AI007 《人工智能》Week08

推荐系统

Recommender Systems

许岩岩

上海交通大学 · 人工智能研究院

2021年11月4日

Major Ref: https://deeprs-tutorial.github.io/

提纲

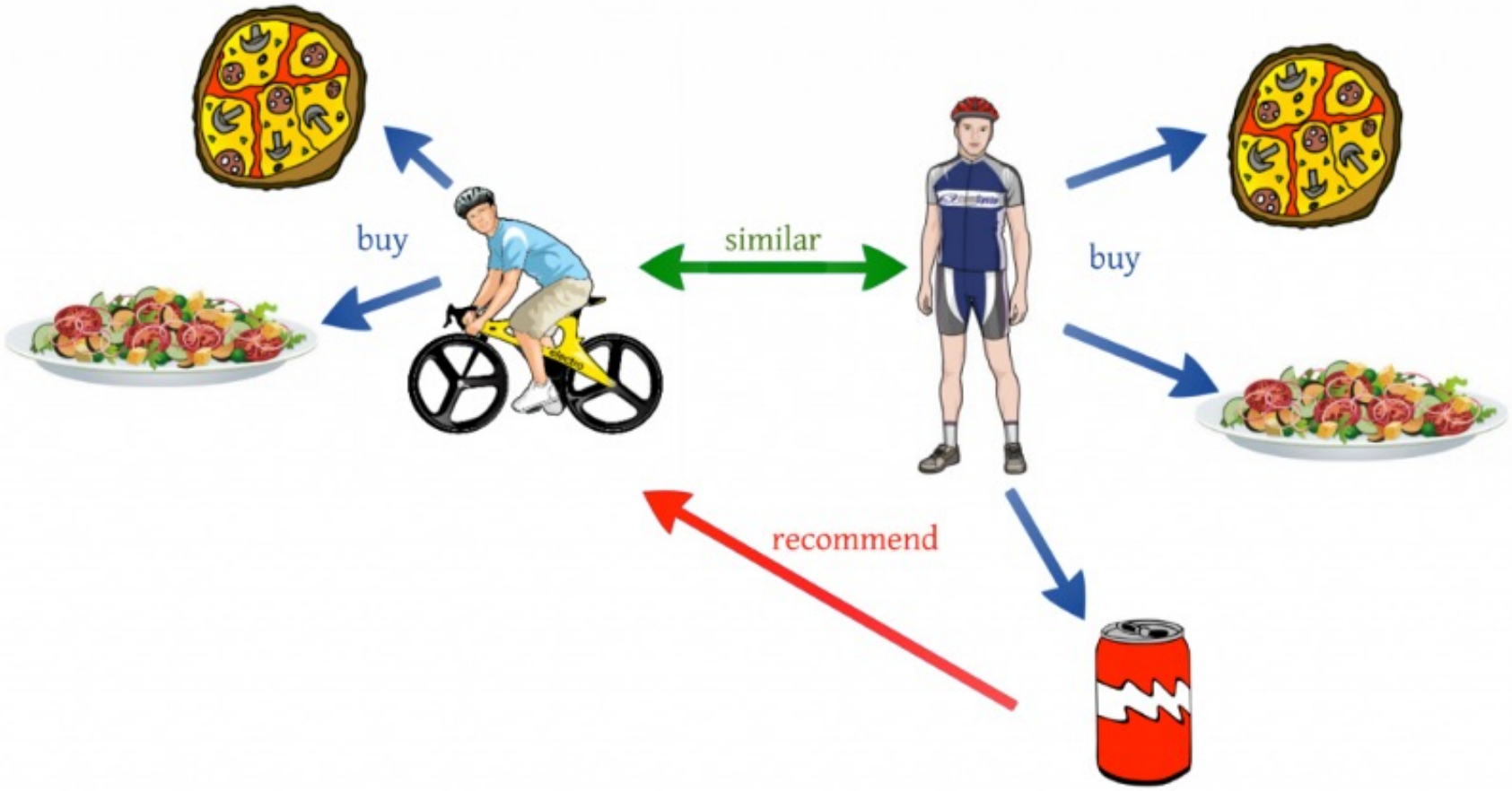
1 推荐系统概述

2 DNN推荐算法

3 RL推荐算法

4 GNN推荐算法

推荐系统（Recommender Systems）的任务就是联系用户和信息，一方面帮助用户发现对自己有价值的信息，另一方面让信息能够展现在对它感兴趣的用户面前，从而帮忙用户从海量的信息中发掘自己潜在的需求。



推荐系统（Recommender Systems）广泛应用于各大**在线销售平台**、内容分享平台、社交网络

amazon

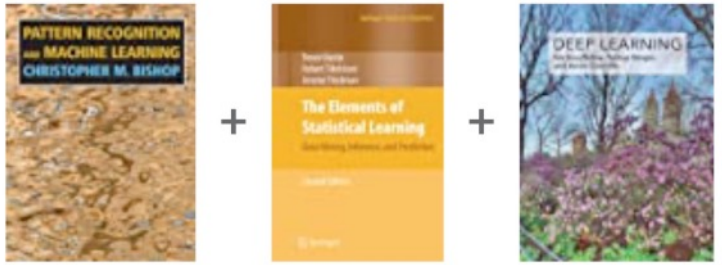
ebay

淘宝网
Taobao.com

JD.COM

Product Recommendation

Frequently bought together



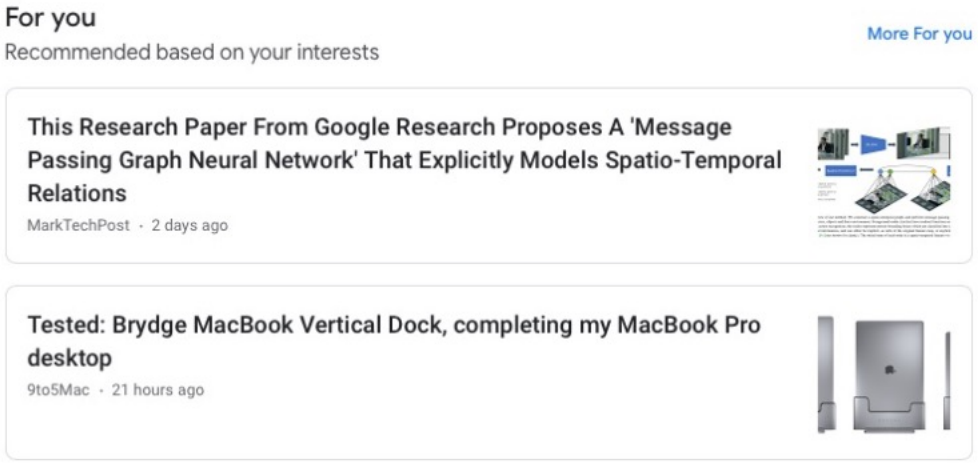
Total price: $208.9

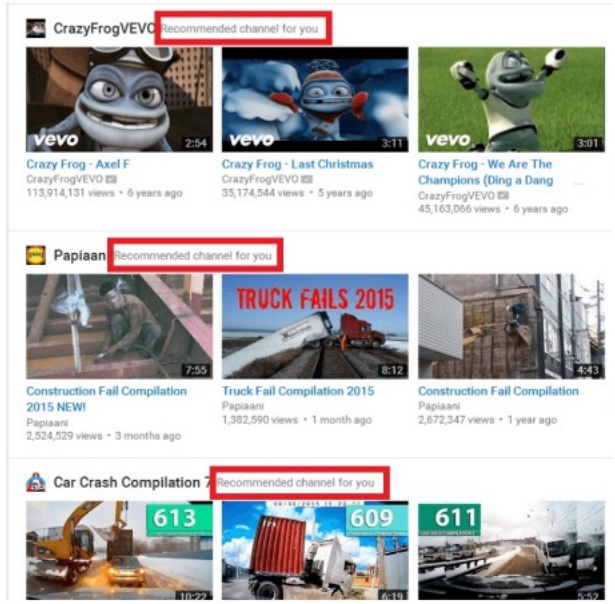
Add all three to Cart

Add all three to List

推荐系统（Recommender Systems）广泛应用于各大在线销售平台、**内容分享平台**、社交网络

News/Video/Image Recommendation





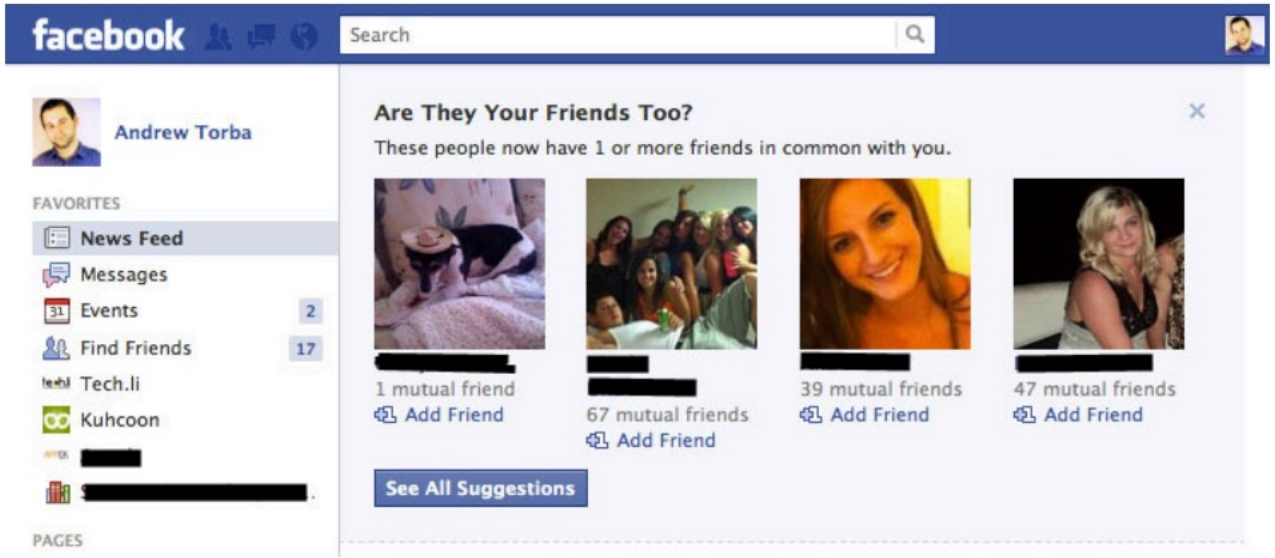
推荐系统（Recommender Systems）广泛应用于各大在线销售平台、内容分享平台、**社交网络**

facebook

新浪微博 weibo.com

LinkedIn

Friend Recommendation



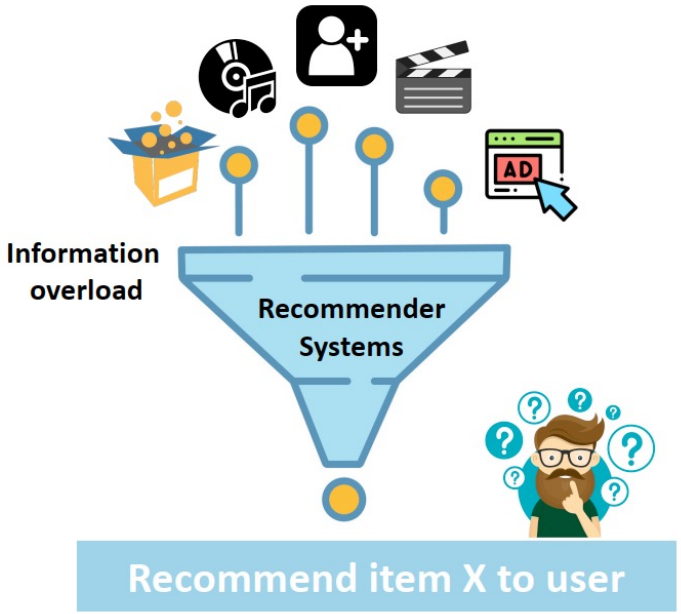
搜索引擎 vs 推荐引擎

搜索引擎

- 用户目的明确，对信息的需求可转化为精准的关键字
- 用户对于搜索引擎是主动的
- **马太效应**的问题，即会造成越流行的东西随着搜索过程的迭代会越流行，使得那些越不流行的东西石沉大海。

