NEURAL INFORMATION PROCESSING SYSTEMS

Infinite Recommendation Networks

A Data-Centric Approach

Question: Is **more data** what you need for **better recommendation**?

Noveen Sachdeva , Mehak Preet Dhaliwal , Carole-Jean Wu , Julian McAuley

@noveens97

UC San Diego & Meta AI

NeurIPS, 2022

∞-AE

Infinite-width Autoencoder for Recommendation

Premise: Does stretching the hidden layers of an autoencoder till ∞ help in better recommendation?

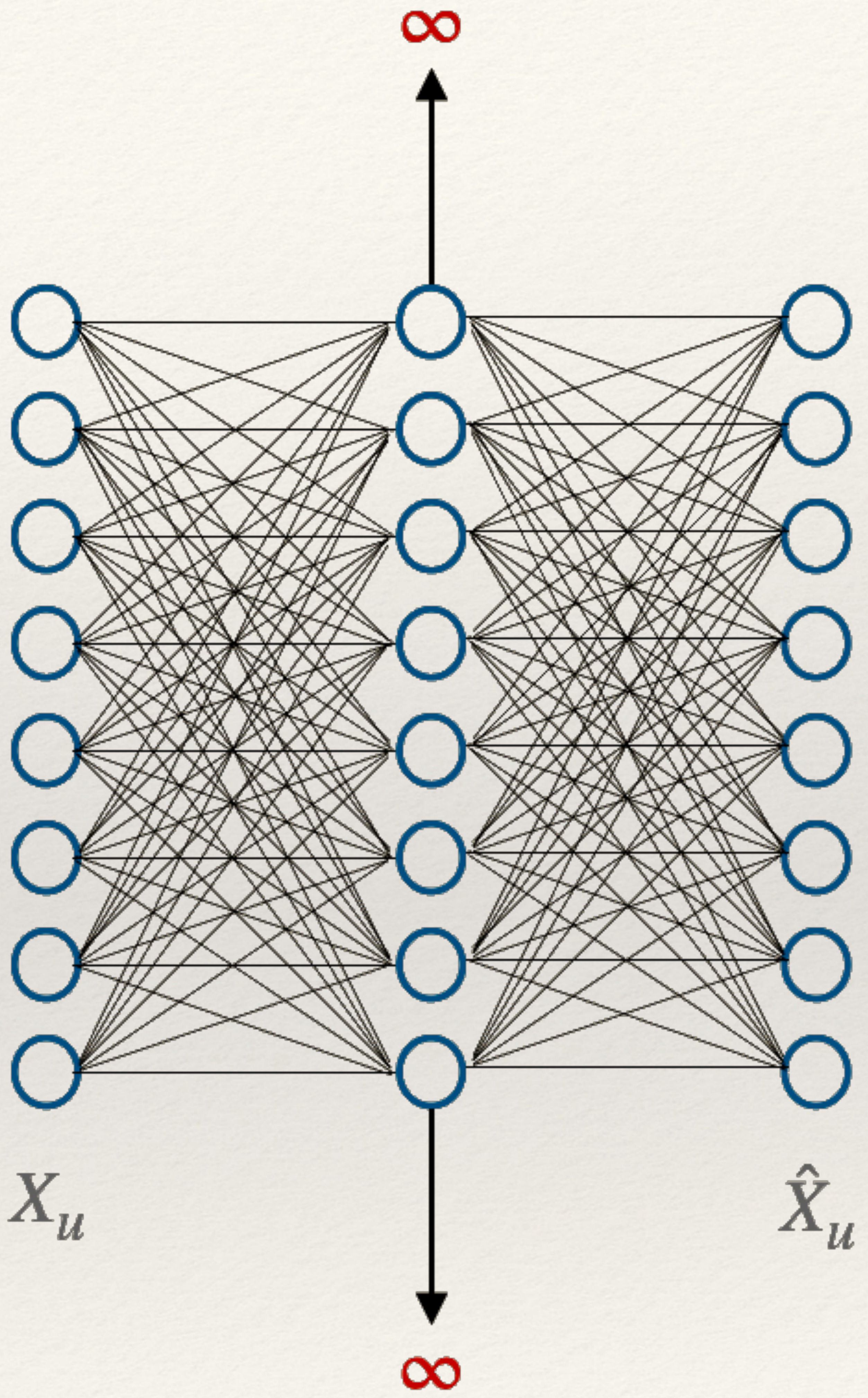
∞-AE

Methodology

- **Infinite-width Correspondence:** Performing Kernelized Ridge Regression with the Neural Tangent Kernel (NTK) emulates the training of an infinite-width NN for an infinite number of SGD steps.
- X_u is the bag-of-items representation for user u i.e. all the items that u interacted with, and we aim to reconstruct it along with **missing user preferences**
- Due to the infinite-width correspondence, ∞-AE **optimizes in closed-form**:

