Data Science and Engineering Lab

Fundamentals of Deep Recommender Systems

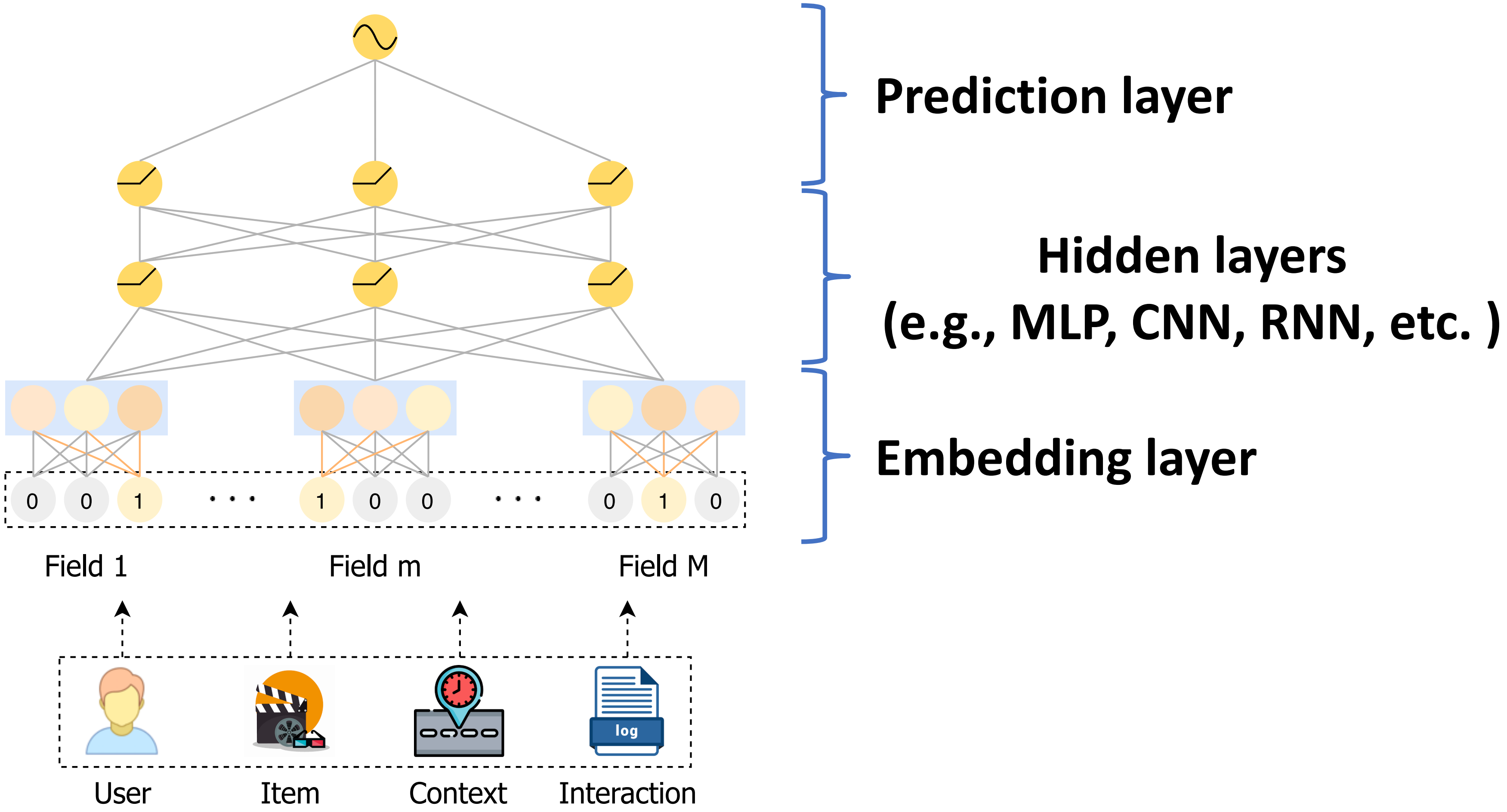
Wenqi Fan

The Hong Kong Polytechnic University

https://wenqifan03.github.io, wenqifan@polyu.edu.hk

Tutorial website: https://advanced-recommender-systems.github.io/ijcai2021-tutorial/

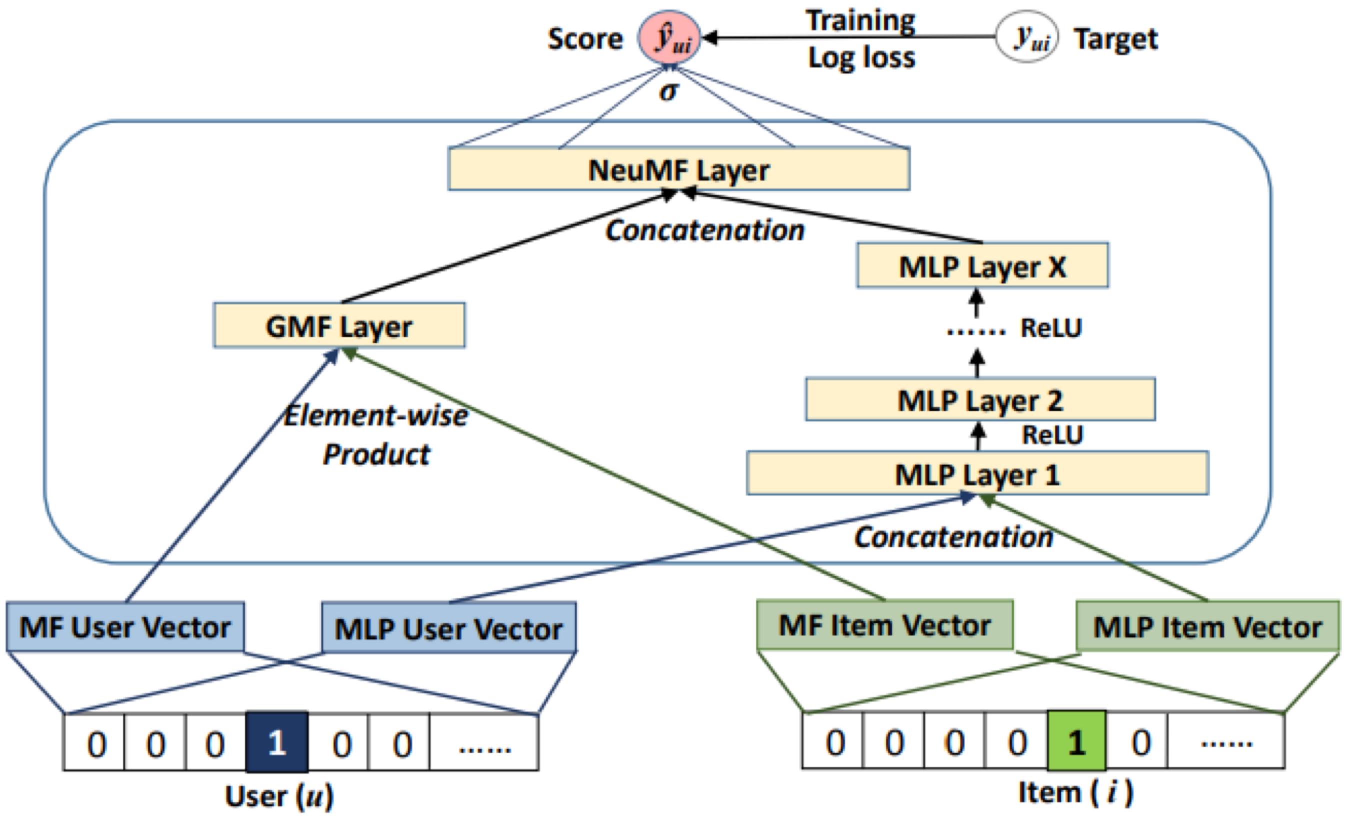
A General Architecture of Deep Recommender System



NeuMF

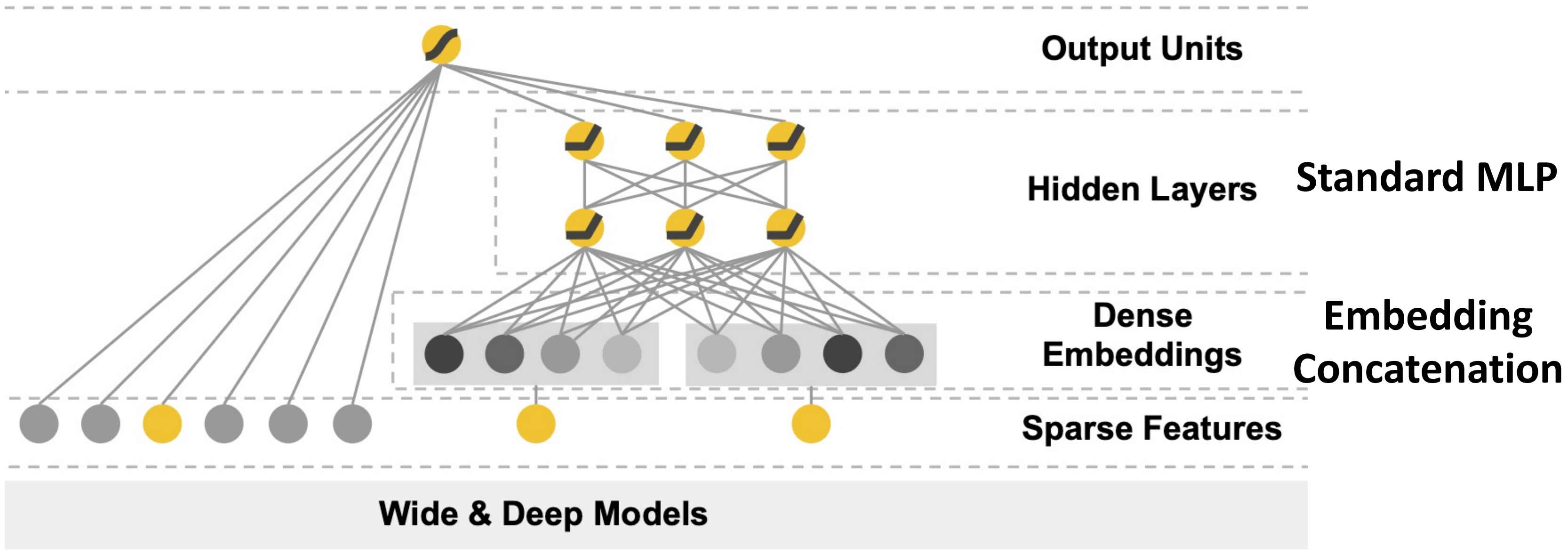
NeuMF unifies the strengths of MF and MLP in modeling user-item interactions.