推荐引擎

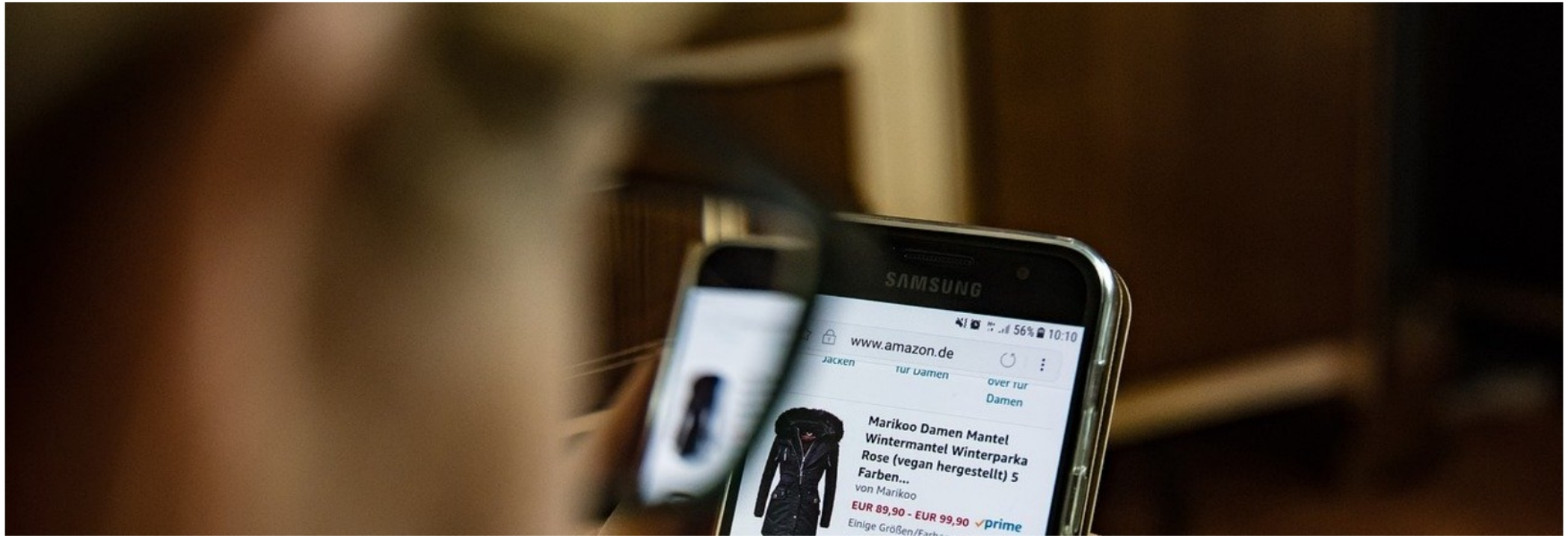
- 用户没有明确目的，或者目的模糊的
- 通过用户的历史行为或者用户的兴趣偏好或者用户的人口统计学特征来送给推荐算法
- 用户对于搜索引擎是被动的



Items can be: Products, News, Movies, Videos, Friends, etc.

Amazon's recommendation algorithm drives 35% of its sales

July 3, 2020



Netflix says 80 percent of watched content is based on algorithmic recommendations

It's not just about subscribing to Netflix, it's about trusting it too

By Sameer Chhabra @SameerChhabra94 AUG 22, 2017 | 8:00 AM EDT | 3 COMMENTS

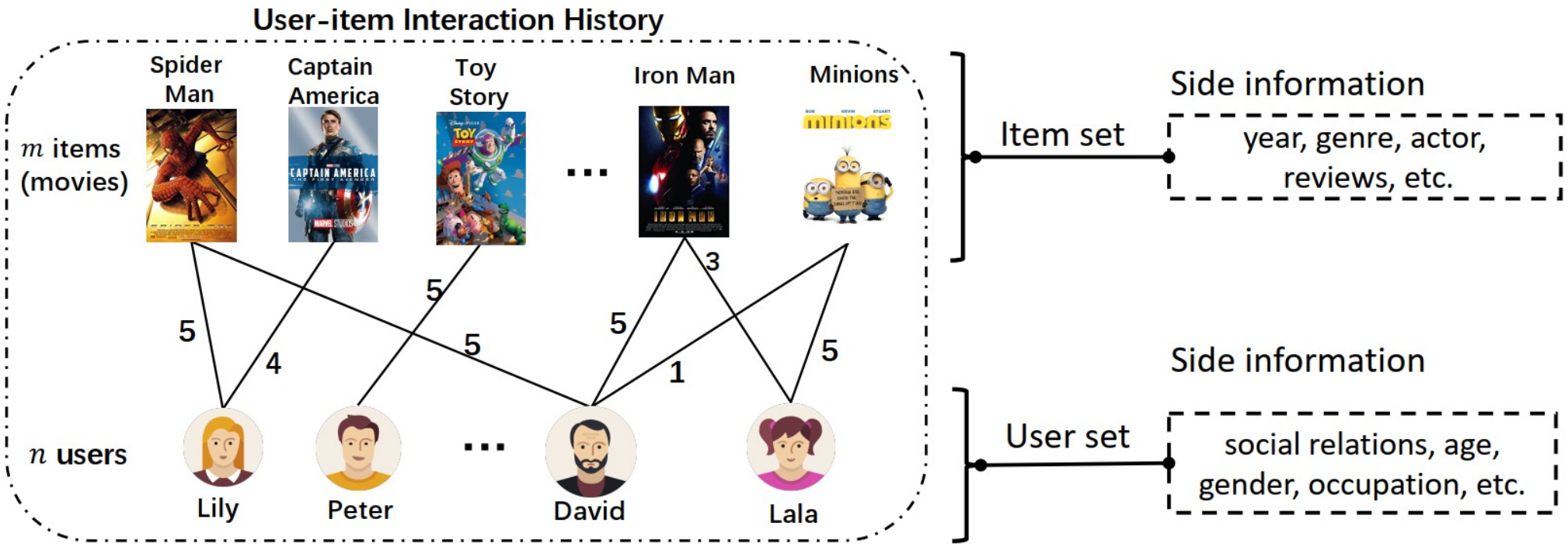
推荐系统 – 问题定义

INPUT

用户-项目之间的历史交互信息及额外的辅助信息（如社交关系，商品属性）

OUTPUT

预测某用户与特定项目交互的可能性（如点击、浏览、购买）



评分预测

此方法一般通过学习用户对物品的历史评分，预测用户可能会为他没有进行评分的物品打多少分，通常用于在线视频、音乐等服务的推荐。评分预测的效果评估一般通过均方根误差（RMSE）和平均绝对误差（MAE）计算。对于测试集 T 中的一个用户 u 和物品 i，令 r_{ui} 是用户u对物品i的实际评分，而 $\hat{r}_{ui}$ 是推荐系统给出的预测评分，则 RMSE 和 MAE 的定义为：

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{u,i \in T}(r_{ui} - \hat{r}_{ui})^2}{|T|}}$$

$$\mathrm{MAE} = \frac{\sum_{u,i \in T} |r_{ui} - \hat{r}_{ui}|}{|T|}$$

TopN 推荐

此方法一般不考虑评分，而是为用户提供一个个性化推荐列表，通过预测用户对物品的兴趣度对列表进行排序，选取其中前 N 个物品推荐给用户，通常用于电子商务、社交网络、互联网广告推荐。TopN 推荐一般通过准确率（precision）、召回率（recall）和 F1 值（平衡分数）度量。令 R(u) 是为用户推荐的物品列表，T(u) 是用户在测试集上的行为列表。

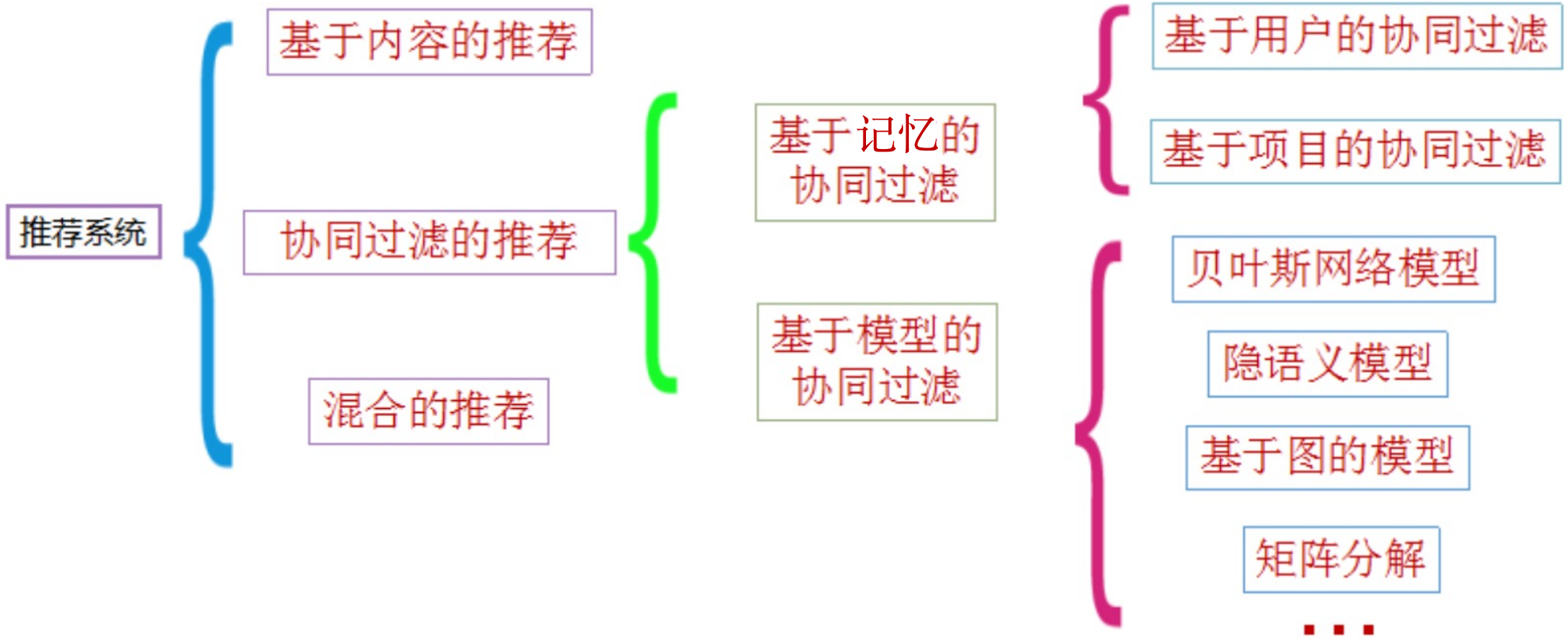
$$\text{Recall} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|}$$

$$\text{Precision} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|}$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

