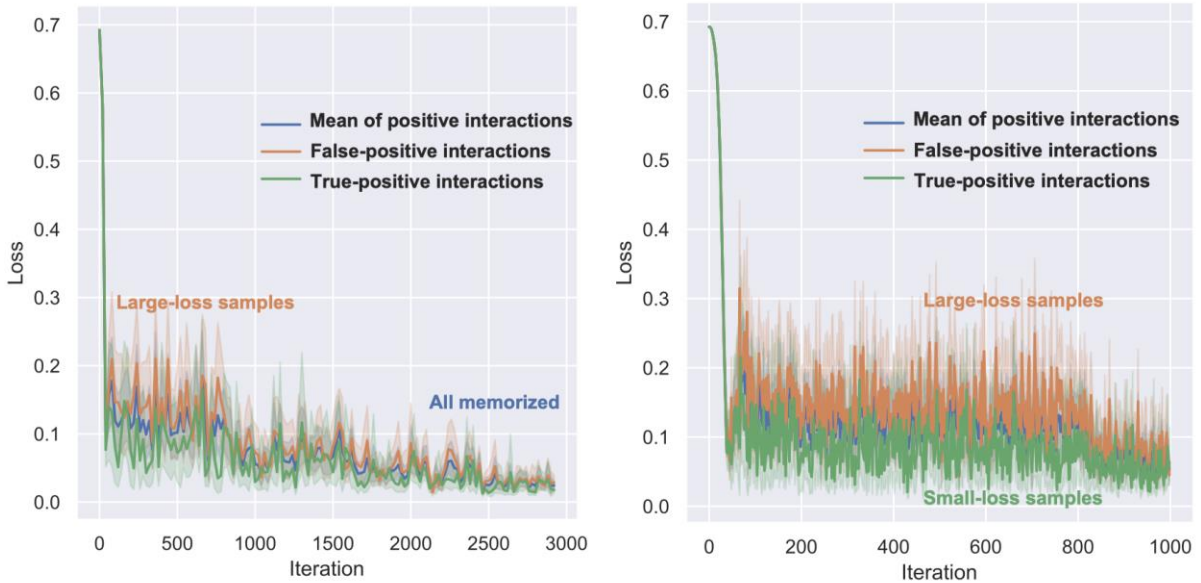


Figure 2. The trend of loss over true- and false-positive interactions in Adressa during the normal training of NeuMF.

Preliminary

- False-positive interactions are **harder** to fit in the early training stages
- Deep models fit easy samples first and then memorize the hard samples

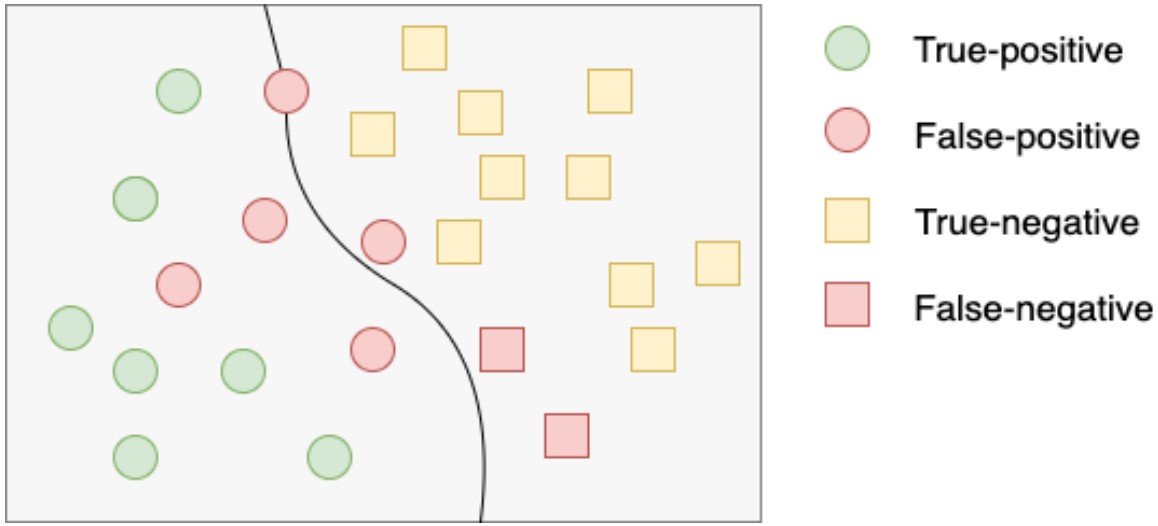


Figure 3. Intuitive illustration of large loss of false-positive interactions.