- **MF** uses an inner product as the interaction function
- **MLP** is more sufficient to capture the complex structure of user interaction data



Neural Collaborative Filtering, WWW, 2017

Wide&Deep

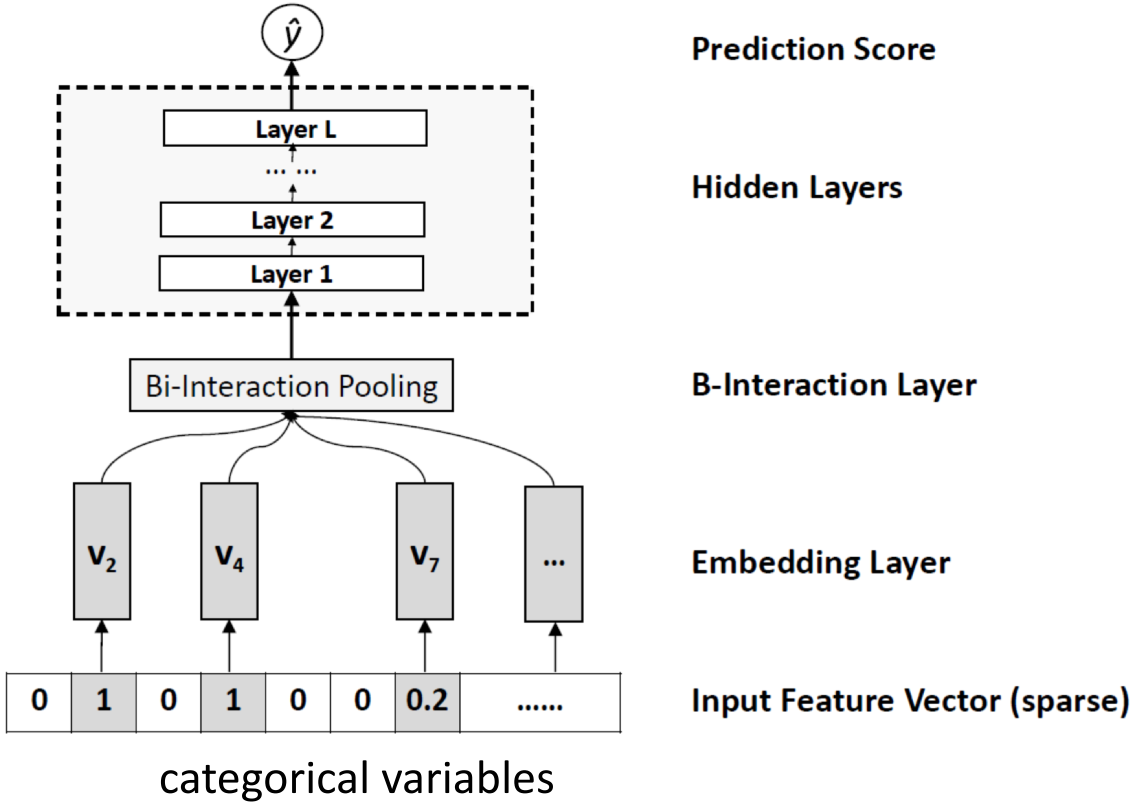


- The **wide linear models** can memorize seen feature interactions using cross-product feature transformations.
- The **deep models** can generalize to previously unseen feature interactions through low- dimensional embeddings.

Wide & Deep Learning for Recommender Systems, 1st DLRS, 2016

Neural FM

Neural **Factorization Machines** (NFMs) “deepens” FM by placing hidden layers above second-order **feature interaction** modeling.



Neural Factorization Machines for Sparse Predictive Analytics, SIGIR, 2017

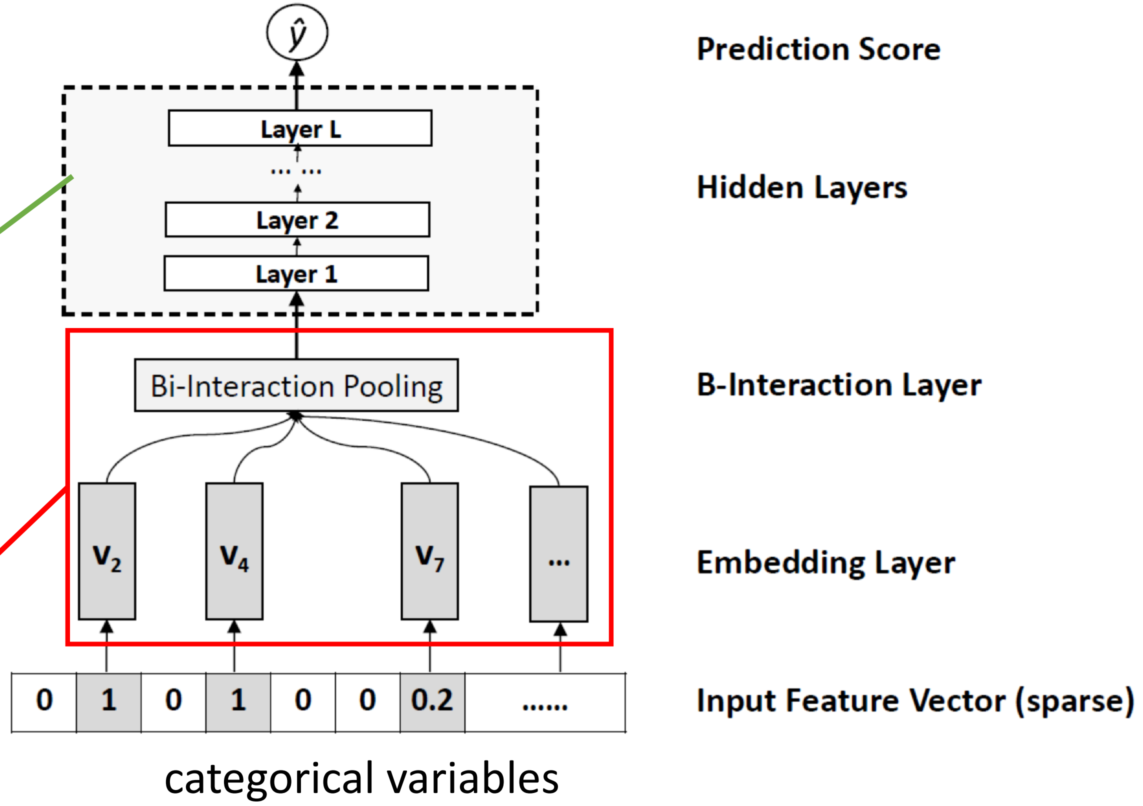
Neural FM

Neural Factorization Machines (NFMs) "deepens" FM by placing hidden layers above second-order **feature interaction** modeling.

"Deep layers" learn **higher-order** feature interactions only, being much easier to train.

Bilinear Interaction Pooling:

$$f_{BI}(\mathcal{V}_x) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i \mathbf{v}_i \odot x_j \mathbf{v}_j$$

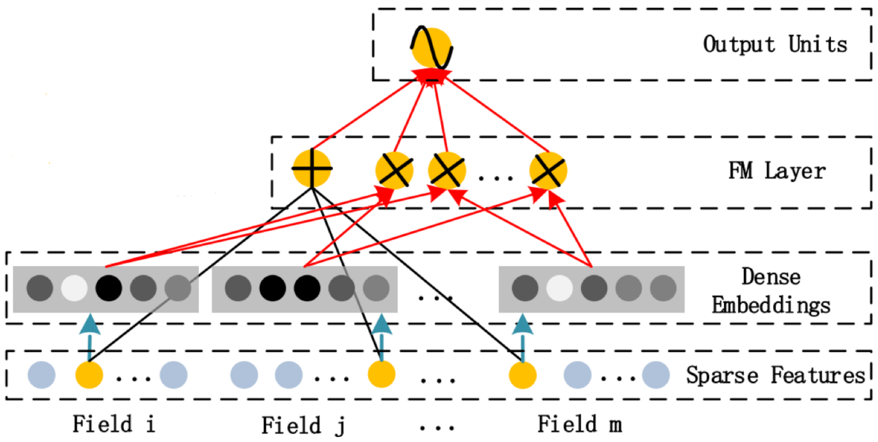


Neural Factorization Machines for Sparse Predictive Analytics, SIGIR, 2017

DeepFM

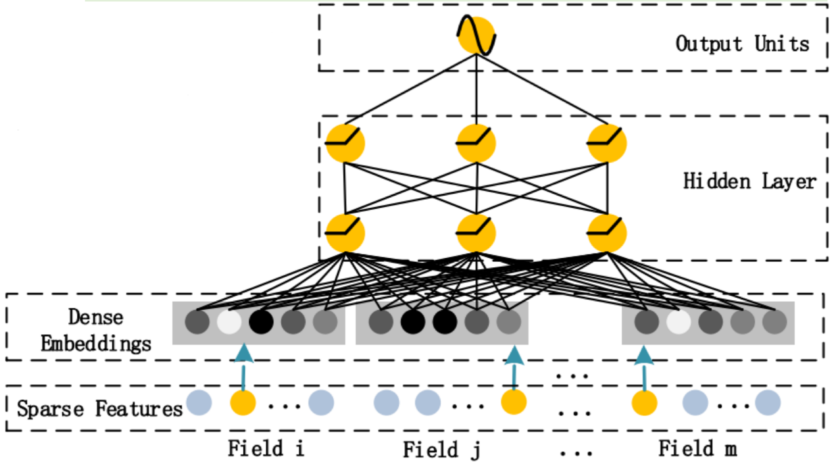
DeepFM ensembles FM and DNN and to low- and high-order feature interactions simultaneously from the input raw features.