推荐系统 – 算法分类

推荐系统基础

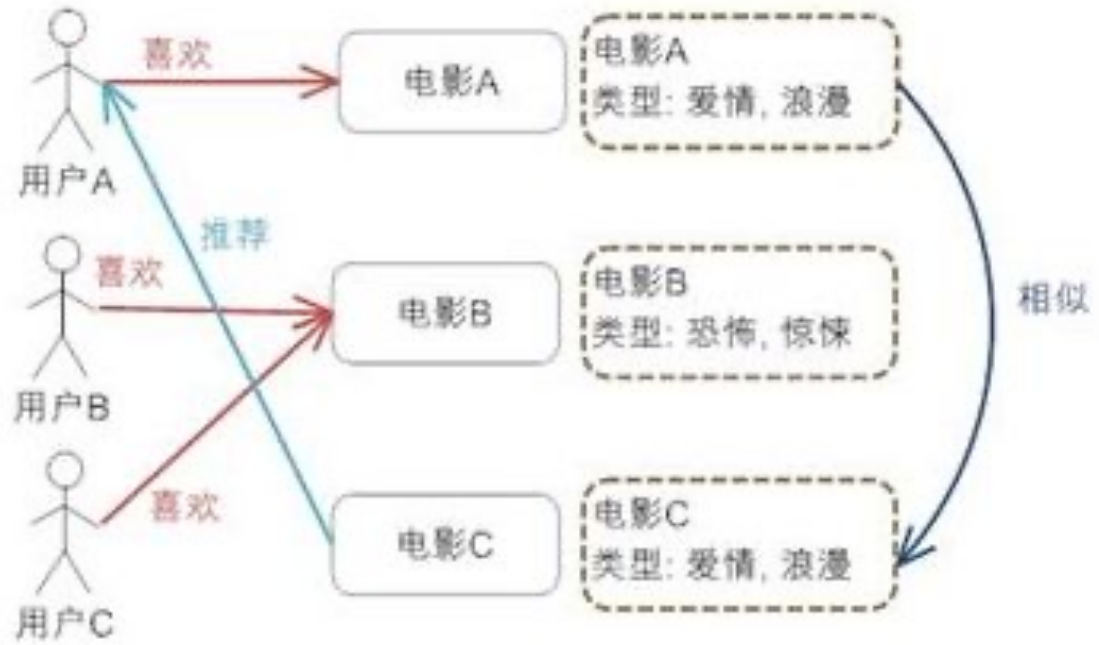


Ref: https://zhuanlan.zhihu.com/p/27502172

基于内容的推荐

基于内容的推荐（Content-based Recommendations, CB）的过程一般包括以下三步：

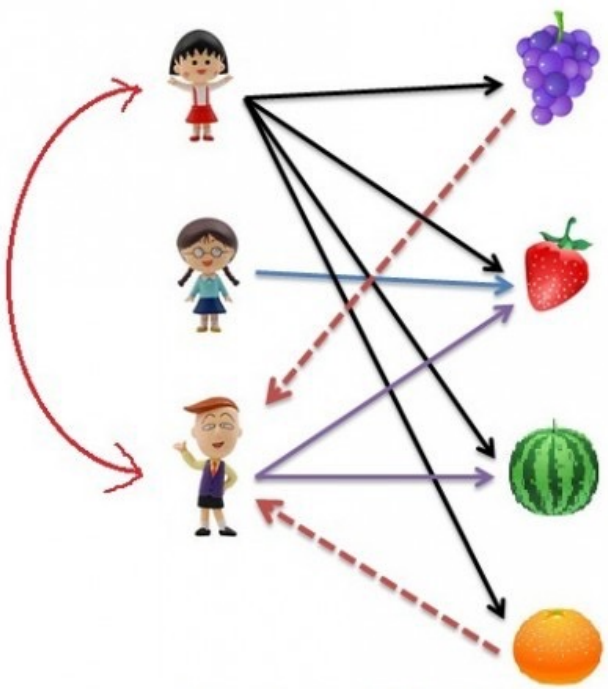
1. **Item Representation**：为每个item抽取出一些特征（也就是item的content了）来表示此item；
2. **Profile Learning**：利用一个用户过去喜欢（及不喜欢）的item的特征数据，来学习出此用户的喜好特征（profile）；
3. **Recommendation Generation**：通过比较上一步得到的用户profile与候选item的特征，为此用户推荐一组相关性最大的item。

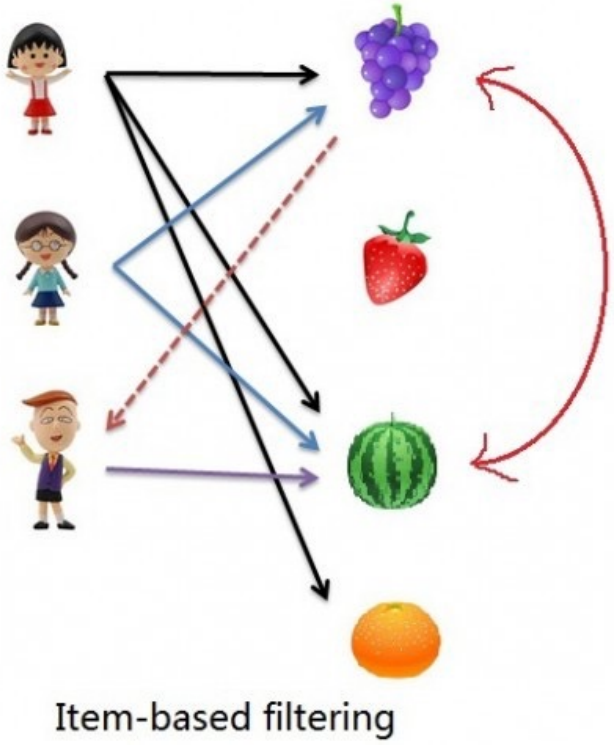


协同过滤（Collaborative filtering ，CF）

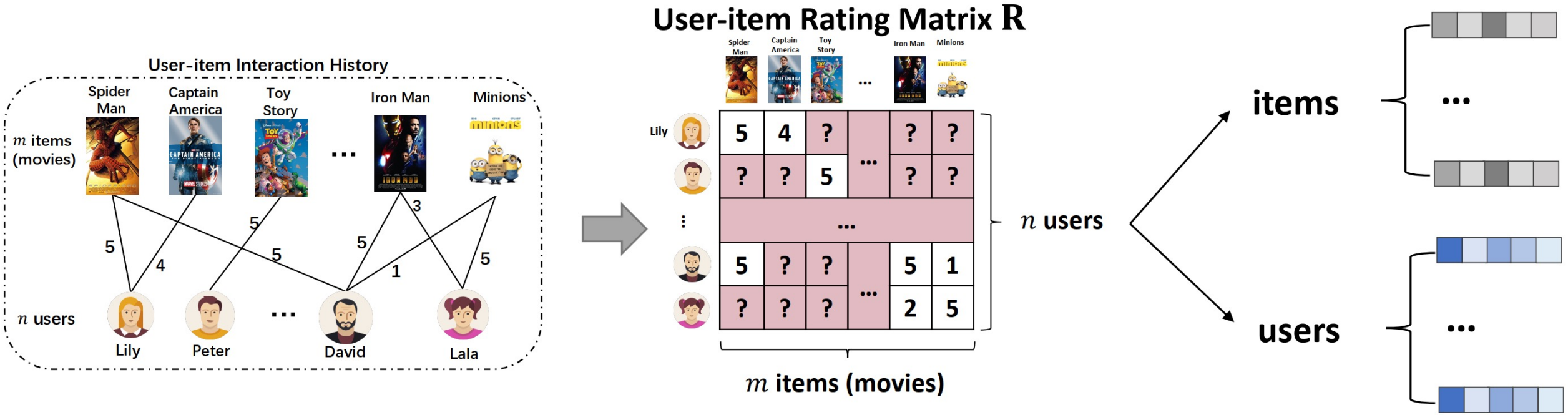
通过集体智慧的力量来进行工作，过滤掉那些用户不感兴趣的项目。

假设：为特定用户找到他真正感兴趣的内容的好方法是首先找到与此用户有相似兴趣的其他用户，然后将他们感兴趣的内容推荐给此用户。





一般采用最近邻技术，利用用户的历史喜好信息计算用户之间的距离，然后利用目标用户的最近邻居用户对商品评价的加权评价值来预测目标用户对特定商品的喜好程度，系统从而根据这一喜好程度来对目标用户进行推荐，通常需要用到用户-项目（User-Item，UI）矩阵的信息。



Task: predicting missing movie ratings in Netflix.

基于记忆的协同过滤

- **基于记忆的协同过滤（Memory-Based CF）**
  - **基于用户(user-based)的协同过滤**：

    利用用户访问行为的相似性来互相推荐用户可能感兴趣的资源
  - **基于项目(item-based)的协同过滤**：

    利用项目被访问记录计算项目之间的相似度，为用户推荐类似资源
- **基于模型的协同过滤（Model-Based CF）**

  利用用户对项目的访问行为，构建用户与项目之间的关联规则，预测未发生关联的“用户-项目”之间的评分关系

基于用户的协同过滤

思想：给用户推荐和他兴趣相似的用户感兴趣的物品。当需要为一个用户A进行推荐时，首先，找到和A兴趣相似的用户集合U，然后，把集合U中用户感兴趣而A没有听说过的物品推荐给A。

步骤：（1）计算用户之间的相似度，选取最相似的N个用户（2）根据相似度计算用户评分。

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

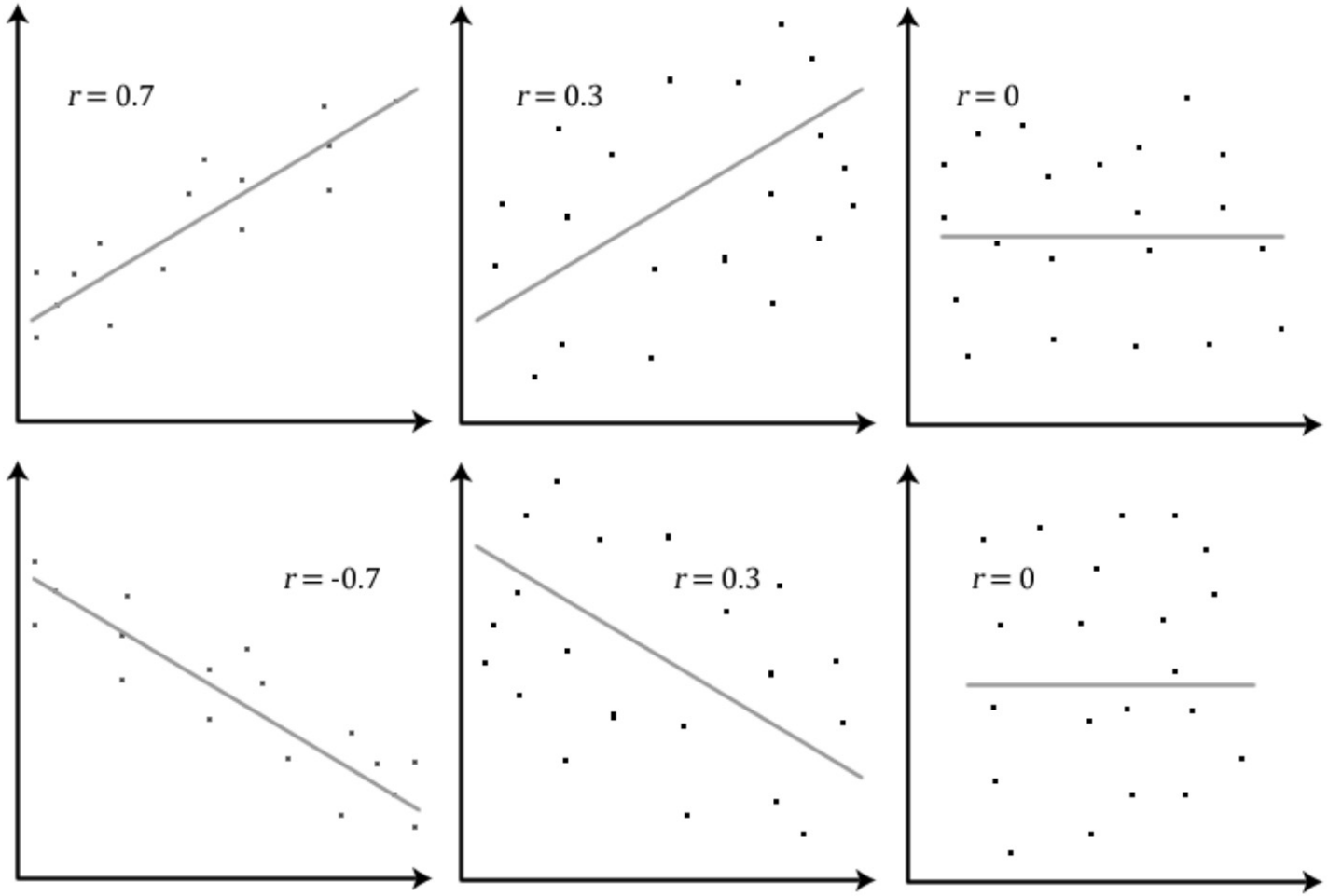
sim = 0,85

sim = 0,00

sim = 0,70

sim = -0,79

用户相似度评价：皮尔森相关系数、Spearman 相关系数、余弦相似度等



基于项目的协同过滤

思想：给用户推荐和他们以前喜欢的物品相似的物品，这里所说的相似并非从物品的内容角度出发，而是基于一种假设：喜欢物品A的用户大多也喜欢物品B代表着物品A和物品B相似。