$$\hat{X} = K \cdot (K + \lambda I)^{-1} \cdot X \quad \text{s.t.} \quad K_{u,v} := \mathbb{K}(X_u, X_v) \quad \forall u, v$$

- The optimization has only a **single hyper-parameter** λ
- **Time complexity** Training: $\mathcal{O}(U^2 \cdot I + U^{2.376})$ Inference: $\mathcal{O}(U \cdot I)$
- **Memory complexity** Training: $\mathcal{O}(U \cdot I + U^2)$ Inference: $\mathcal{O}(U \cdot I)$



∞-AE

Experiments

Dataset	NeuMF	GCN	MVAE	EASE	∞-AE
Magazine	13.6	22.5	12.1	22.8	**23.0**
ML-1M	25.6	28.8	22.1	29.8	**32.8**
Douban	13.3	16.6	16.1	19.4	**24.9**
Netflix	12.0	—	20.8	26.8	**30.5***

Table 1: nDCG@10 performance (higher is better) of various recommendation algorithms.
* represents training on 5% random users.

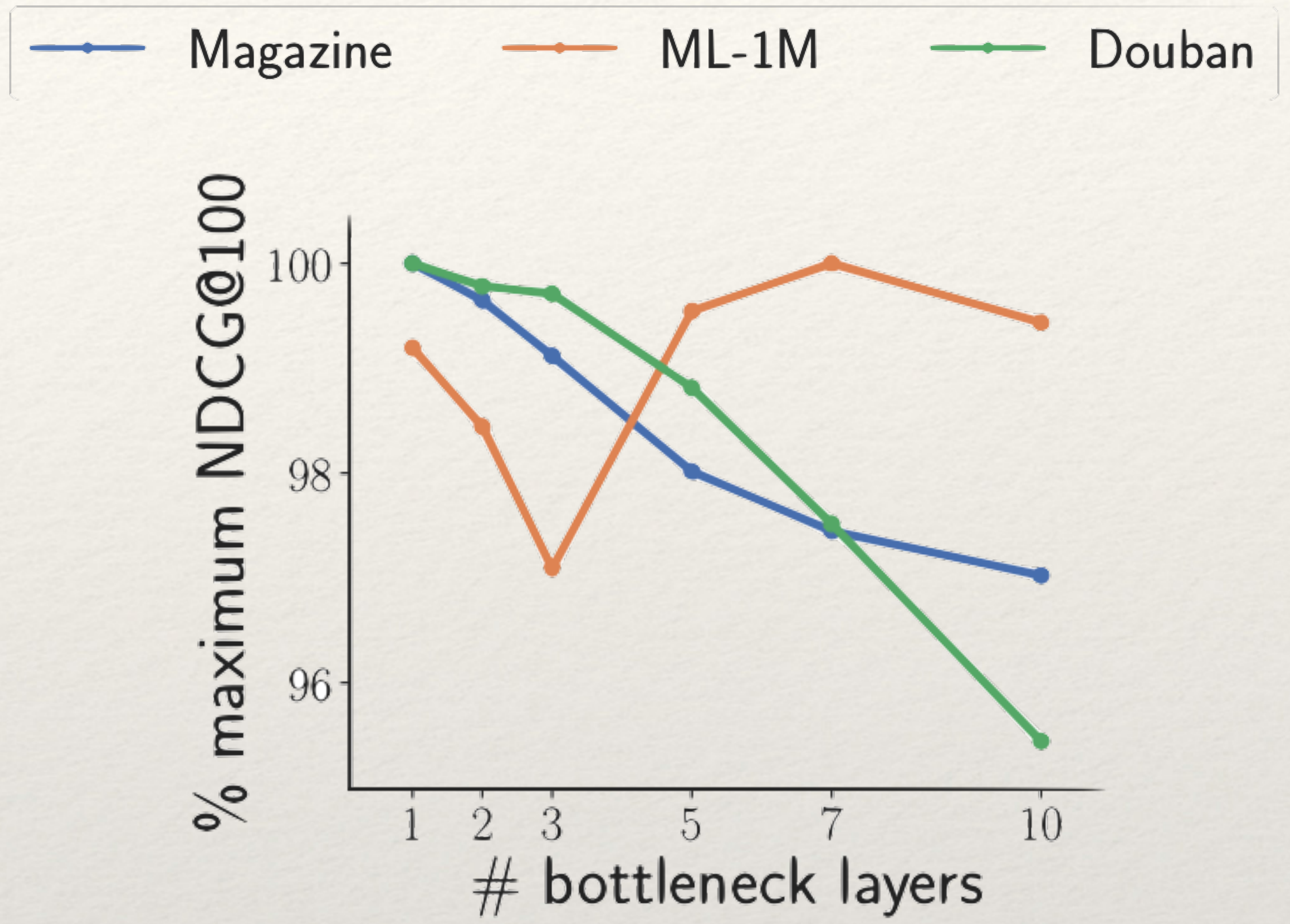


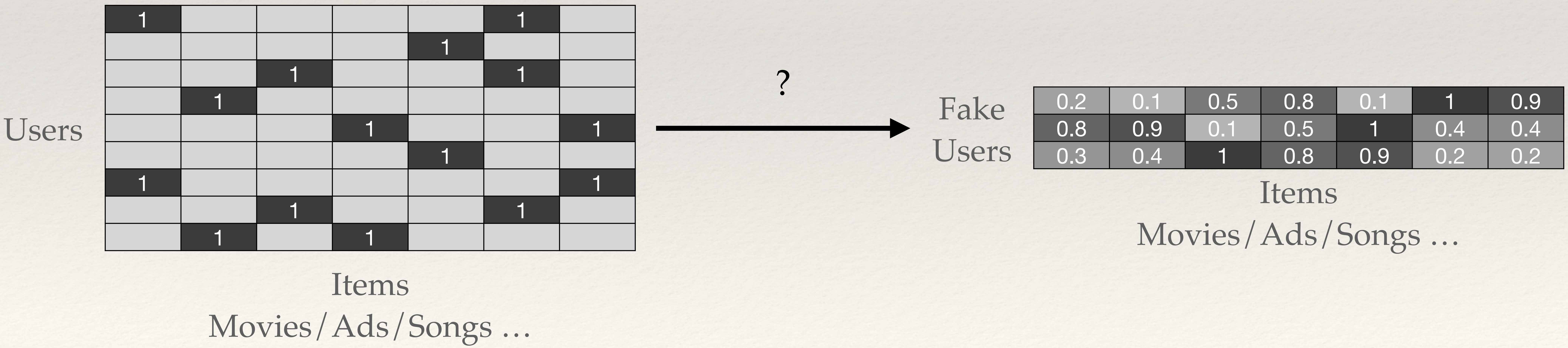
Figure 2: Performance of ∞-AE with varying depth.

- ∞-AE outperforms various state-of-the-art methods, even when trained on just 5% random users
- 1 layer seems to be enough for optimal recommendation performance: common folk-knowledge