FM component (low-order)



$$y_{FM} = \langle w, x \rangle + \sum_{j_1=1}^{d} \sum_{j_2=j_1+1}^{d} \langle V_i, V_j \rangle x_{j_1} \cdot x_{j_2}$$

Deep component (high-order)



$$a^{(0)} = [e_1, e_2, ..., e_m]$$
$$a^{(l+1)} = \sigma(W^{(l)} a^{(l)} + b^{(l)})$$
$$y_{DNN} = \sigma(W^{|H|+1} \cdot a^H + b^{|H|+1})$$

Prediction Model: $\hat{y} = sigmoid(y_{FM} + y_{DNN})$

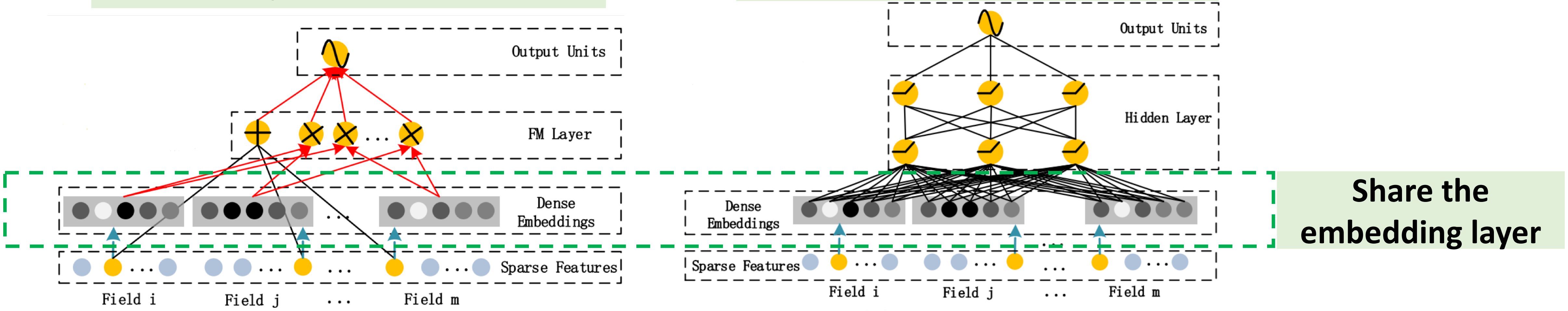
DeepFM: A Factorization-Machine based Neural Network for CTR Prediction, IJCAI, 2017

DeepFM

DeepFM ensembles FM and DNN and to low- and high-order feature interactions simultaneously from the input raw features.

FM component (low-order)

Deep component (high-order)



$$y_{FM} = \langle w, x \rangle + \sum_{j_1=1}^{d} \sum_{j_2=j_1+1}^{d} \langle V_i, V_j \rangle x_{j_1} \cdot x_{j_2}$$

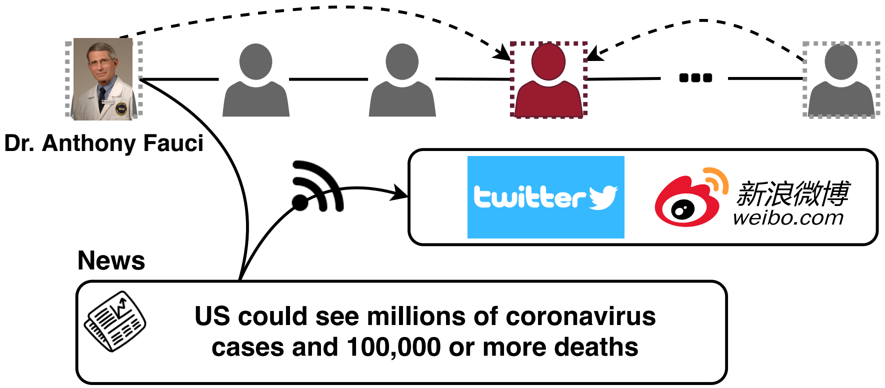
$$a^{(0)} = [e_1, e_2, ..., e_m]$$
$$a^{(l+1)} = \sigma(W^{(l)} a^{(l)} + b^{(l)})$$
$$y_{DNN} = \sigma(W^{|H|+1} \cdot a^H + b^{|H|+1})$$

Prediction Model: $\hat{y} = sigmoid(y_{FM} + y_{DNN})$

DeepFM: A Factorization-Machine based Neural Network for CTR Prediction, IJCAI, 2017

DSCF

Collaborative Filtering with users' social relations
(**Social Recommendation**)



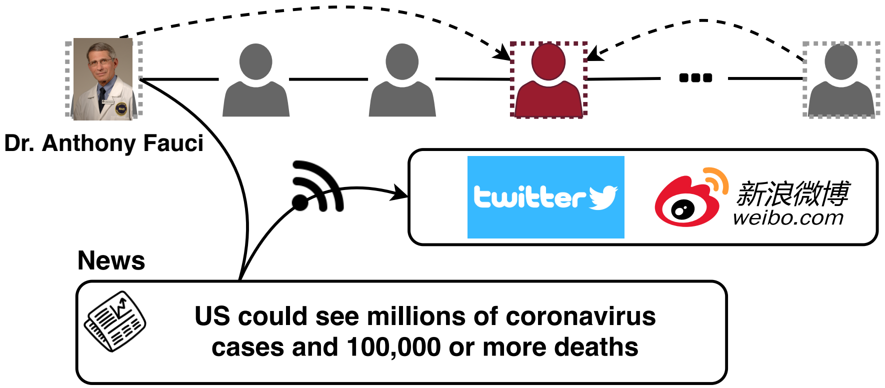
Deep Social Collaborative Filtering, RecSys, 2019

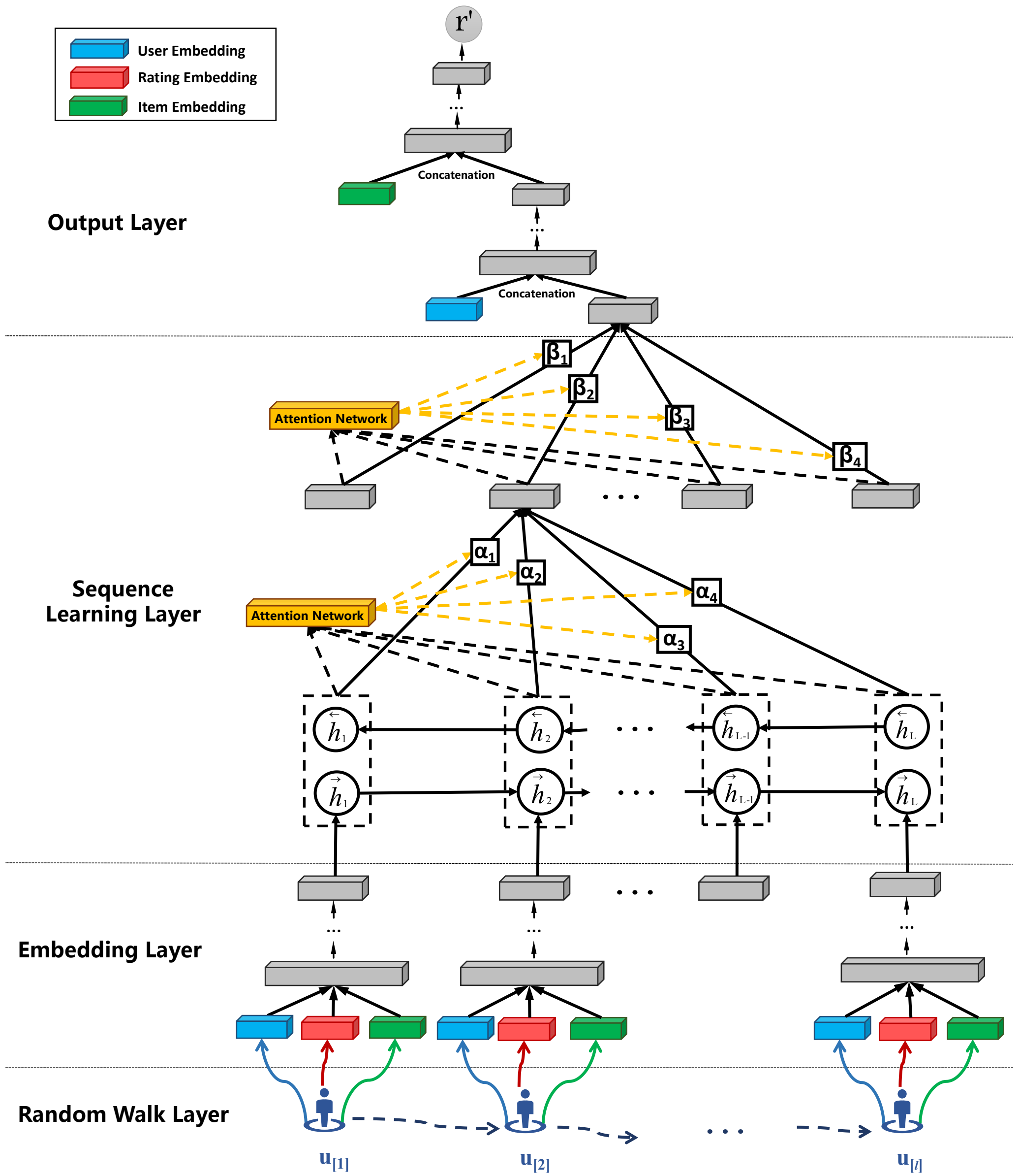
DSCF

Collaborative Filtering with users' social relations
(**Social Recommendation**)

Users might be affected by direct/distant neighbors.