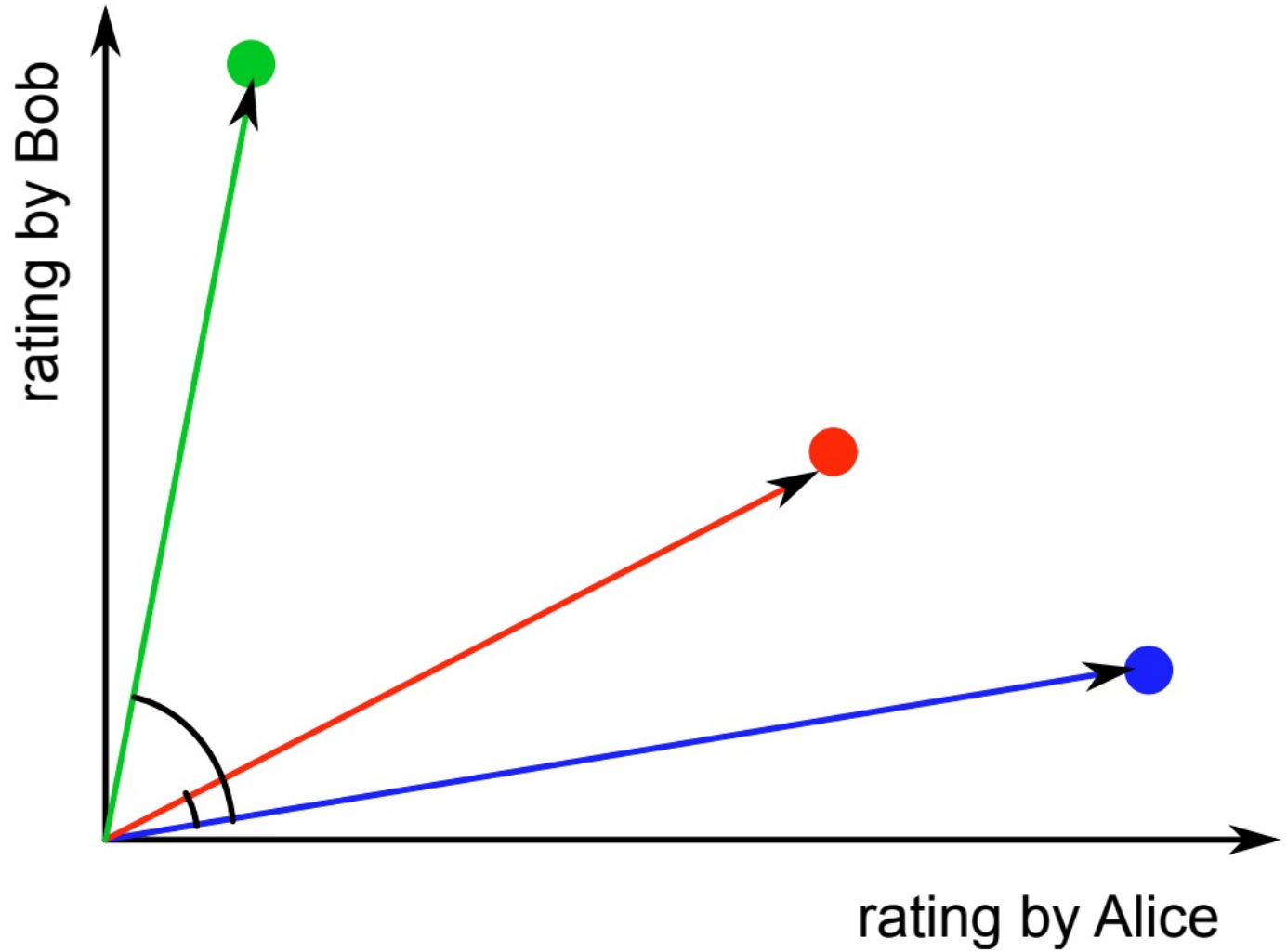
步骤：（1）计算物品相似度，选出最相似的N个物品；（2）根据相似度计算用户评分。

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

基于项目的协同过滤

项目相似度评价：余弦相似度

$$\cos(\alpha) = \frac{A \cdot B}{\|A\| \|B\|}$$



奇异值分解（Singular Value Decomposition, SVD）

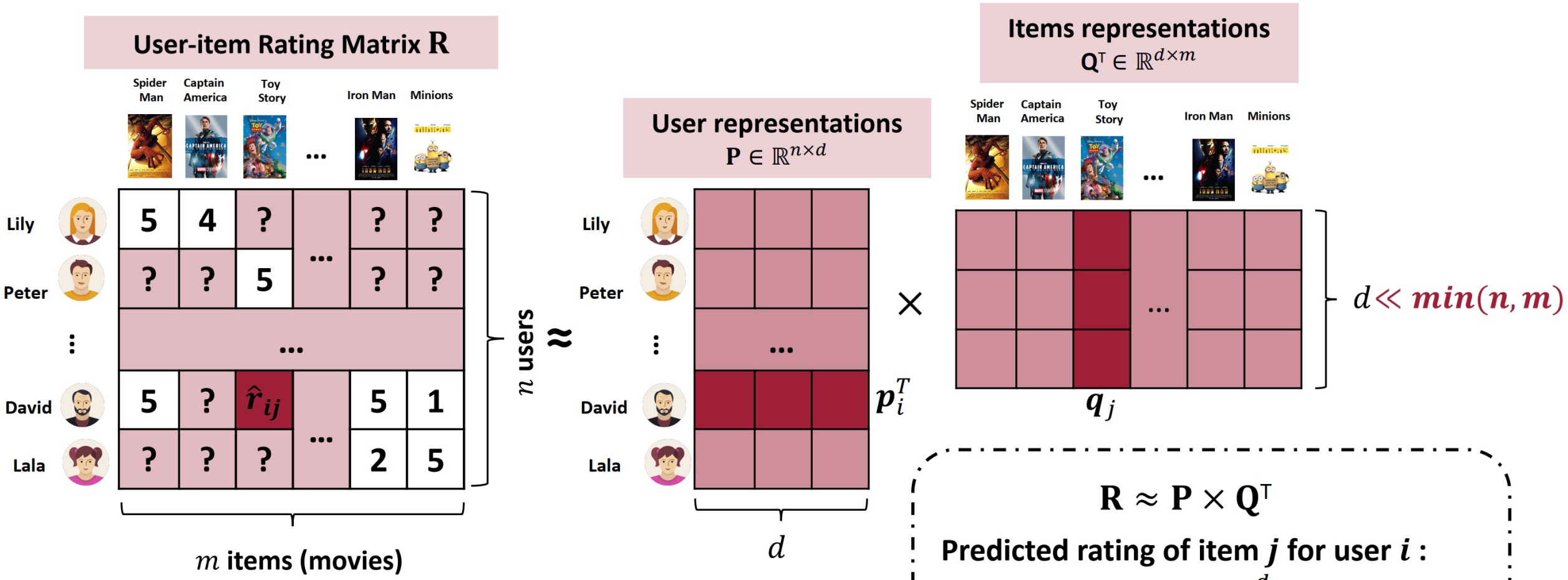
$$\underset{m\times n}{X} = \underset{m\times r}{U}\,\underset{r\times r}{S}\,\underset{r\times n}{V^{\mathsf{T}}}$$

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix} = \begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix} \begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix} \begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}$$

Note: 要求矩阵是稠密的，即矩阵里的元素要非空，否则就不能运用SVD分解。

对于缺失值，先用均值或者其他统计学方法来填充矩阵

矩阵分解（Matrix Factorization）



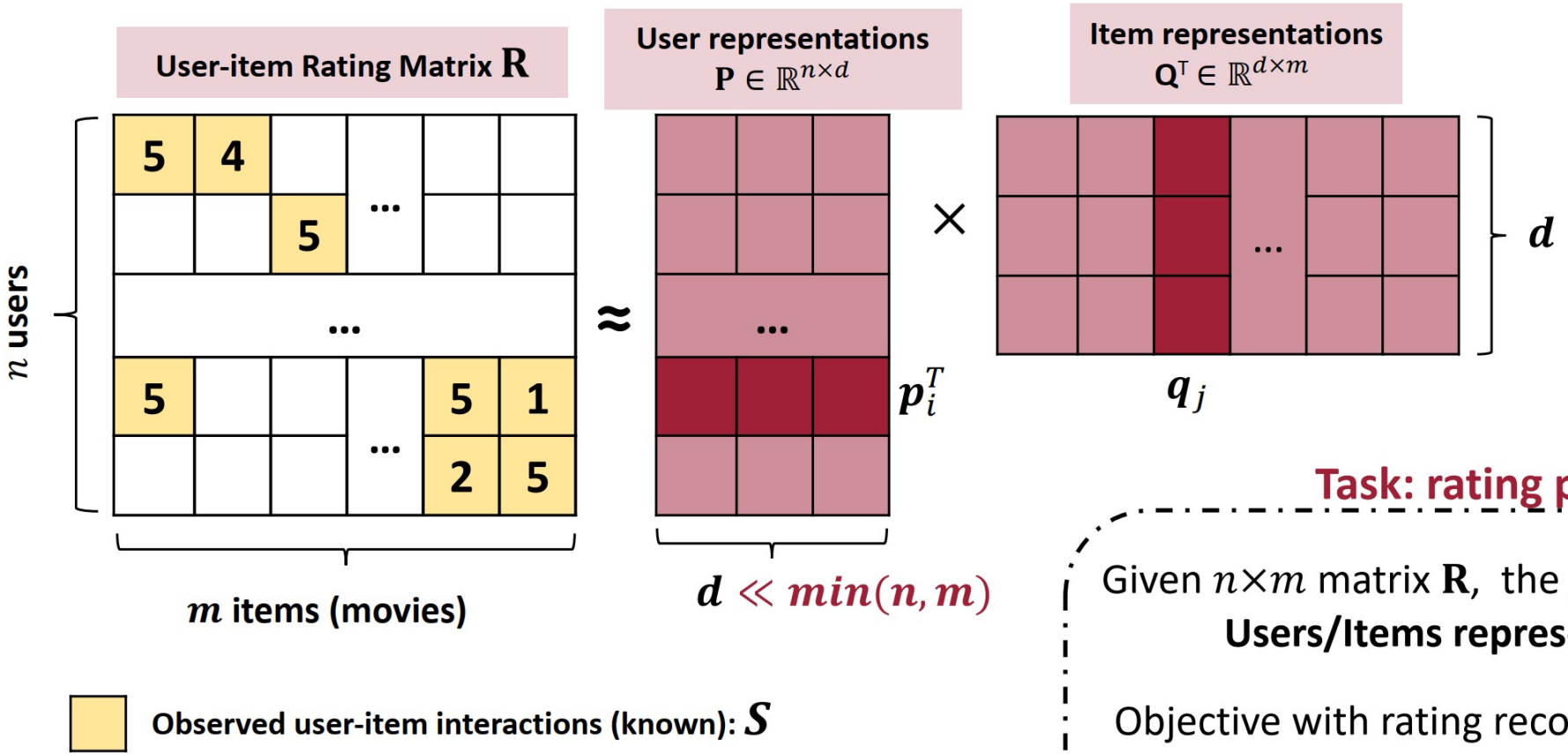
$$R \approx P \times Q^T$$

Predicted rating of item j for user i :

$$\hat{r}_{ij} \approx p_i^T q_j = \sum_{k=1}^{d} p_{ik} q_{jk}$$

FunkSVD：将用户项目矩阵分解为2个低秩的用户、项目矩阵

矩阵分解（Matrix Factorization）



Task: rating prediction in Netflix

Given $n \times m$ matrix $\mathbf{R}$, the goal is to learn:

Users/Items representations: $\mathbf{P} \in \mathbb{R}^{n \times d}$, $\mathbf{Q} \in \mathbb{R}^{m \times d}$

Objective with rating reconstruction error:

$$min_{\mathbf{P},\mathbf{Q}} \sum_{i,j \in S} (r_{ij} - \hat{r}_{ij})^2 = \sum_{i,j \in S} (r_{ij} - \boldsymbol{p}_i^T \boldsymbol{q}_j)^2$$

observed rating score

predicted rating score

FunkSVD：将用户项目矩阵分解为2个低秩的用户、项目矩阵

矩阵分解（Matrix Factorization）

- 带正则项的FunkSVD：避免对观测数据过拟合
- 概率矩阵分解PMF： FunkSVD的概率解释版本
- BiasSVD：增加评分偏置
- SVD++：评分中增加隐藏信息，如收藏
- timeSVD： 评分建模为时间的函数
- NMF：带有非负约束的矩阵分解
- ConvMF：将CNN与MF相结合，捕捉文档推荐中的上下文信息