Method

- Adaptive Denoising Training
  dynamically prunes the large-loss interactions during training

  - Truncated Loss: truncate the loss values of large-loss interactions to 0 with a **dynamic threshold function**

  - Reweighted Loss: It adaptively assigns hard samples (i.e., the large-loss ones) with **smaller weights** during training

Method

- Truncated Loss

$$\mathcal{L}_{\text{T-CE}}(u,i) = \begin{cases} 0, & \mathcal{L}_{CE}(u,i) > \tau \wedge \bar{y}_{ui} = 1 \\ \mathcal{L}_{CE}(u,i), & \text{otherwise,} \end{cases}$$

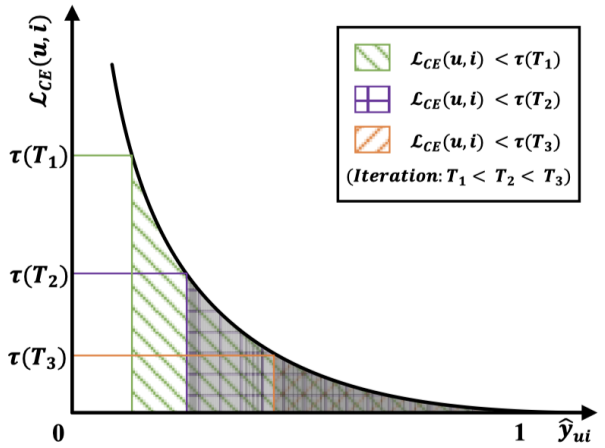


Figure 4: Illustration of T-CE loss for the observed user-item interactions (*i.e.*, samples labeled with $\bar{y}_{ui} = 1$). T_i denotes the iteration number and $\tau(T_i)$ refers to the threshold function. Note that the dash area indicates the effective loss and the loss values larger than $\tau(T_i)$ are truncated.

Method

- Truncated Loss
  - We devise the threshold function as a drop rate function $\epsilon(T)$ since loss values vary across different datasets
  - Drop rate function should have the following properties
    a) upper bound
    b) $\epsilon(0) = 0$
    c) increase smoothly
  - Various functions
    a) linear function
    b) polynomial function
    c) logarithm function

Method

- Reweighted Loss

  down-weight large-loss interactions, which is defined as

$$\mathcal{L}_{R\text{-}CE}(u,i) = \omega(u,i)\mathcal{L}_{CE}(u,i)$$

  In this work, we employ exponential function to formulate the weight function:

$$f(\hat{y}_{ui}) = \hat{y}_{ui}^{\beta},$$

  And two loss functions are instantiated on the cross-entropy loss.

Method

- Reweighted Loss

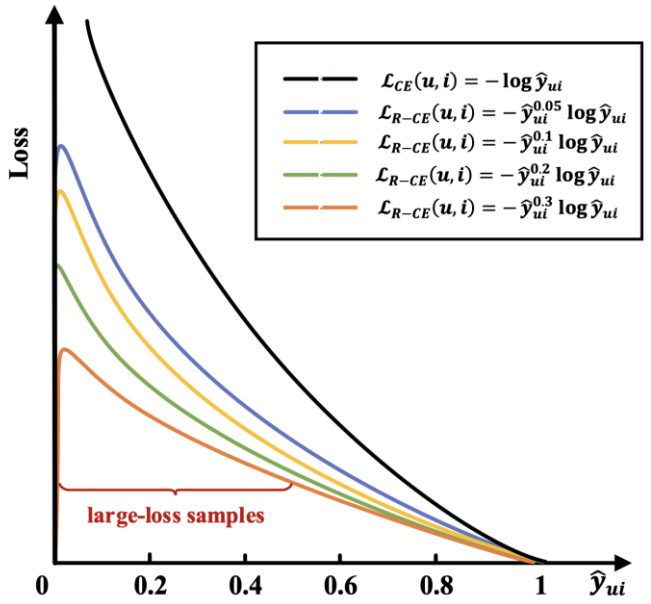


Figure 5. Illustration of R-CE loss for the observed positive interactions.