- Information diffusion
- Users with high reputations





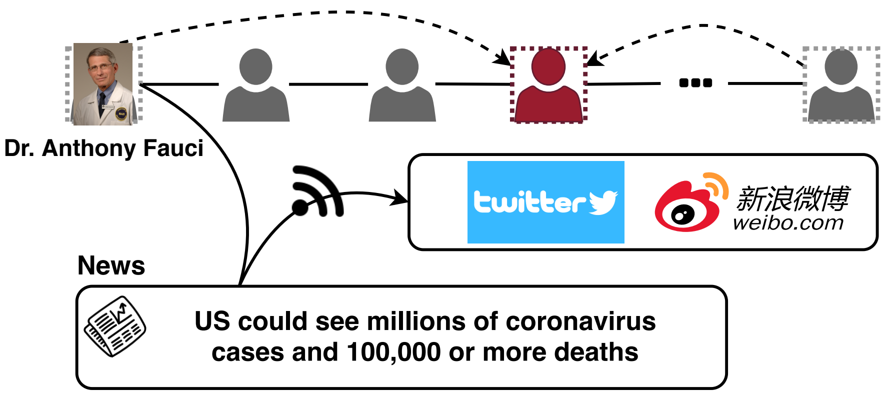
Deep Social Collaborative Filtering, RecSys, 2019

DSCF

Collaborative Filtering with users' social relations
(**Social Recommendation**)

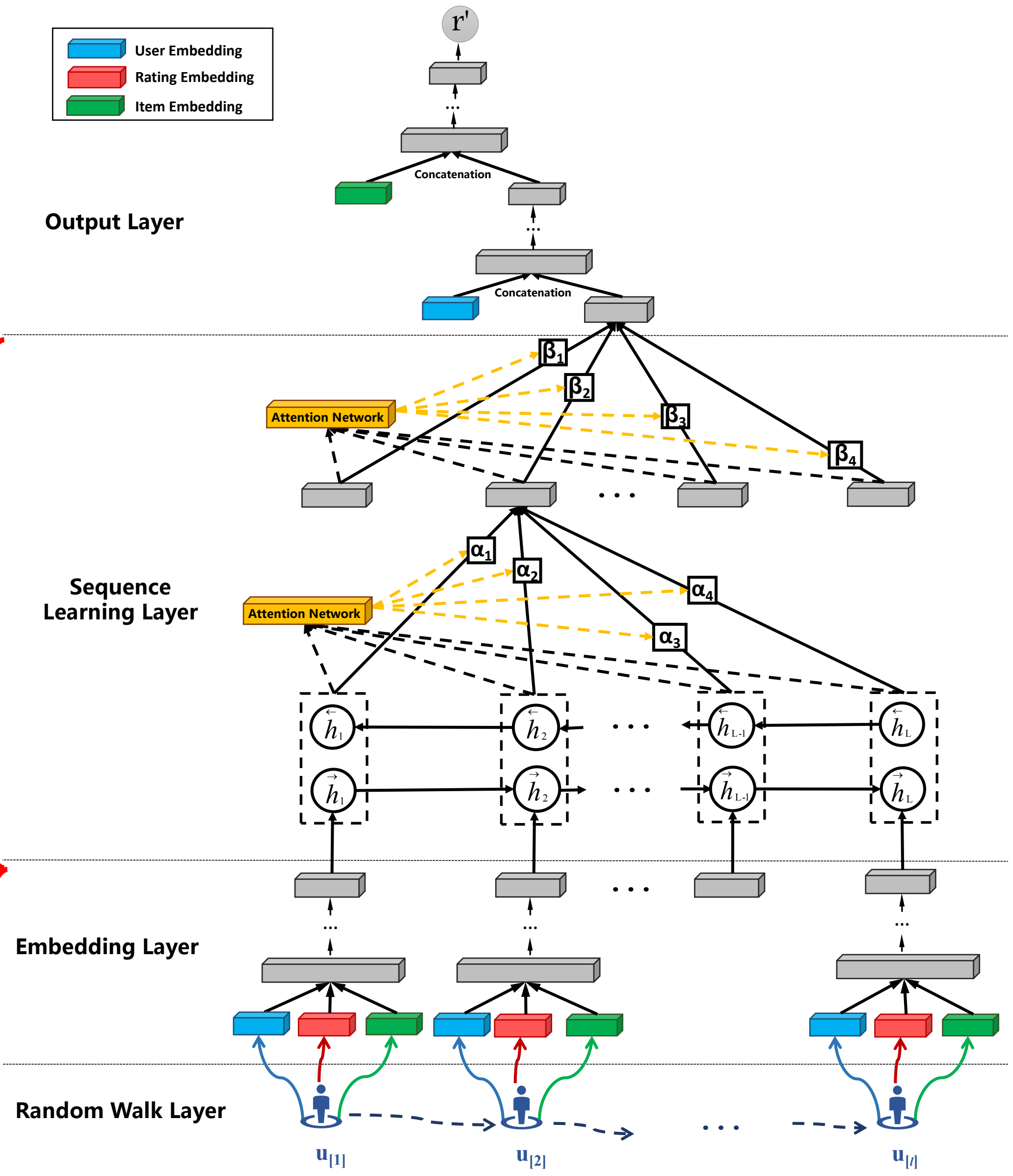
Users might be affected by direct/distant neighbors.

- Information diffusion
- Users with high reputations



Bi-LSTM with attention mechanisms

Social Sequences via Random Walk techniques

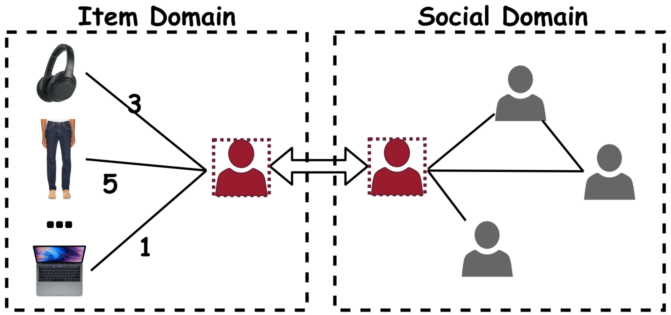


Deep Social Collaborative Filtering, RecSys, 2019

DASO

Collaborative Filtering with users' social relations
(**Social Recommendation**)

- **User behave and interact differently in the item/social domains.**



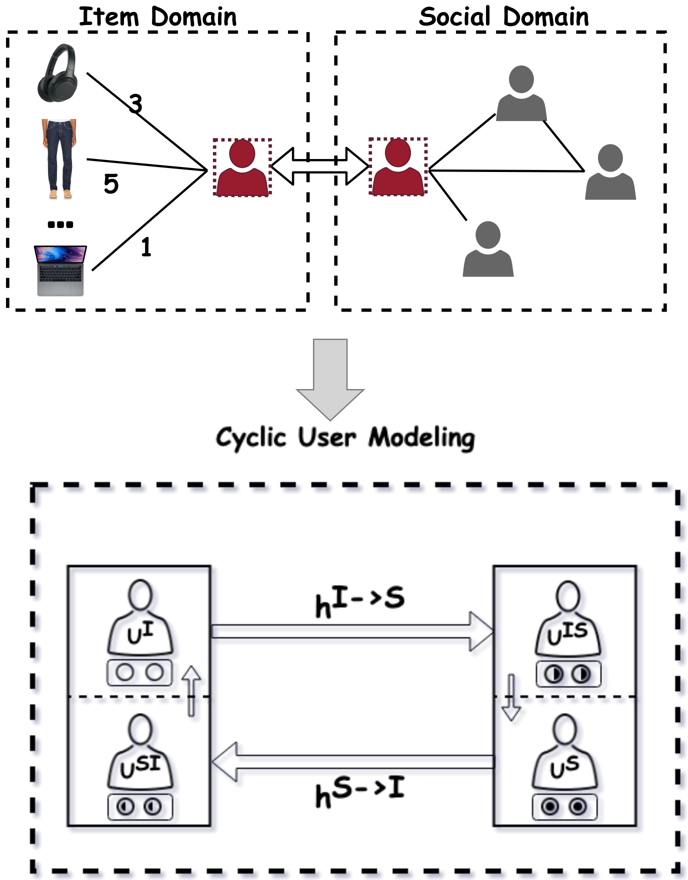
Deep Adversarial Social Recommendation, IJCAI, 2019

DASO

Collaborative Filtering with users' social relations
(**Social Recommendation**)

- **User behave and interact differently in the item/social domains.**

Learning separated user representations in two domains.



Deep Adversarial Social Recommendation, IJCAI, 2019

DASO

Collaborative Filtering with users' social relations
(**Social Recommendation**)

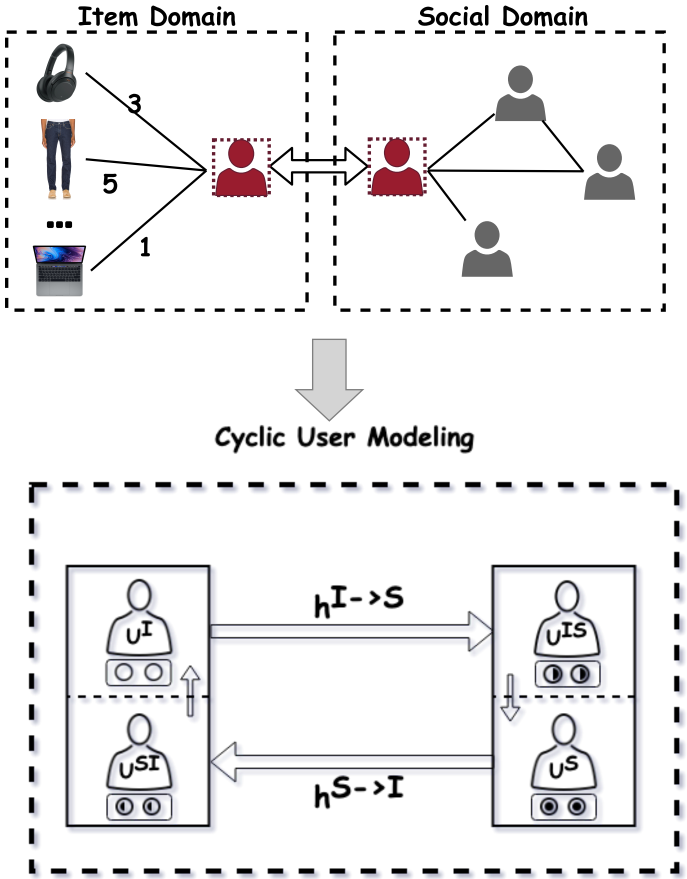
- **User behave and interact differently in the item/social domains.**

Learning separated user representations in two domains.