张量分解（Tensor Factorization）

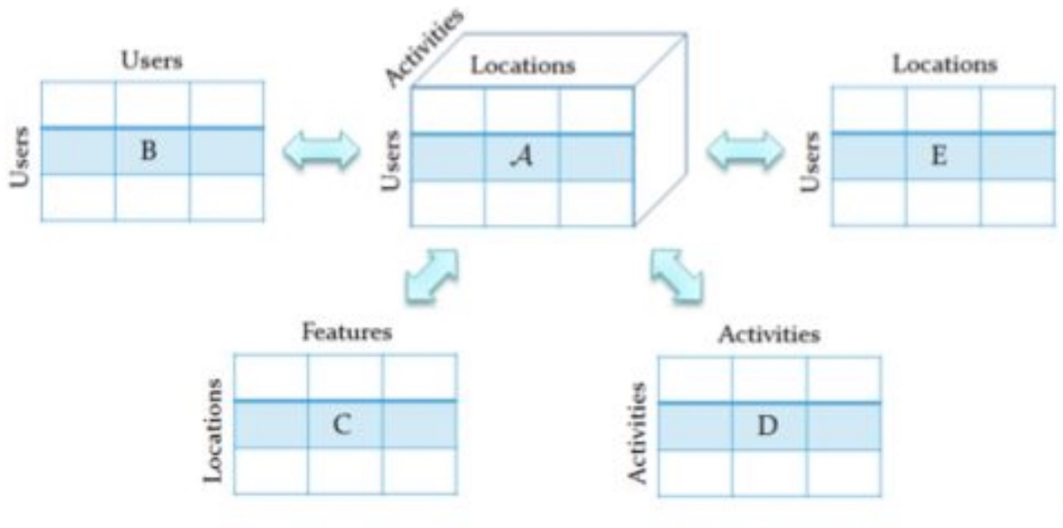
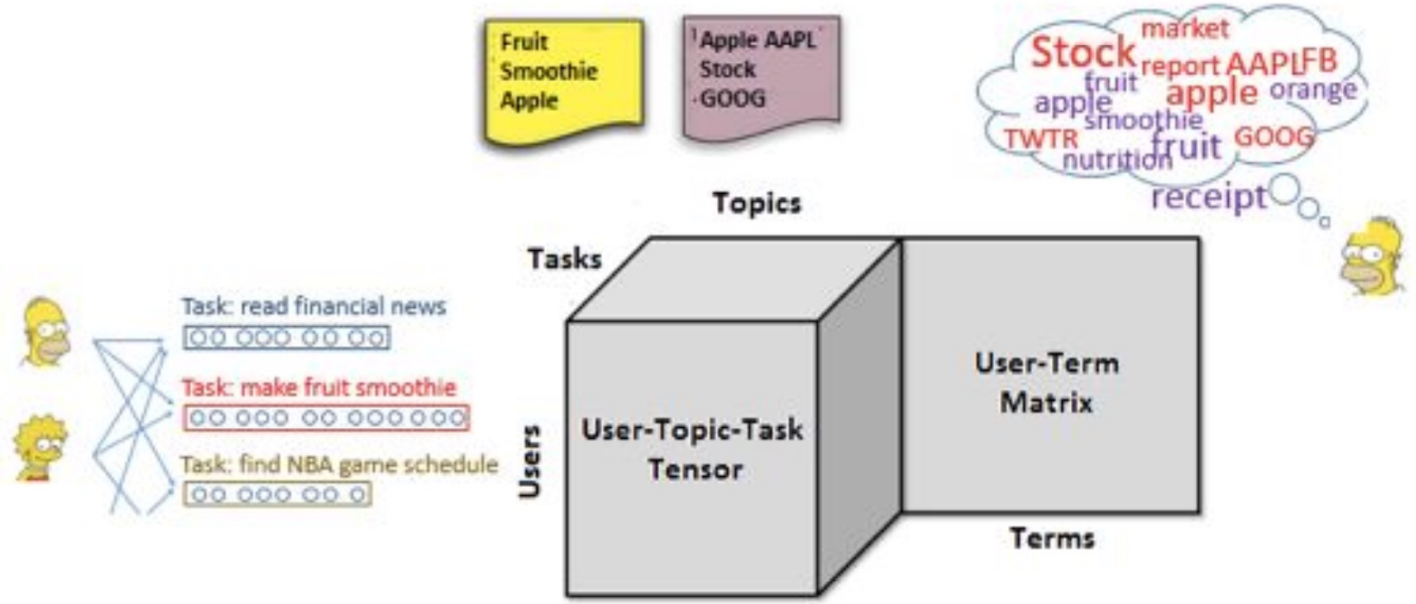


Figure 2: Model illustration in a tensor/matrix form.



AAAI 2010: Collaborative Filtering Meets Mobile Recommendation: A User-Centered Approach

提纲

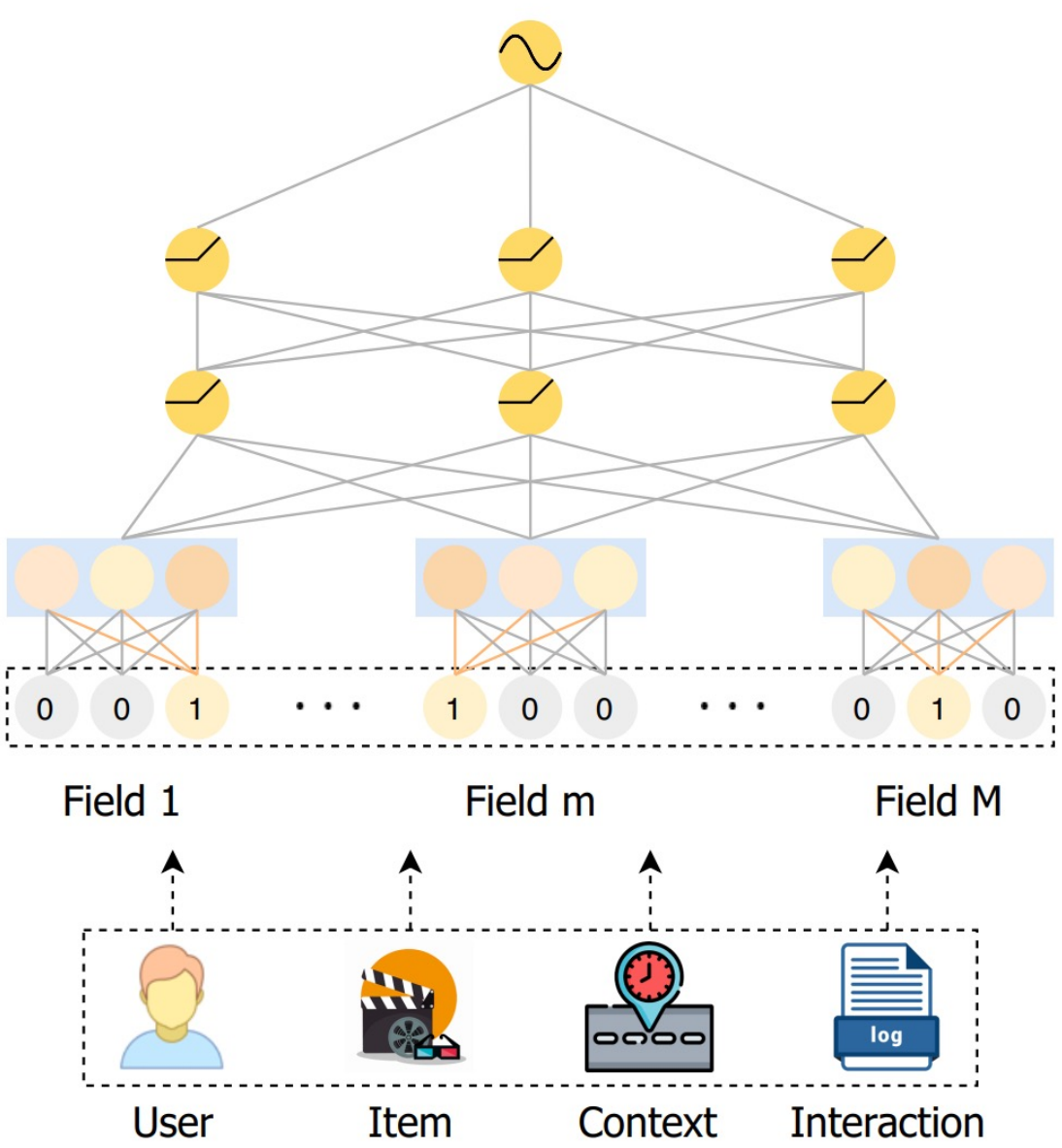
1 推荐系统概述

2 DNN推荐算法

3 RL推荐算法

4 GNN推荐算法

基于深度学习的推荐系统



Prediction layer

Hidden layer
(e.g., MLP, CNN, RNN, etc.)