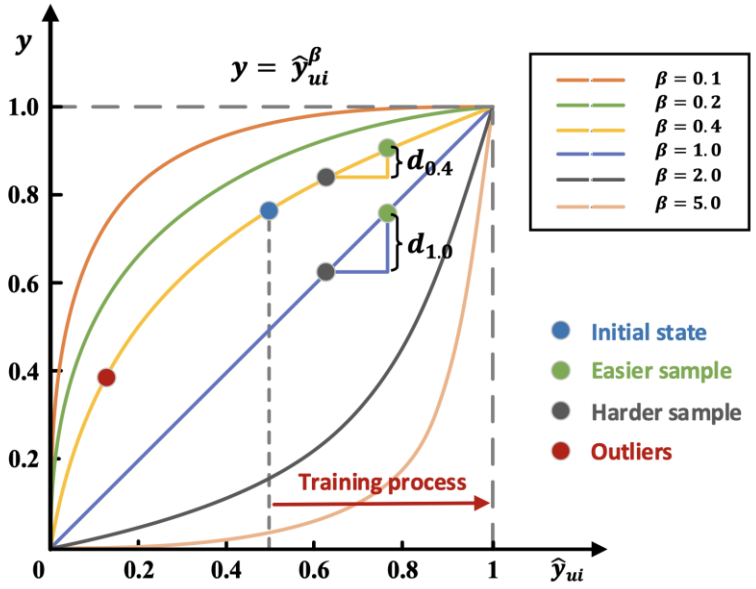


Figure 6: The weight function with different parameters β.

Experiment

Dataset

- Dwell time < 10s as false-positive interactions in the test set
  - Adressa
- Rating score [1-5] < 3 as false-positive interactions in the test set
  - Amazon-book
  - Yelp

Setting

- Training: data with false-positive interactions
- Testing: data without false-positive interactions

Recommenders

- GMF
- NeuMF
- CDAE: CDAE corrupts the observed interactions with random noises, and then employs a MLP model to reconstruct the original interactions, partly increasing its anti-noise capability.

Experiment

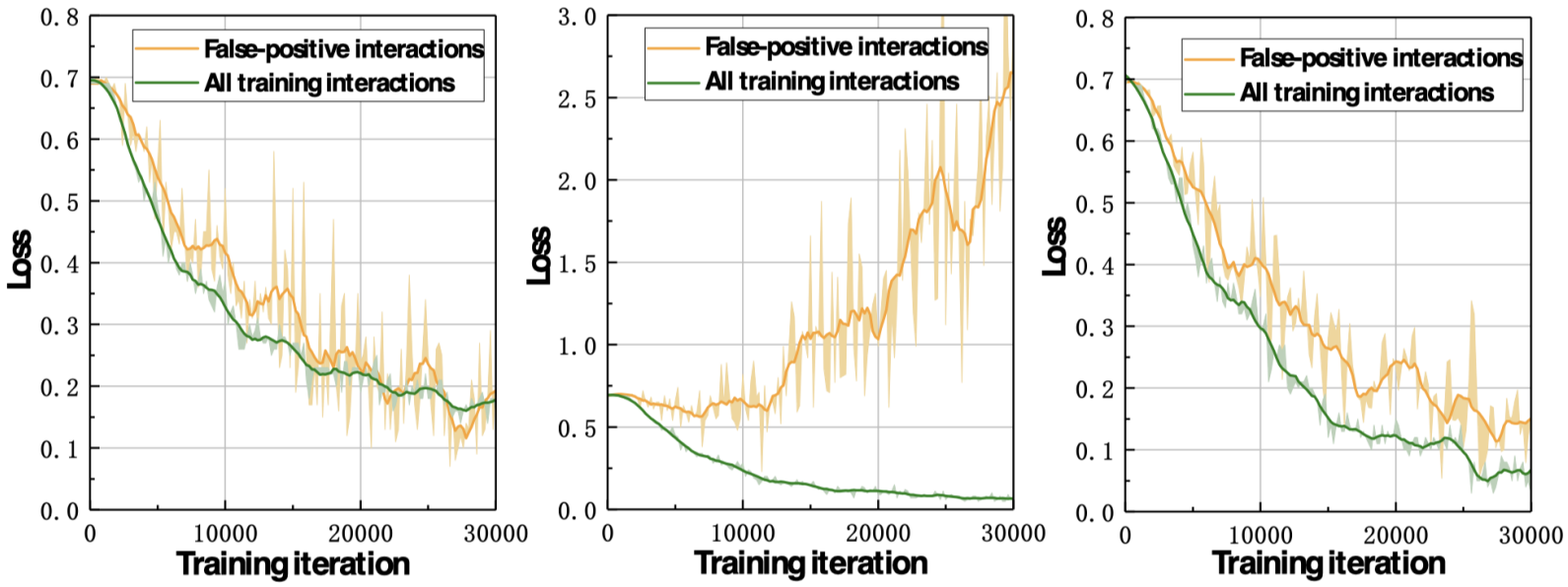
Table 3: Overall performance of three testing recommenders trained with ADT strategies and normal training over three datasets. Note that Recall@K and NDCG@K are shorted as R@K and N@K to save space, respectively, and "RI" in the last column denotes the relative improvement of ADT over normal training on average. The best results are highlighted in bold.