Bidirectional Knowledge Transfer with Cycle Reconstruction

$$\boxed{\mathbf{p}_i^I} \rightarrow h^{I\rightarrow S}(\mathbf{p}_i^I) \rightarrow h^{S\rightarrow I}(h^{I\rightarrow S}(\mathbf{p}_i^I)) \approx \boxed{\mathbf{p}_i^I}$$

$$\mathcal{L}_{cyc}(h^{S\rightarrow I}, h^{I\rightarrow S}) = \sum_{i=1}^{N}\left(\left\|h^{S\rightarrow I}(h^{I\rightarrow S}(\mathbf{p}_i^I)) - \mathbf{p}_i^I\right\|_2 + \left\|h^{I\rightarrow S}(h^{S\rightarrow I}(\mathbf{p}_i^S)) - \mathbf{p}_i^S\right\|_2\right)$$



Deep Adversarial Social Recommendation, IJCAI, 2019

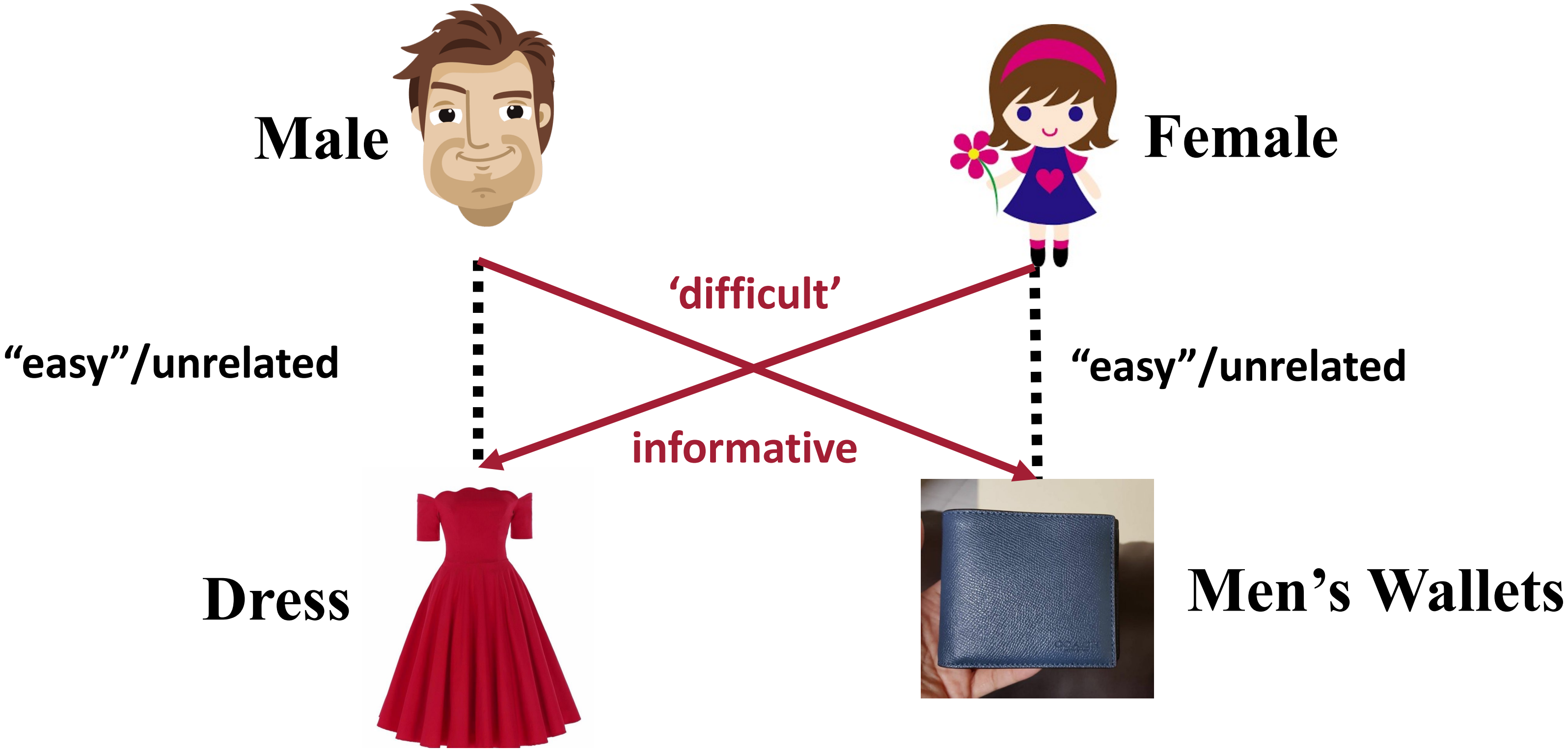
Optimization for Ranking Tasks

☐ Negative Sampling's Main Issue:

- It often generates low-quality negative samples that do not help you learn good representation.

Optimization for Ranking Tasks

- **Negative Sampling's Main Issue:**
  - It often generates low-quality negative samples that do not help you learn good representation [Cai and Wang, 2018; Wang *et al.*, 2018b].

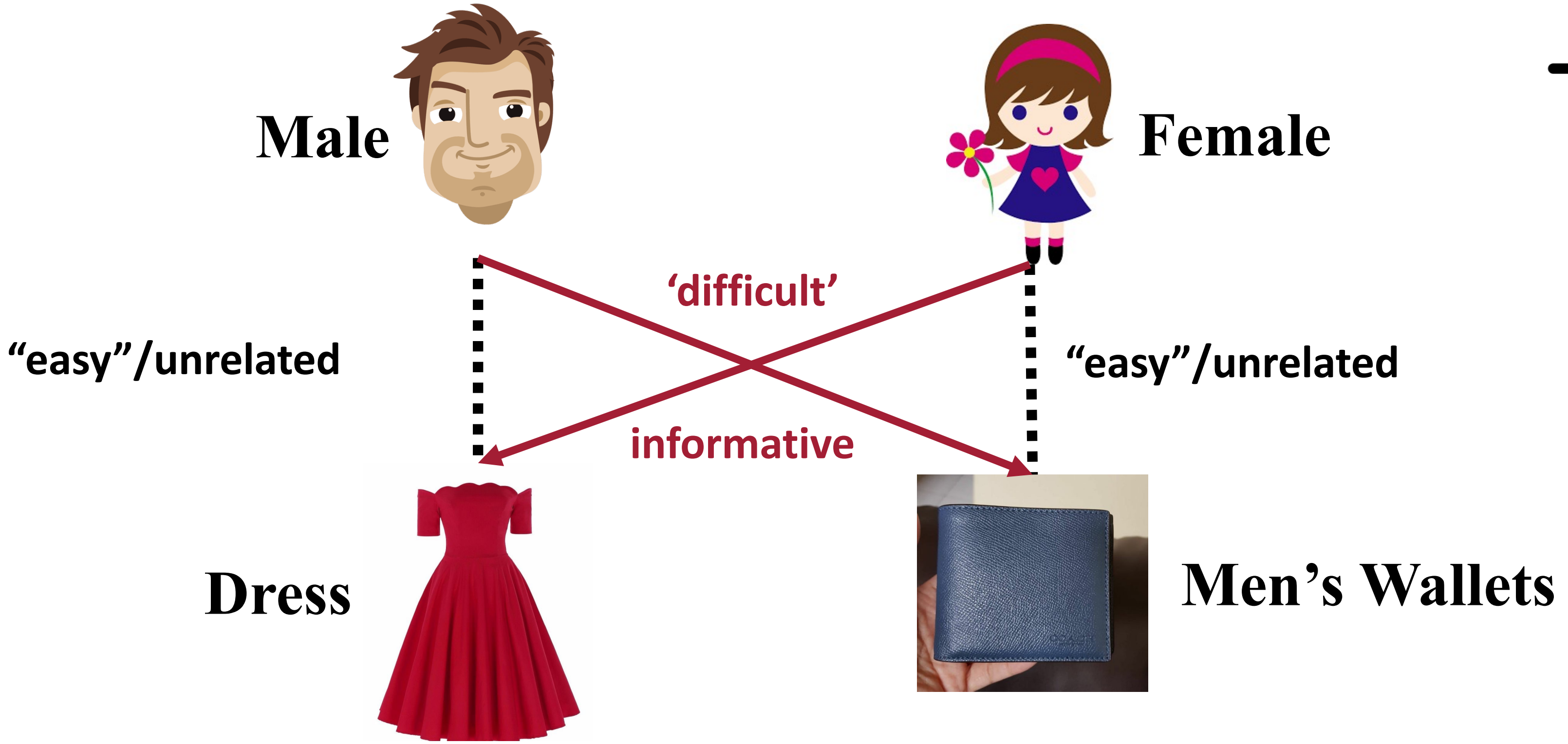


Deep Adversarial Social Recommendation, IJCAI, 2019

Optimization for Ranking Tasks

☐ Negative Sampling’s Main Issue:

- It often generates low-quality negative samples that do not help you learn good representation [Cai and Wang, 2018; Wang *et al.*, 2018b].

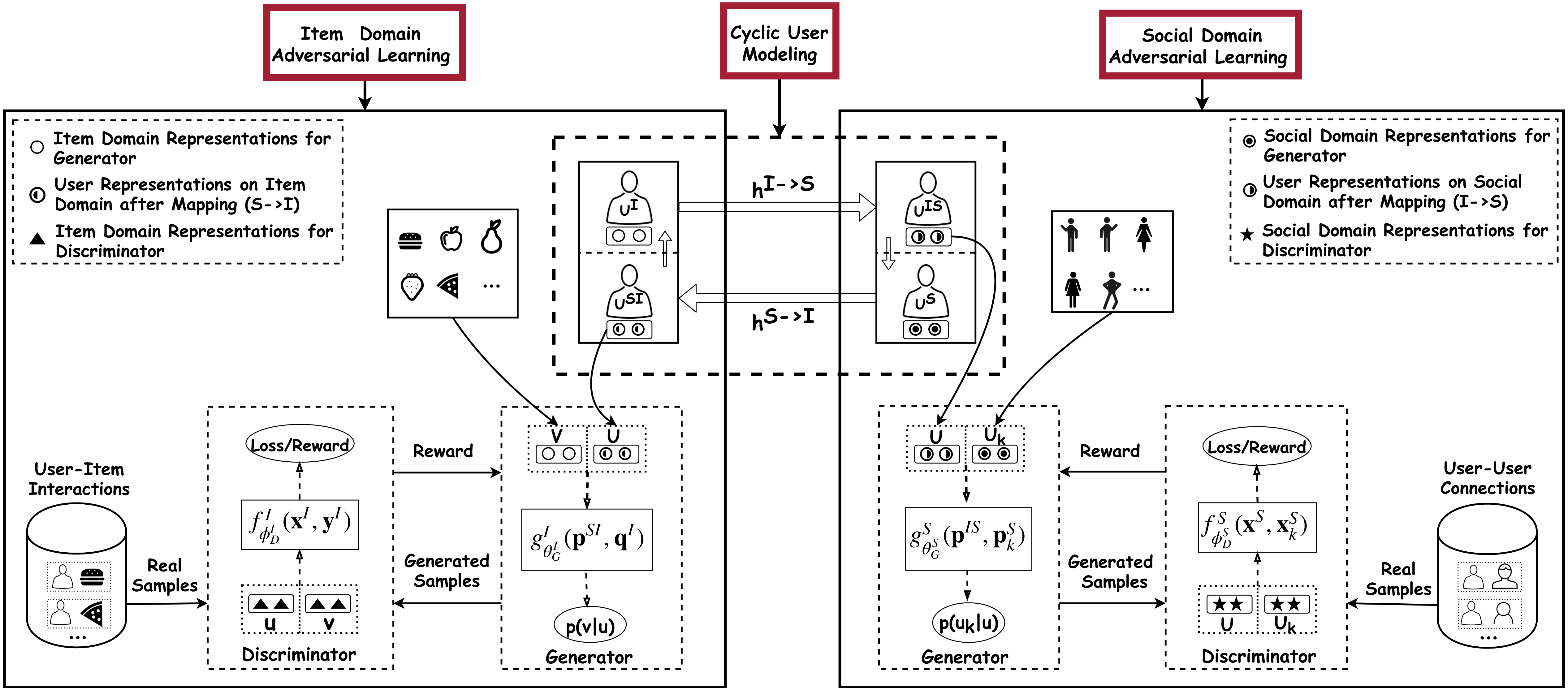


Dynamically generate “difficult” negative samples

▶ Optimization with Adversarial Learning (GAN)