Embedding layer

通用深度学习架构

One-hot 编码

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

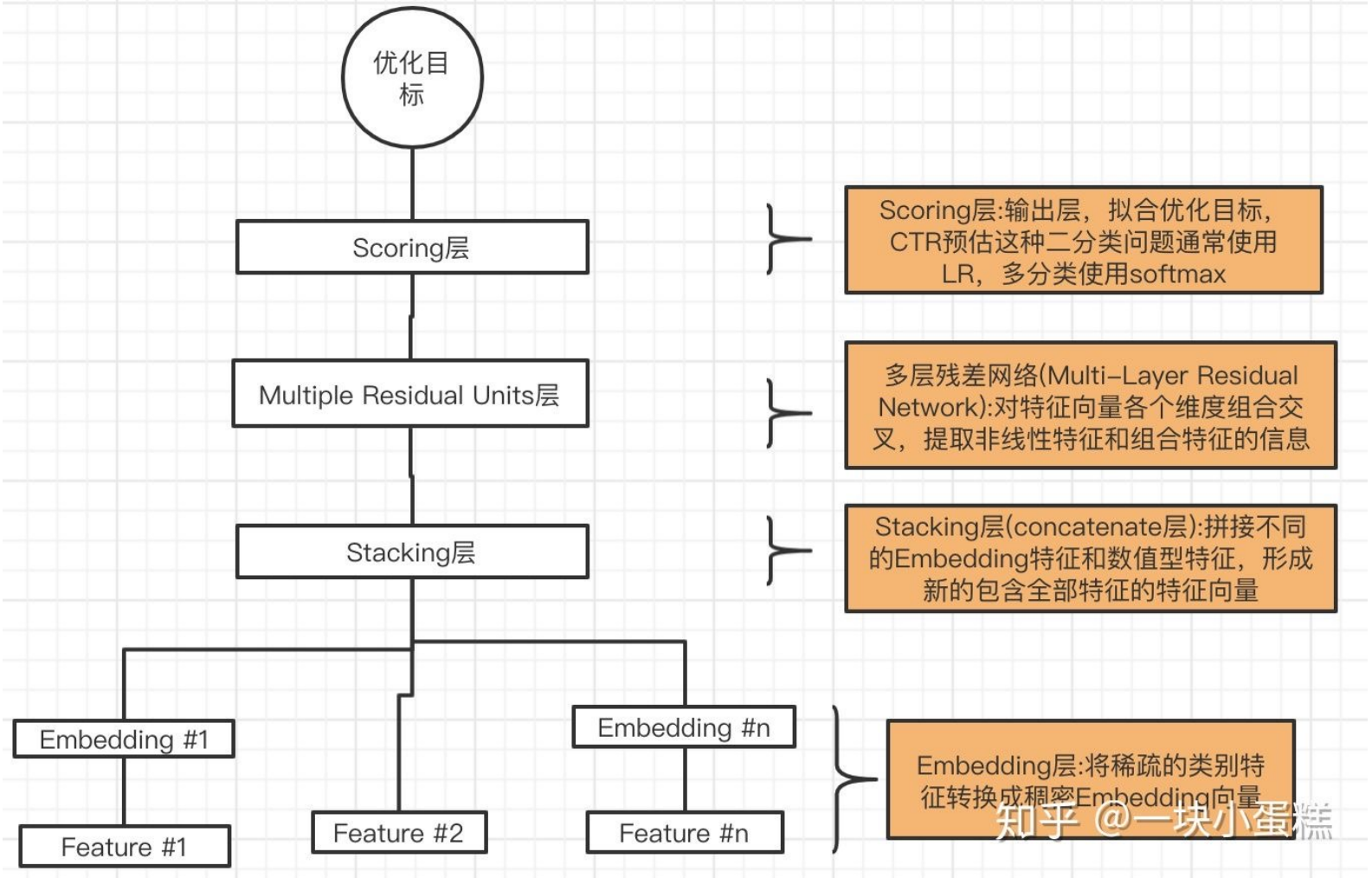
→

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

Deep Crossing 模型

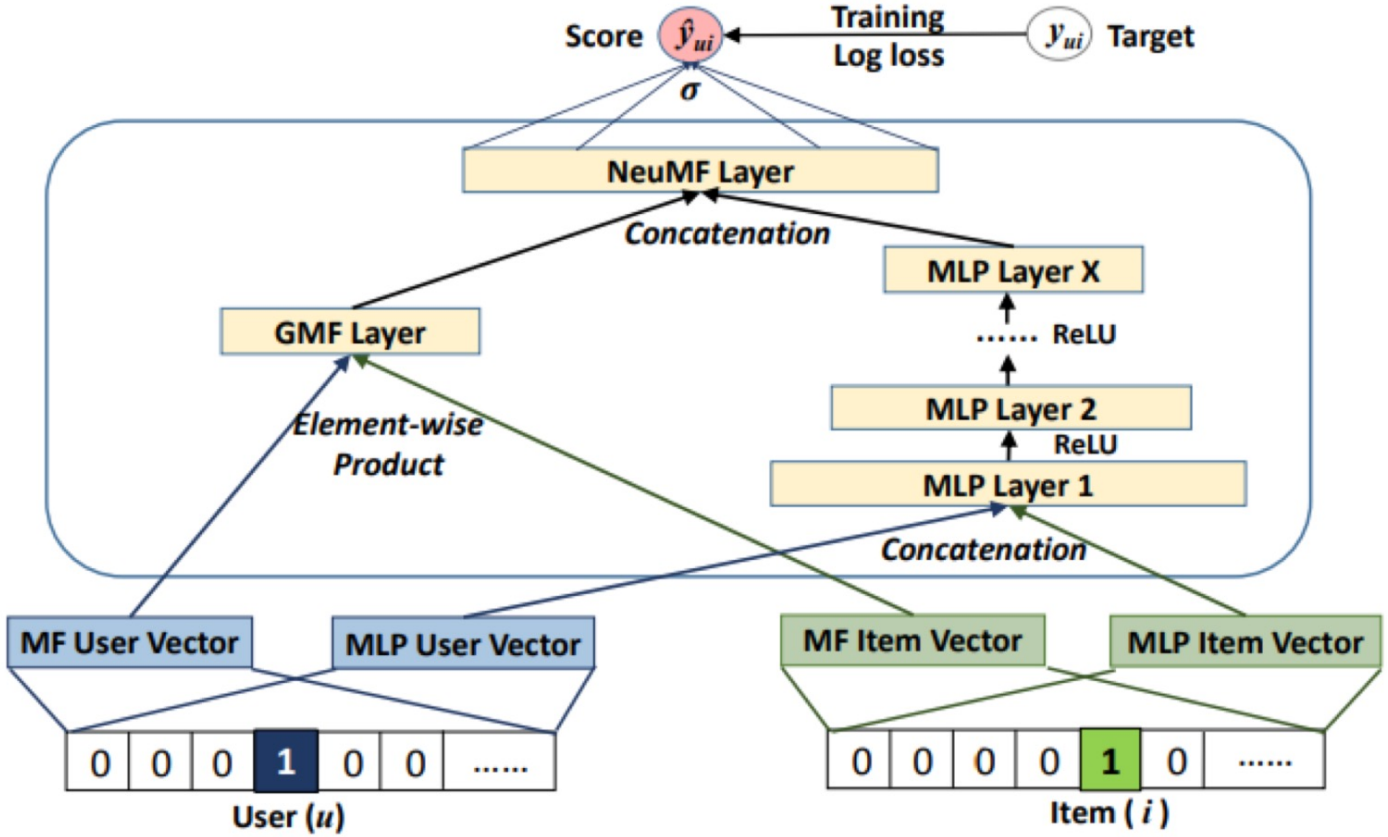
Deep Crossing：由微软于2016年发布，用于其搜索引擎Bing中的搜索广告推荐场景。Deep Crossing完善了深度学习在推荐领域的实际应用流程，提出了一套完整的从特征工程、稀疏向量稠密化、多层神经网络进行优化目标拟合的解决方案，开启了无需任何人工特征工程的时代。



Ying Shan, et al. Deep Crossing: Web-Scale Modeling without Manually Crafted Combinatorial Features. *SIGKDD 2016*

NeuralCF 模型

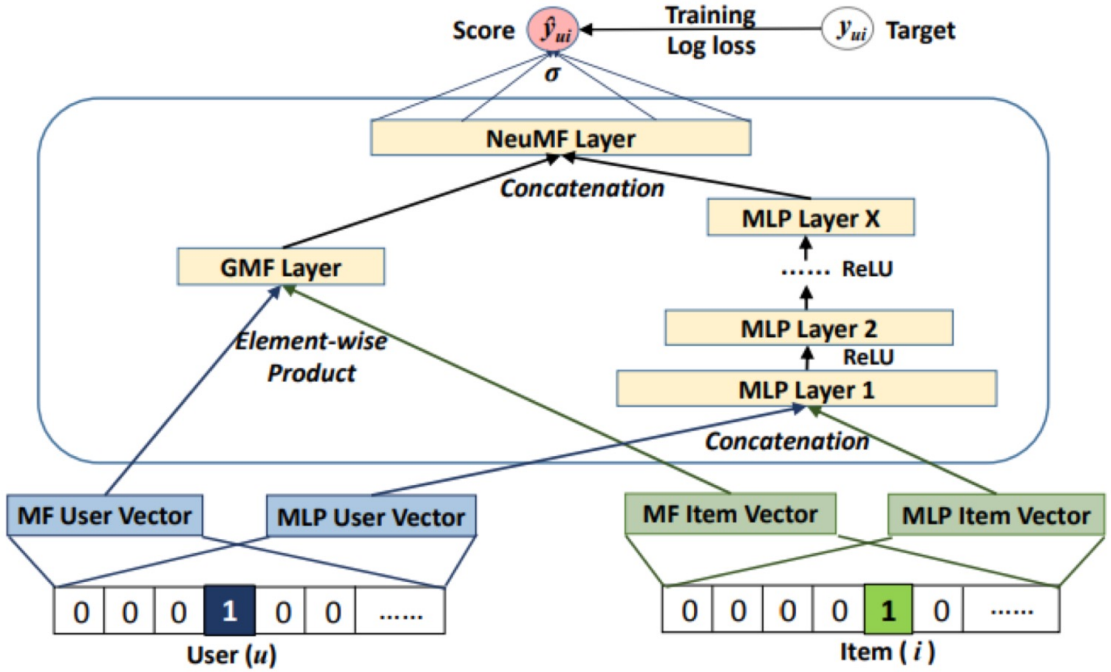
NeuralCF是一个典型的基于深度学习推荐算法。它结合了传统矩阵分解和多层感知机，可以同时抽取低维和高维的特征，具有不错的推荐效果。



Neural Collaborative Filtering, WWW, 2017

NeuralCF 模型

- GMF (Generalized Matrix Factorization) 输入是用户 id 和物品 id，并经过一个嵌入层，将其转化成用户特征矢量和物品特征矢量，然后将用户特征矢量和物品特征矢量对位相乘，得到 GMF 层。
- MLP (Multi-Layer Perceptron) 输入同样也是用户 id 和物品 id 经过嵌入后得到对应的特征矢量。中间部分是多个全连接层串联起来并对每层的输出加上一个激活函数 (Relu)，学习非线性关系。
- 最后将 GMF 层和 MLP 层进行连接，并经过最后一个激活函数为 sigmoid 的全连接层，输出一个 0 到 1 之间的推荐值。



Neural Collaborative Filtering, WWW, 2017

PNN 模型

2016年的PNN模型在Deep&Crossing的基础上使用乘积层(Product Layer)代替Stacking层。即不同特征的Embedding向量不再是简单的拼接，而是通过Product操作两两交互。这里的Product操作包含两种：内积操作和外积操作。

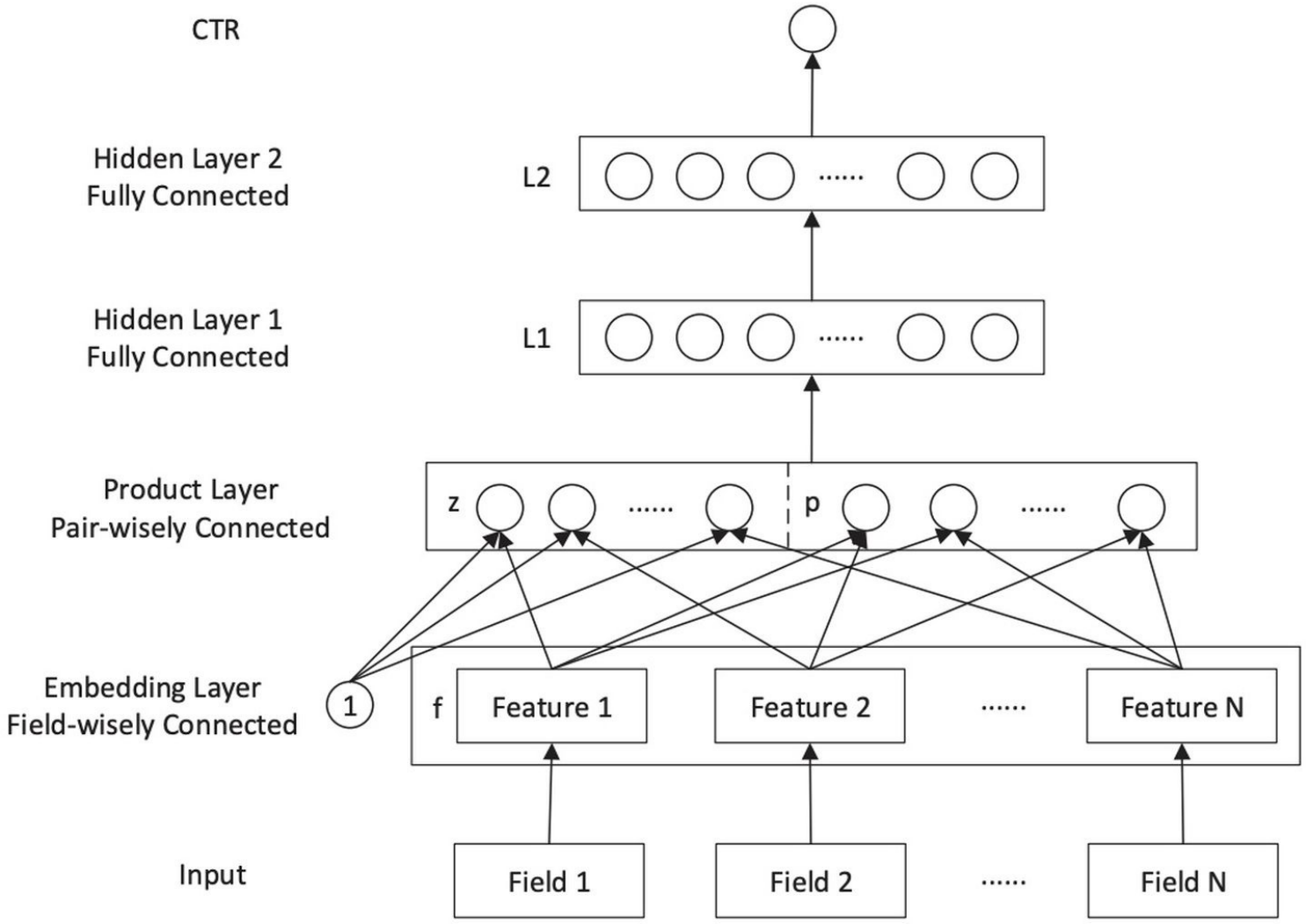


Fig. 1: Product-based Neural Network Architecture.

Qu, Yanru, et al. "Product-based neural networks for user response prediction." *ICDM,* 2016.