- Even though the model is expensive; it is simplistic, easy to implement (thanks, `JAX`), and the performance is great! But how to scale it up? 🤔

Distill-CF

Data Distillation for Collaborative Filtering Data

Idea: Treat the to-be-synthesized data as **parameters**, and **learn** them through a bilevel optimization.



Distill-CF

Overview & Challenges

- Challenges:
    - Data consists of **discrete** (u, i, r) tuples: how to optimize?
    - Data is typically extremely **sparse**
    - **Dynamic popularity**: some users/items are more popular than others
    - Expensive **bilevel optimization**
        - Use ∞-AE for closed-form computation of the inner loop
- **Optimizes** for data-quality rather than quantity

Outer loop — optimize the support set for a fixed learning algorithm

Support dataset

Differentiable cost-function

$$\arg\min_{\boxed{\mathcal{D}^s}} \mathop{\mathbb{E}}_{(u,i,r)\sim\mathcal{D}^{\text{val}}} \left[\boxed{l}\left(\phi^*_{\mathcal{D}^s}(u,i),r\right)\right]$$

$$\text{s.t.} \quad \boxed{\phi^*_{\mathcal{D}^s}} := \arg\min_{\phi} \mathop{\mathbb{E}}_{(u,i,r)\sim\mathcal{D}^s} \left[l(\phi(u,i),r)\right]$$