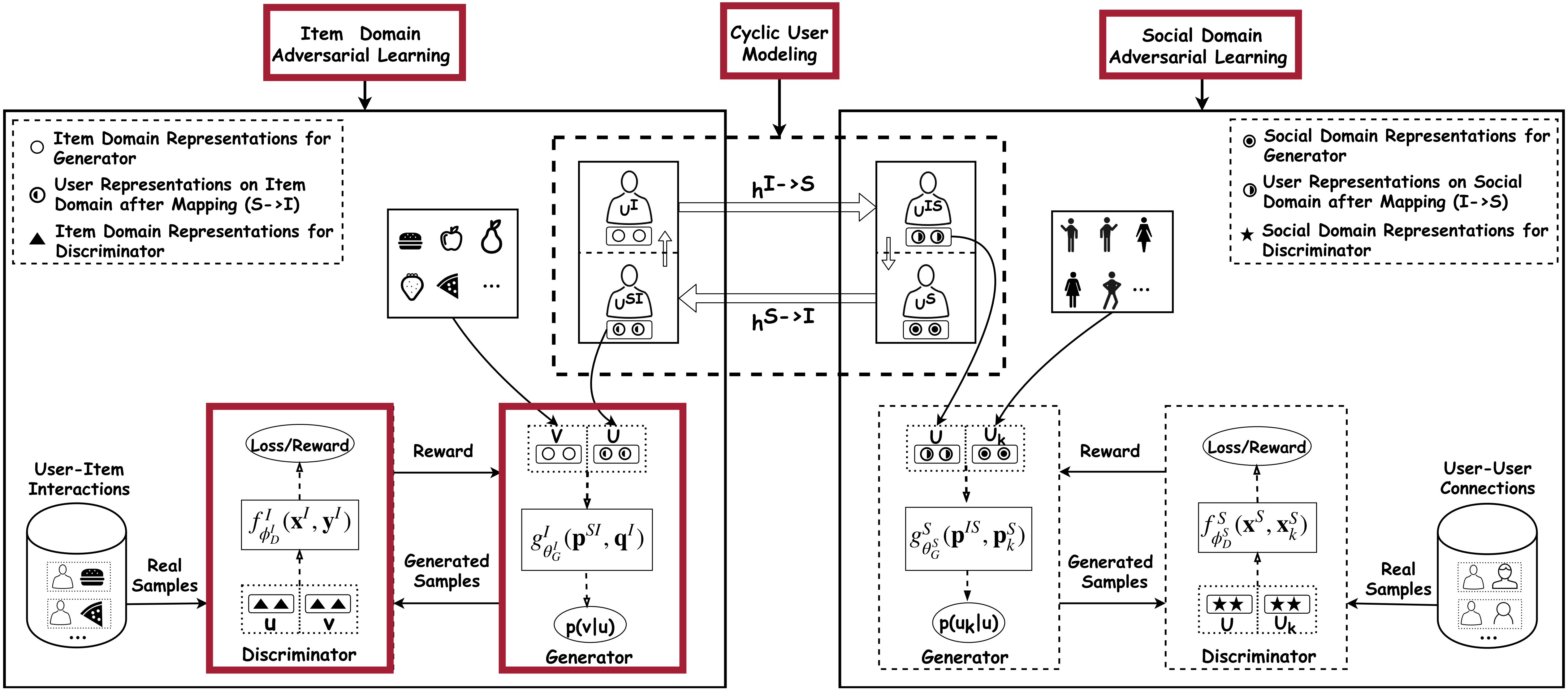
Deep Adversarial Social Recommendation, IJCAI, 2019

DASO



Deep Adversarial Social Recommendation, IJCAI, 2019

DASO



Deep Adversarial Social Recommendation, IJCAI, 2019

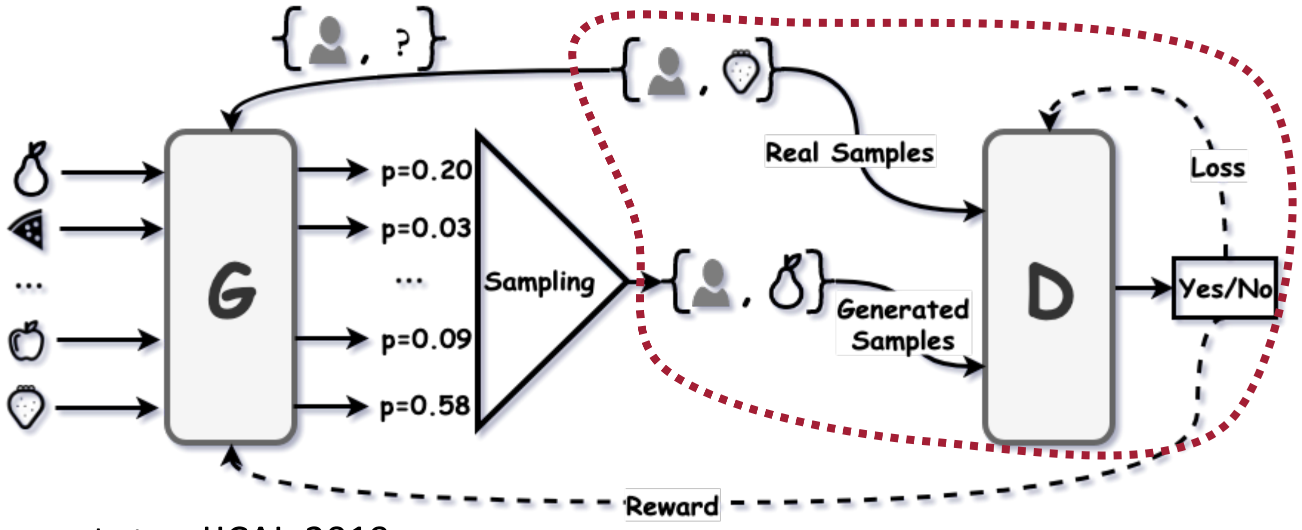
Item Domain Discriminator Model

❑ Discriminator

Goal: distinguish real user-item pairs (i.e., real samples) and the generated “fake” samples (relevant)

$$D^I(u_i, v_j; \phi_D^I) = \sigma(f_{\phi_D^I}^I(\mathbf{x}_i^I, \mathbf{y}_j^I)) = \frac{1}{1 + exp(-f_{\phi_D^I}^I(\mathbf{x}_i^I, \mathbf{y}_j^I))}$$ **(Sigmoid)**

Score function: $f_{\phi_D^I}^I(\mathbf{x}_i^I, \mathbf{y}_j^I) = (\mathbf{x}_i^I)^T \mathbf{y}_j^I + a_j,$



Deep Adversarial Social Recommendation, IJCAI, 2019

Item Domain Generator Model

Generator Model

Goal:

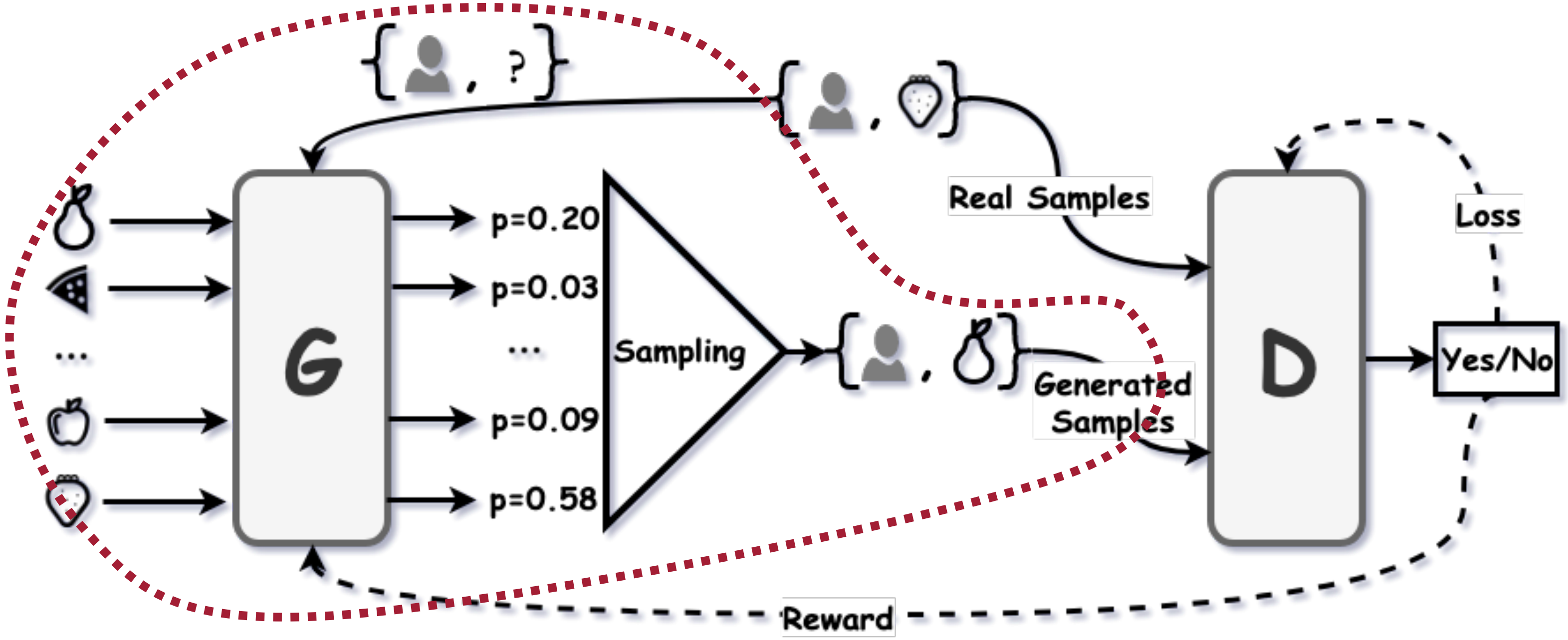
1. Approximate the underlying real conditional distribution **$p^{I}_{real}(v|u_i)$**
2. Generate (select/sample) the most relevant items for any given user u_i.