Dataset	Adressa				Amazon-book				Yelp				
Metric	R@3	R@20	N@3	N@20	R@50	R@100	N@50	N@100	R@50	R@100	N@50	N@100	RI
GMF	0.0880	0.2141	0.0780	0.1237	0.0609	0.0949	0.0256	0.0331	0.0840	0.1339	0.0352	0.0465	-
GMF+T-CE	**0.0892**	**0.2170**	**0.0790**	**0.1254**	**0.0707**	**0.1113**	**0.0292**	**0.0382**	**0.0871**	**0.1437**	0.0359	**0.0486**	7.14%
GMF+R-CE	0.0891	0.2142	0.0765	0.1229	0.0682	0.1075	0.0275	0.0362	0.0861	0.1361	**0.0366**	0.0480	4.34%
NeuMF	0.1416	0.3159	0.1267	0.1886	0.0512	0.0829	0.0211	0.0282	0.0750	0.1226	0.0304	0.0411	-
NeuMF+T-CE	**0.1418**	0.3106	0.1227	0.1840	**0.0725**	**0.1158**	**0.0289**	**0.0385**	**0.0825**	**0.1396**	**0.0323**	**0.0451**	15.62%
NeuMF+R-CE	0.1414	**0.3185**	0.1266	**0.1896**	0.0628	0.1018	0.0248	0.0334	0.0788	0.1304	0.0320	0.0436	8.77%
CDAE	0.1394	0.3208	0.1168	0.1808	0.0989	0.1507	0.0414	0.0527	0.1112	0.1732	0.0471	0.0611	-
CDAE+T-CE	**0.1406**	**0.3220**	0.1176	**0.1839**	**0.1088**	**0.1645**	**0.0454**	**0.0575**	**0.1165**	**0.1806**	**0.0504**	**0.0652**	5.36%
CDAE+R-CE	0.1388	0.3164	**0.1200**	0.1827	0.1022	0.1560	0.0424	0.0542	0.1161	0.1801	0.0488	0.0632	2.46%

Observations:

1. Our proposed strategy effectively improves the performance of three testing recommenders over the clean testing set.
2. Truncated Loss performs better in most cases than Reweighted Loss.
3. The strategies achieve the bigger performance increase on NeuMF and GMF than CDAE.

Experiment



(a) Normal Training (b) Truncated Loss (c) Reweighted Loss

Figure 7: Loss comparison of false-positive interactions between Normal Training (a), Truncated Loss (b) and Reweighted Loss (c).