Memorization vs Generalization

- Memorization:

  之前大规模稀疏输入的处理是：通过线性模型 + 特征交叉。通过特征交叉能够带来很好的效果并且可解释性强。但是Generalization（泛化能力）需要更多的人工特征工程。

- Generalization:

  相比之下，DNN几乎不需要特征工程。通过对低纬度的dense embedding进行组合可以学习到更深层次的隐藏特征。但是，缺点是有点over-generalize（过度泛化）。推荐系统中表现为：会给用户推荐不是那么相关的物品，尤其是user-item矩阵比较稀疏并且是high-rank（高秩矩阵）

Wide&Deep 模型

- **Wide部分：利用了广义线性模型，提高可解释性。**

  在大规模的在线推荐系统中，logistic regression应用非常广泛，因为其简单、易扩展、可解释性。LR的输入多半是二值化后的one-hot稀疏特征。Memorization可通过在稀疏特征上做特征交叉来实现，例如：user_installed_app=netflix，impression_app=pandora，当user_installed_app与impression_app的取值都为1时，其组合特征AND(user_installed_app=netflix, impression_app=pandora)的值则为1，否则为0。

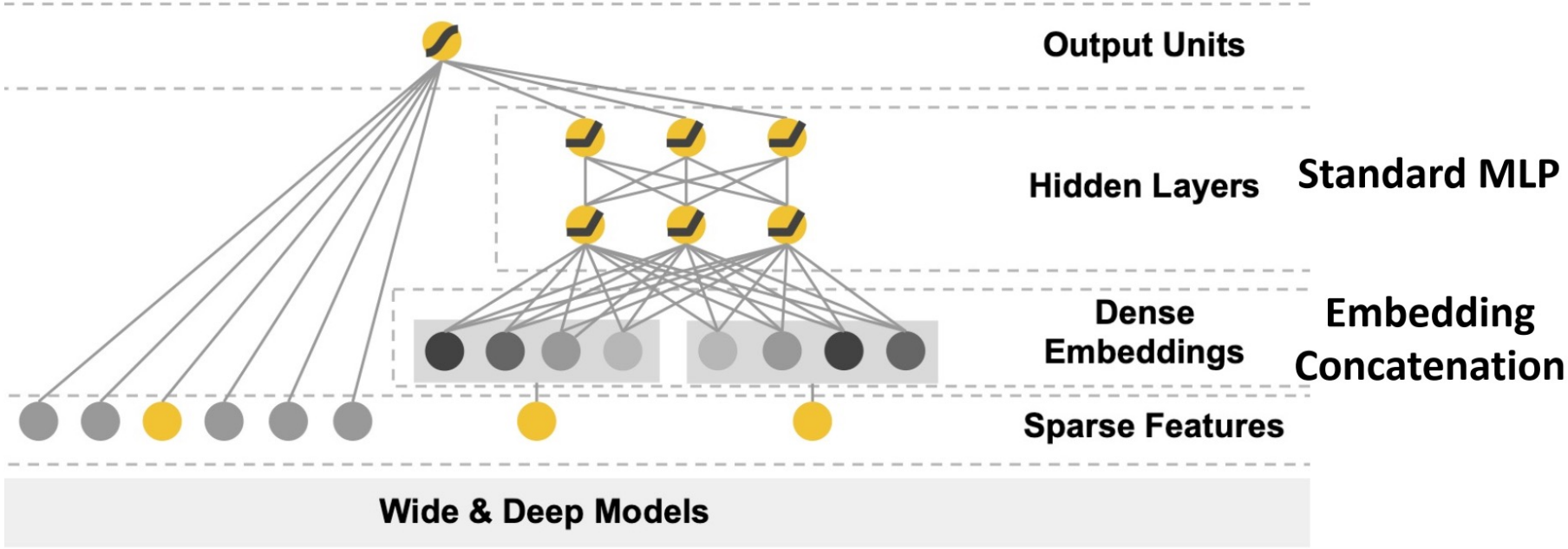
  缺点：无法学习高阶组合特征，并且需要进行人工特征工程。

- **Deep部分：主要是发现训练集中未出现的高阶组合特征。**

  Embedding-based模型可以在很少的特征工程情况下，通过学习一个低维的embedding vector来学习训练集中从未见过的组合特征。例如，FM与DNN。不需要进行复杂的特征工程。

  缺点：当query-item矩阵是稀疏并且是high-rank的时候（比如user有特殊的爱好，或item比较小众），很难非常效率的学习出低维度的表示。这种情况下，大部分的query-item都没有什么关系。但是dense embedding会导致几乎所有的query-item预测值都是非0的，这就导致了推荐过度泛化，会推荐一些不那么相关的物品。

Wide&Deep 模型



- 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

Deep&Cross 模型

Deep&Cross：斯坦福和Google合作基于Wide&Deep的改进。主要思路是使用Cross网络替代Wide部分，目的是通过多层交叉(Cross layer)增加特征之间的交互力度；Deep部分则与Wide&Deep保持一致。

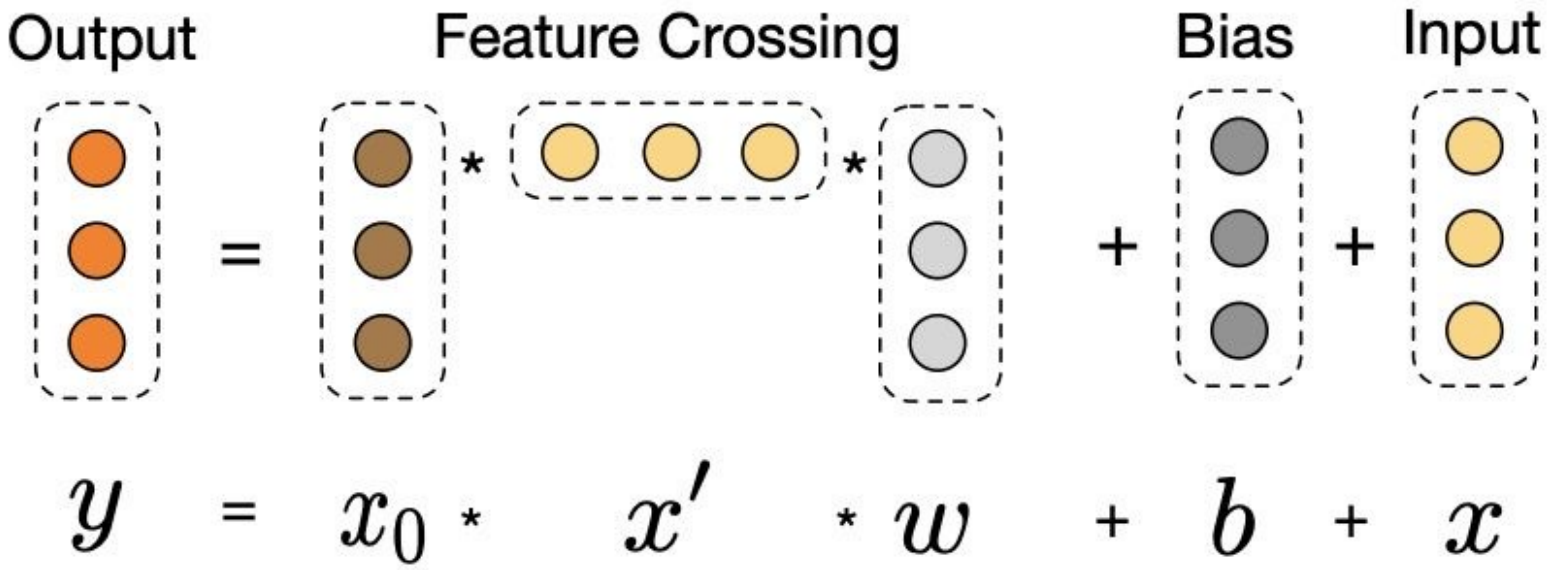


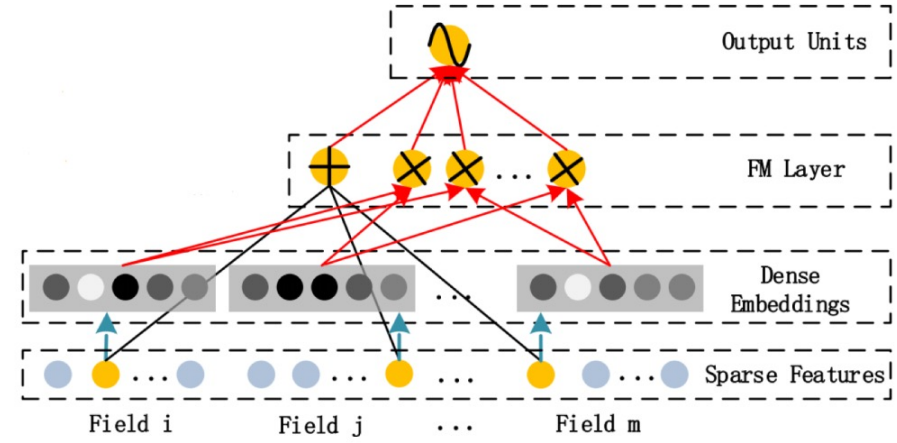
Figure 2: Visualization of a cross layer.

DeepFM 模型

2017年由哈工大&华为提出，使用FM替换Wide&Deep的Wide部分，加强浅层网络组合特征的能力。

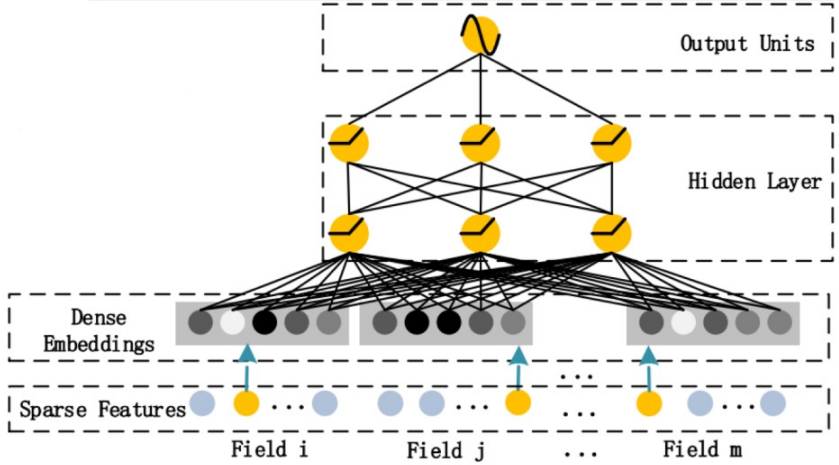
DeepFM的改进目的和Deep&Cross的目的是一致的，只是采用的手段不同。

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)



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

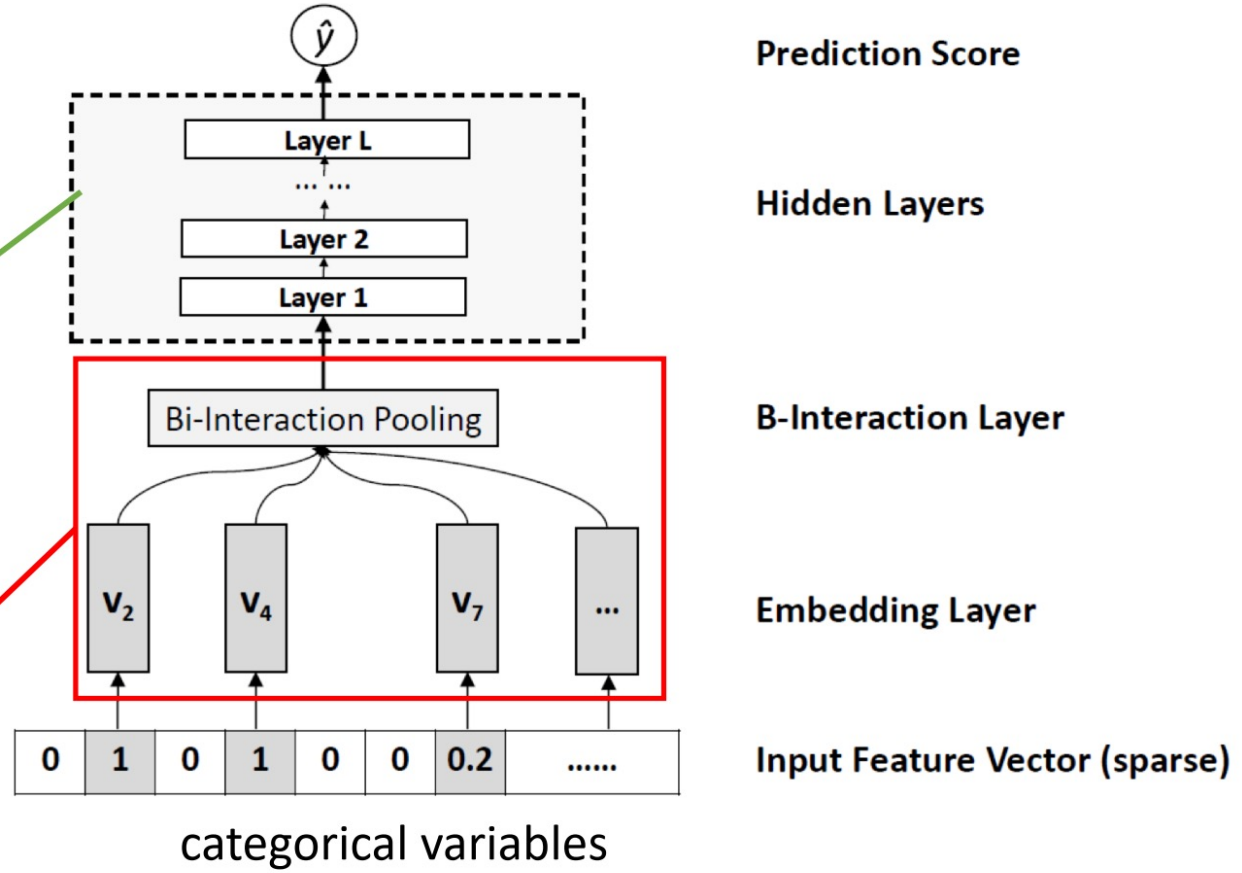
Prediction Model: $\hat{y} = sigmoid(y_{FM} + y_{DNN})$