$$G^I(v_j|u_i;\theta^I_G) = \frac{exp(g^I_{\theta^I_G}(\mathbf{p}^{SI}_i, \mathbf{q}^I_j))}{\sum_{v_j \in \mathcal{V}} exp(g^I_{\theta^I_G}(\mathbf{p}^{SI}_i, \mathbf{q}^I_j))}$$

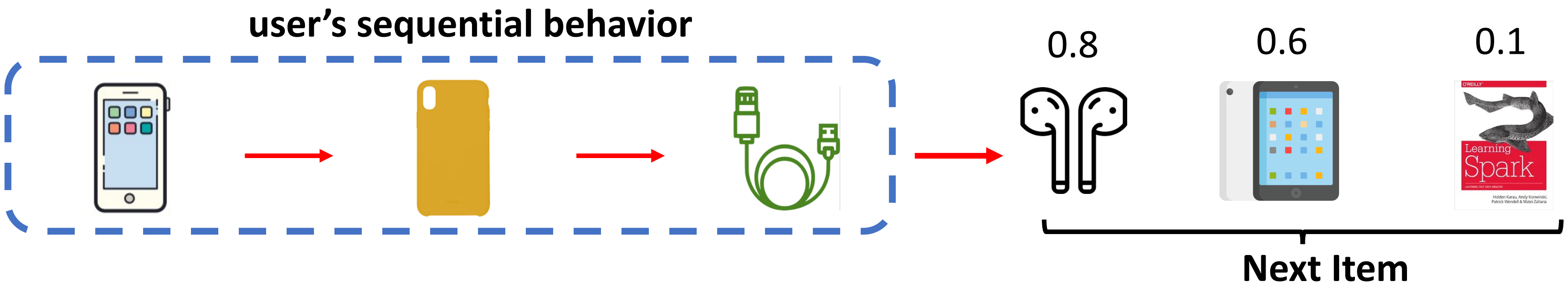
$\mathbf{p}^{SI}_i$ the transferred user representation from social domain

$$g^I_{\theta^I_G}(\mathbf{p}^{SI}_i, \mathbf{q}^I_j) = (\mathbf{p}^{SI}_i)^T \mathbf{q}^I_j + b_j$$

Optimization with Policy Gradient

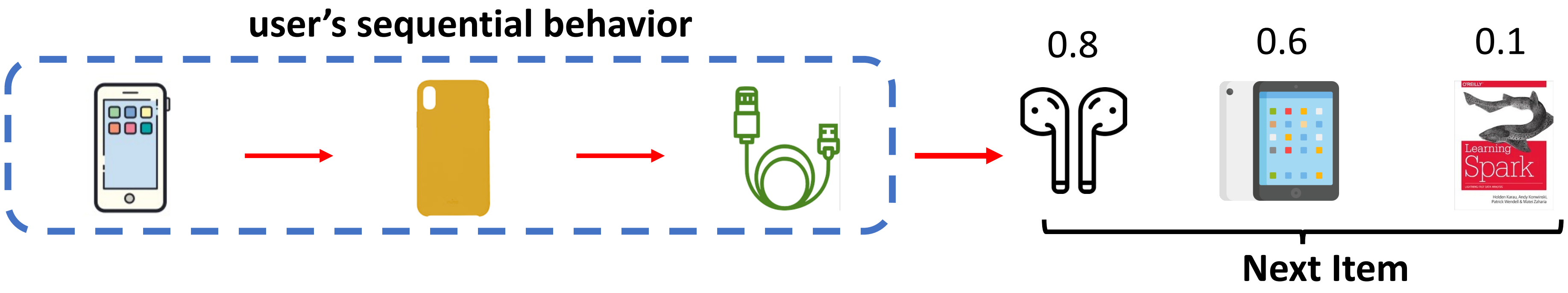


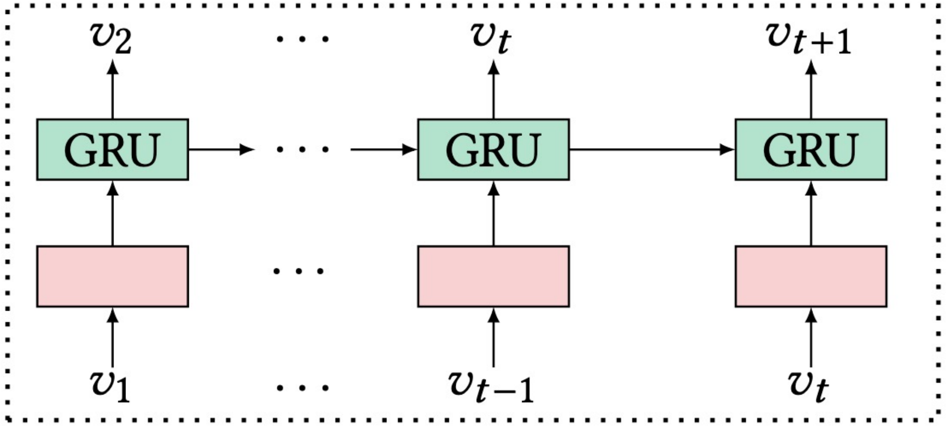
Sequential (Session-based) Recommendation



Session-based Recommendations with Recurrent Neural Networks, ICLR, 2016.
BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer, CIKM, 2019.

Sequential (Session-based) Recommendation

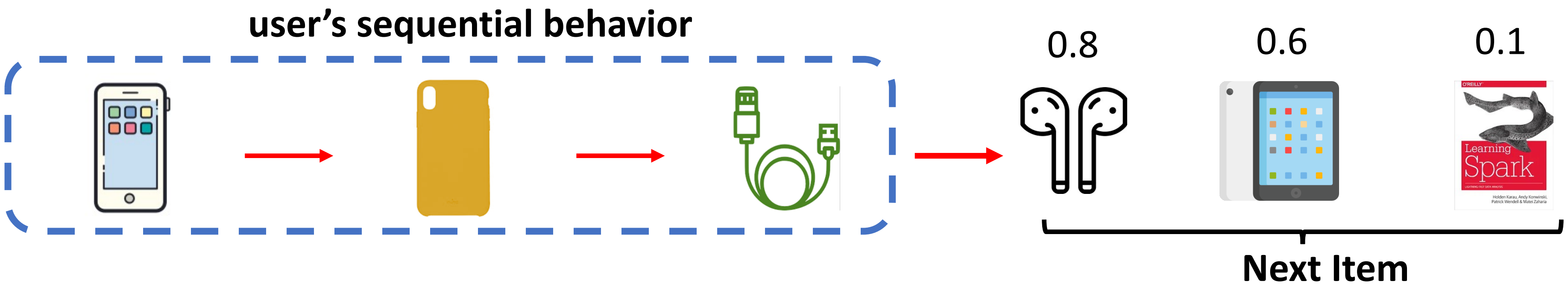


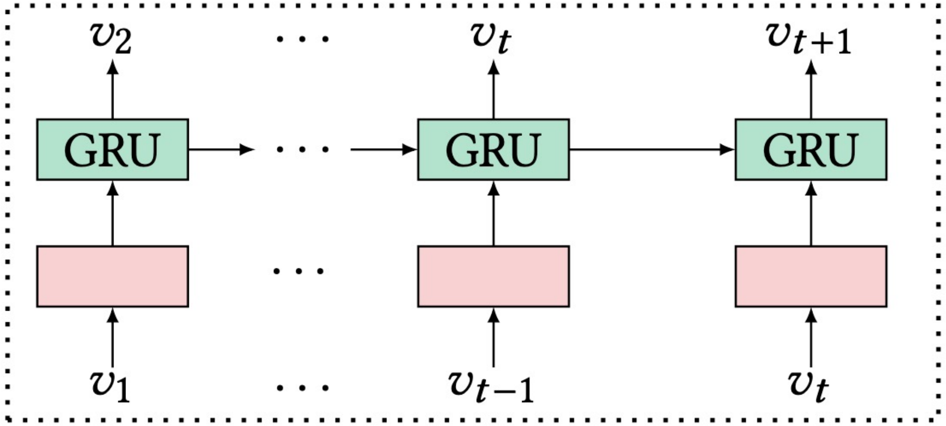


GRU based sequential recommendation method (**GRU4Rec**)

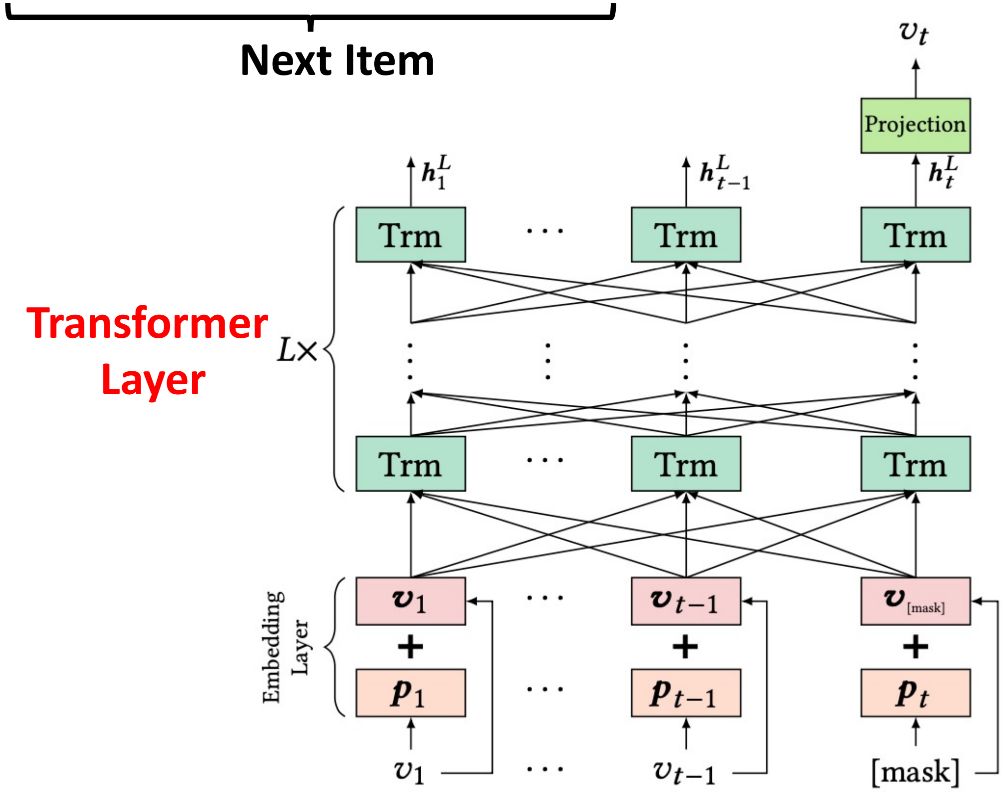
Session-based Recommendations with Recurrent Neural Networks, ICLR, 2016.
BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer, CIKM, 2019.