Experiment

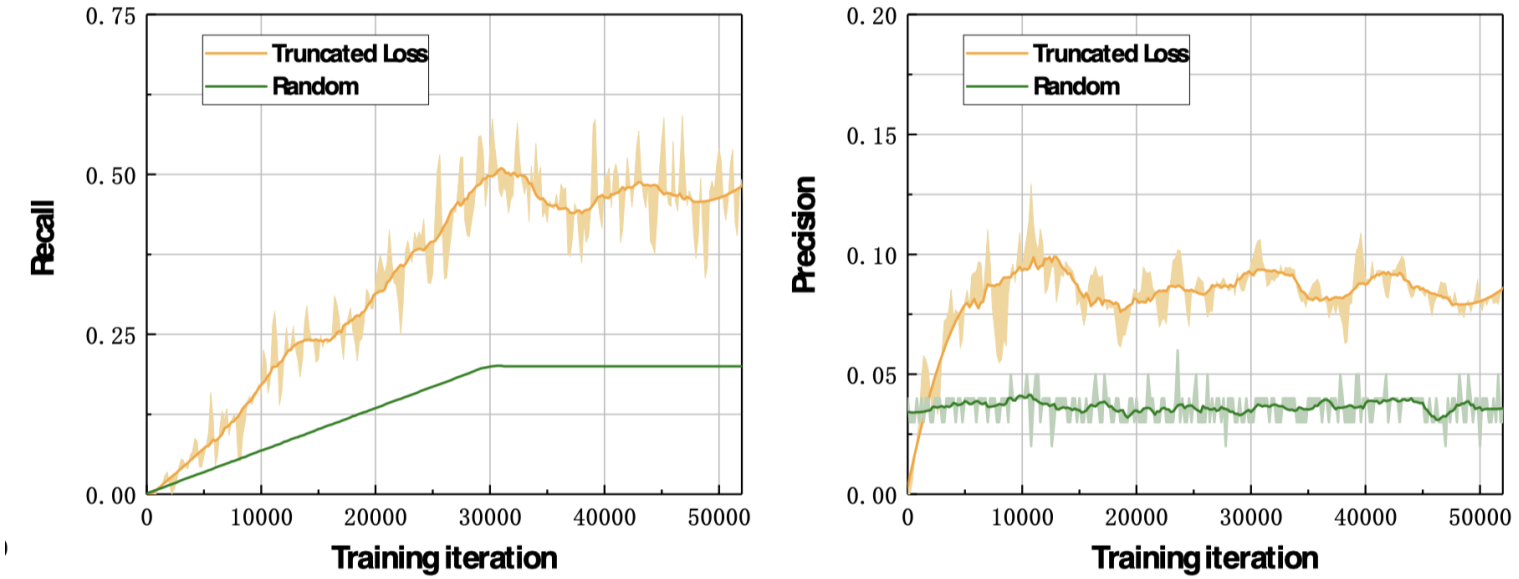


Figure 8: Recall and precision of false-positive interactions over GMF trained the Truncated Loss on Amazon-book.

References

1. Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In Proceedings of the 34th International Conference on Machine Learning. PMLR, 233–242.
2. Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In Proceedings of the 32nd International Conference on Neural Information Processing Systems. MIT Press, Curran Associates Inc., 8527–8537.
3. Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web. IW3C2, 173–182.
4. Youngho Kim, Ahmed Hassan, Ryen W White, and Imed Zitouni. 2014. Modeling dwell time to predict click-level satisfaction. In Proceedings of the 7th ACM international conference on Web search and data mining. ACM, 193–202.
5. Hongyu Lu, Min Zhang, and Shaoping Ma. 2018. Between Clicks and Satisfaction: Study on Multi-Phase User Preferences and Satisfaction for Online News Reading. In Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 435–444.
6. Hongyi Wen, Longqi Yang, and Deborah Estrin. 2019. Leveraging Post-click Feedback for Content Recommendations. In Proceedings of the 13th ACM Conference on Recommender Systems. ACM, 278–286.
7. Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In Proceedings of the 9th ACM International Conference on Web Search and Data Mining. ACM, 153–162.
8. Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. 2014. Beyond clicks: dwell time for personalization. In Proceedings of the 8th ACM Conference on Recommender systems. ACM, 113–120.
9. Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530.
10. Jon Atle Gulla, Lemei Zhang, Peng Liu, Özlem Özgöbek, and Xiaomeng Su. 2017. The Adressa Dataset for News Recommendation. In Proceedings of the International Conference on Web Intelligence. ACM, 1042–1048.

THANK YOU

NExT++ research is supported by the National Research Foundation,
Prime Minister's Office, Singapore under its IRC@SG Funding Initiative.

Contact: Wenjie Wang
Email: wenjiewang96@gmail.com