Optimal recommendation algorithm trained on $\mathcal{D}^s$

Inner loop — optimize the learning algorithm for a fixed support set

Distill-CF

Experiments

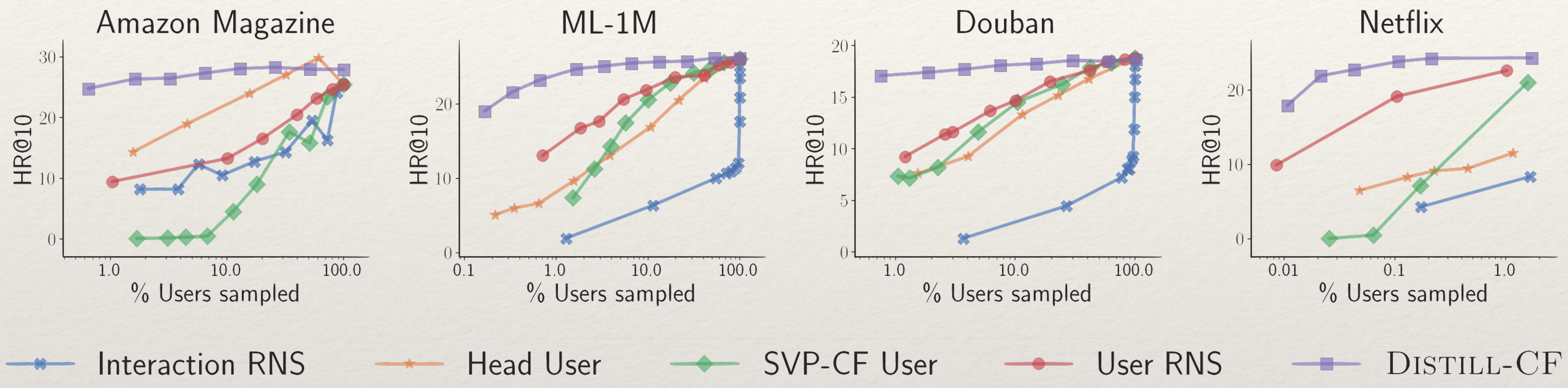


Figure 3: Does Distill-CF outperform other samplers? (Log-scale)

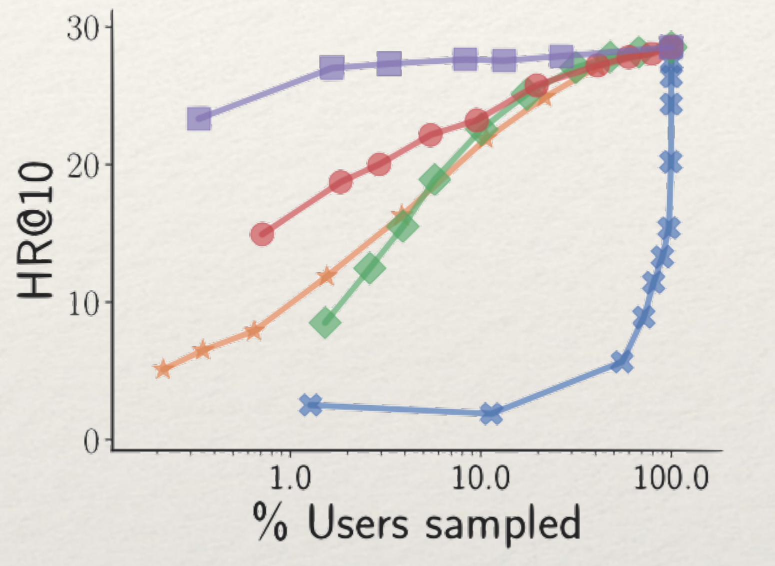


Figure 4: Distill-CF + EASE for the ML-1M dataset.

- Using Distill-CF, we can get **96-105%** of full-data performance on as small as **0.1%** data sub-samples, leading to as much as ~**1000x** time speedup!
- Distill-CF works well for the second-best EASE model, even though data was optimized for ∞-AE

Thank you!

@noveens97

For paper, code, and these slides:

noveens.com