Sequential (Session-based) Recommendation





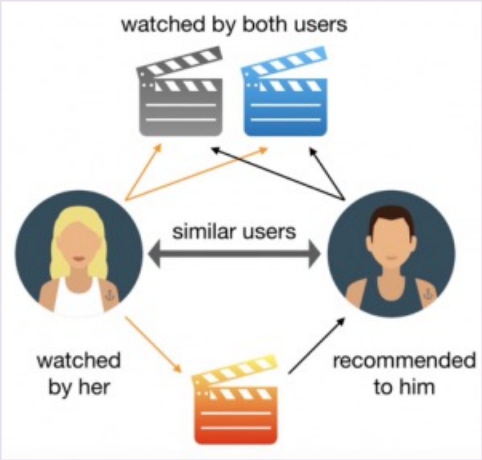
GRU based sequential recommendation method (**GRU4Rec**)



BERT4Rec

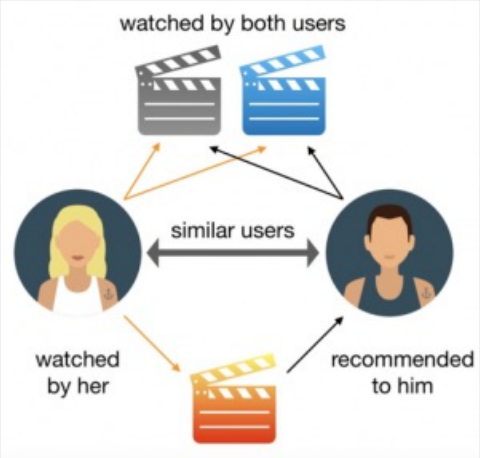
Session-based Recommendations with Recurrent Neural Networks, ICLR, 2016.
BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer, CIKM, 2019.

Shortcomings of Existing Deep Recommender Systems

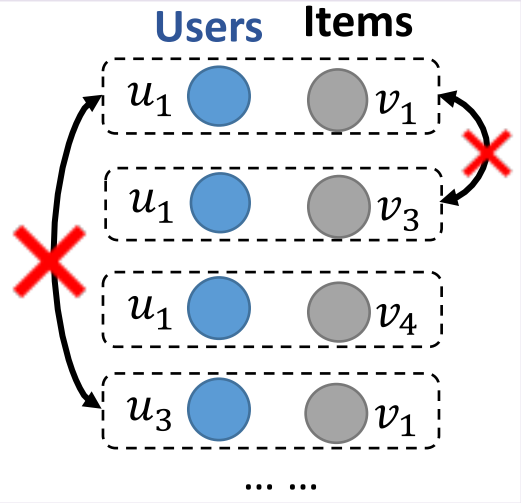


Recommendation Policies
- **Offline optimization**
- **Short-term reward**

Shortcomings of Existing Deep Recommender Systems

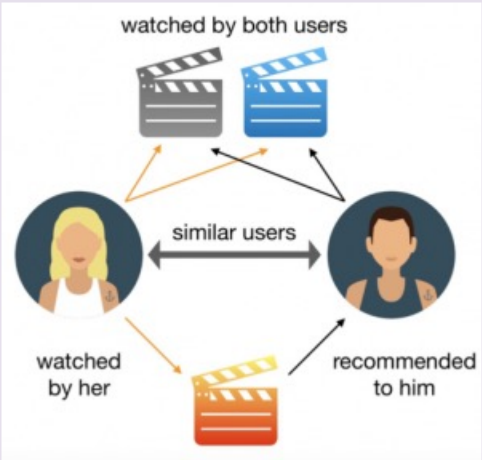


Recommendation Policies
- **Offline optimization**
- **Short-term reward**

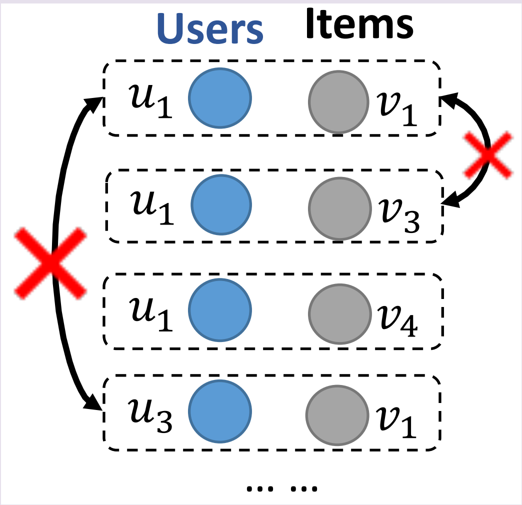


Graph-structured Data
- **Information isolated island Issue:** ignore implicit/explicit relationships among instances

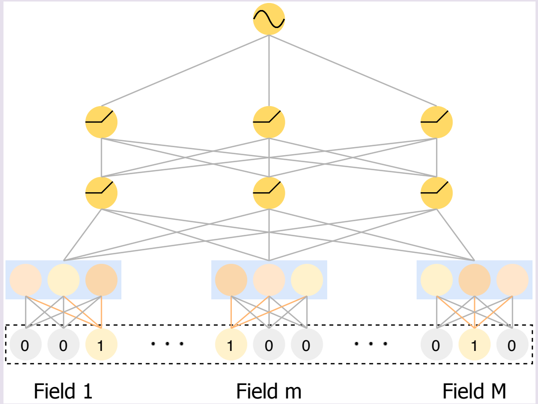
Shortcomings of Existing Deep Recommender Systems



Recommendation Policies
- Offline optimization
- Short-term reward

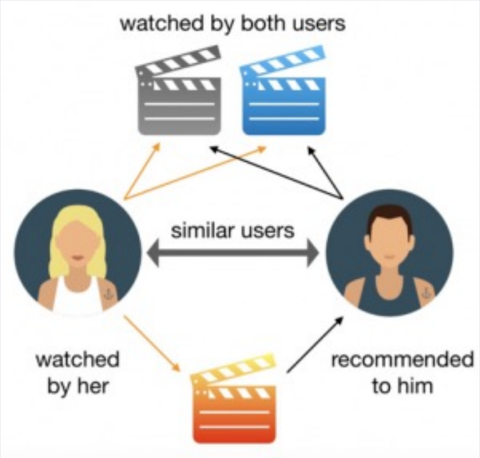


Graph-structured Data
- Information isolated island Issue: ignore implicit/explicit relationships among instances

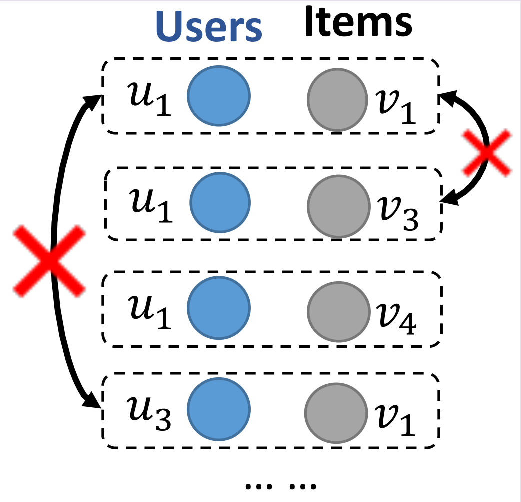


Manually Deisgned Architectures
- Expert knowledge
- Time and engineering efforts

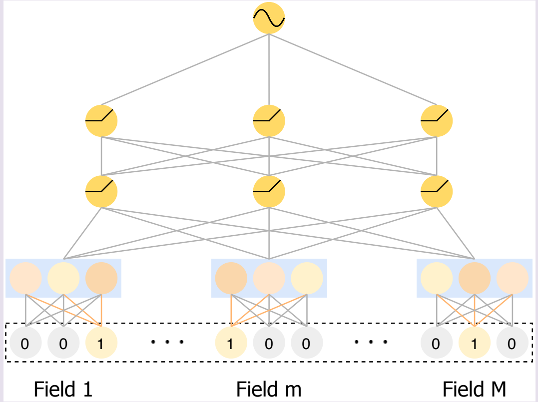
Shortcomings of Existing Deep Recommender Systems



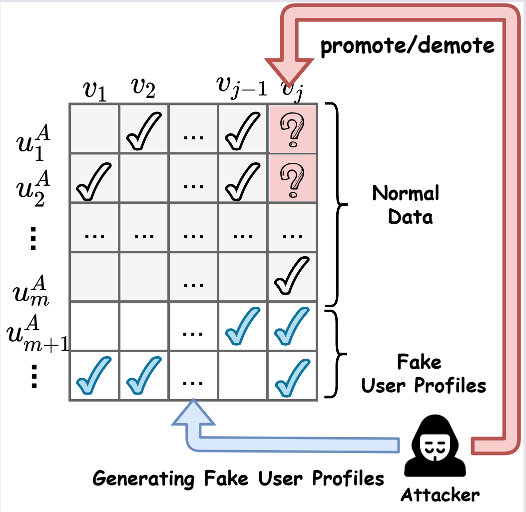
Recommendation Policies
- Offline optimization
- Short-term reward



Graph-structured Data
- Information isolated island Issue: ignore implicit/explicit relationships among instances



Manually Deisgned Architectures
- Expert knowledge
- Time and engineering efforts



Poisoning attacks:
- Promote/demote items
- White/grey/black-box attacks