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

Neural FM 模型

2017年新加坡国立大学提出NeuralFM模型，目的是使用表达能力更强的函数替换原本FM中二阶隐向量内积的部分

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

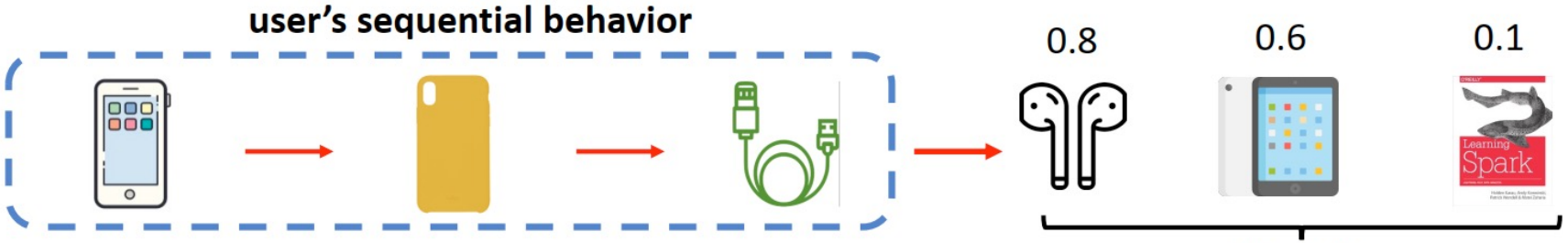


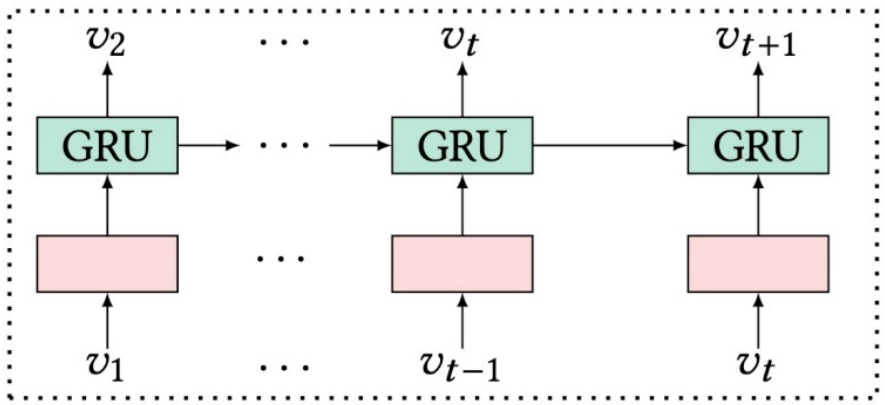
Bilinear Interaction Pooling:

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

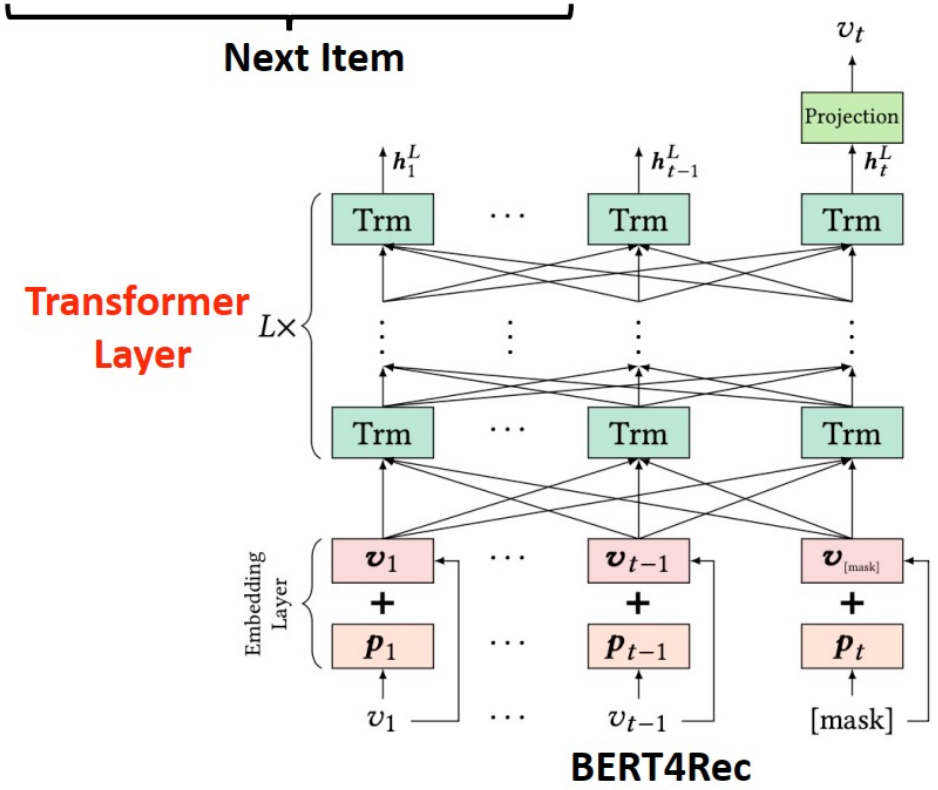
Neural Factorization Machines for Sparse Predictive Analytics, SIGIR, 2017

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.

提纲

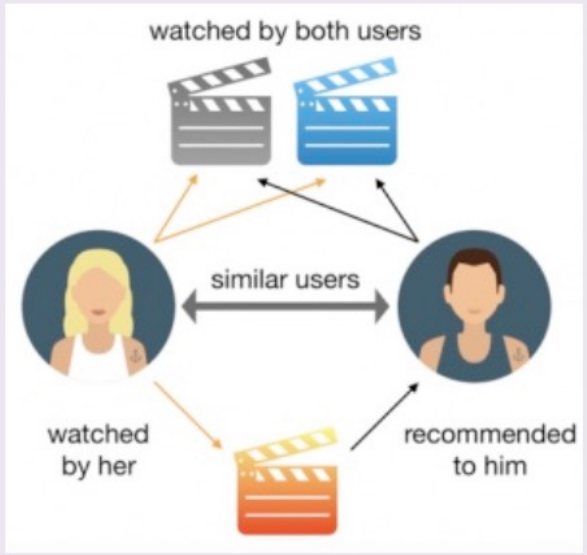
1 推荐系统概述

2 DNN推荐算法

3 RL推荐算法

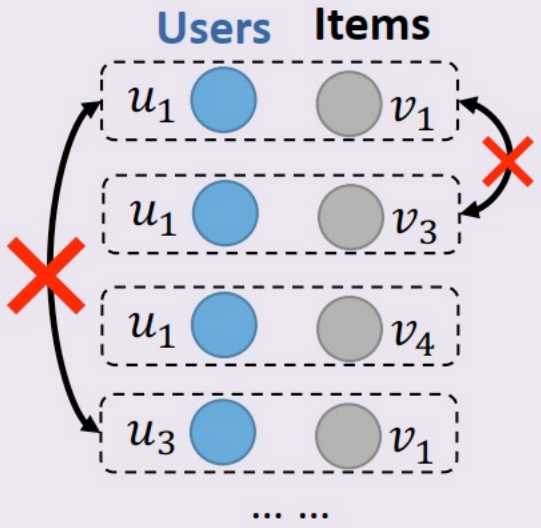
4 GNN推荐算法

DNN模型的缺点



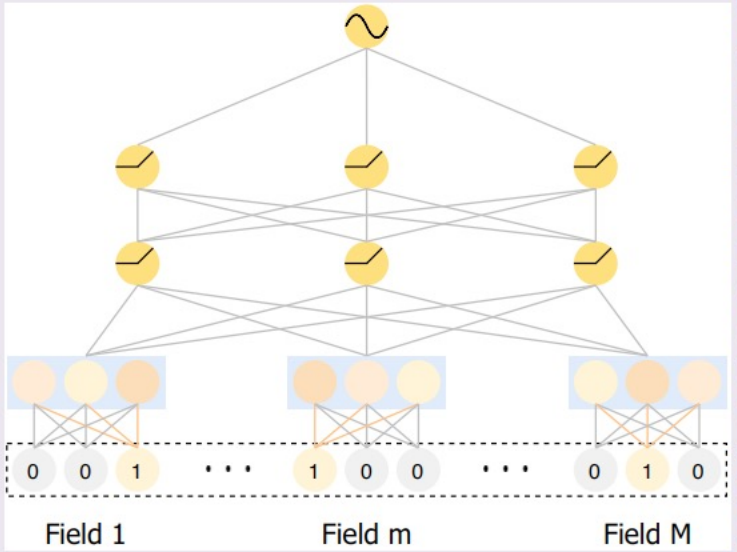
推荐策略问题

- 离线优化
- 短期回报



图结构数据问题

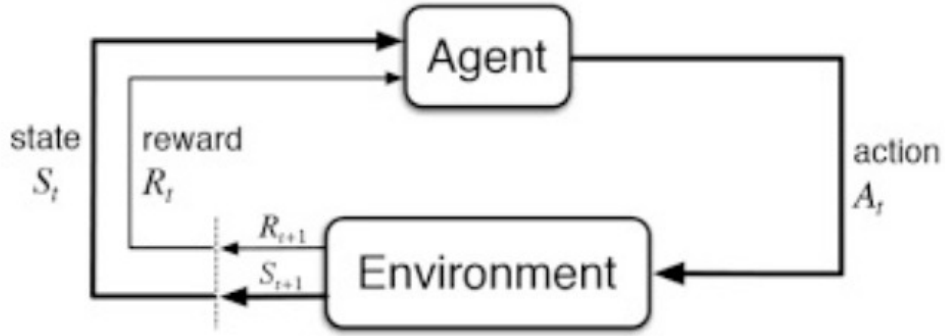
- 用户与项目信息相互独立，忽略了内在关联



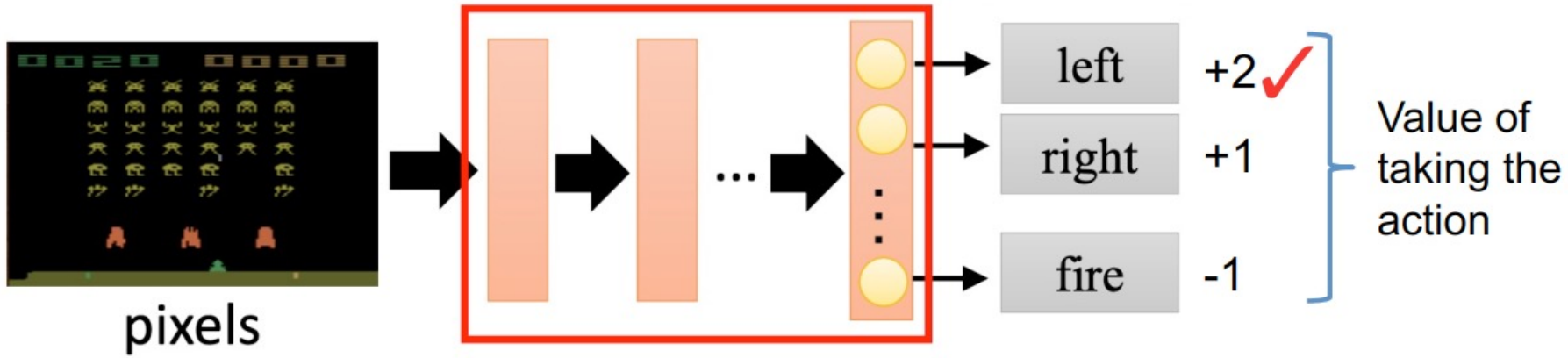
模型设计问题

- 大多依赖手动设计

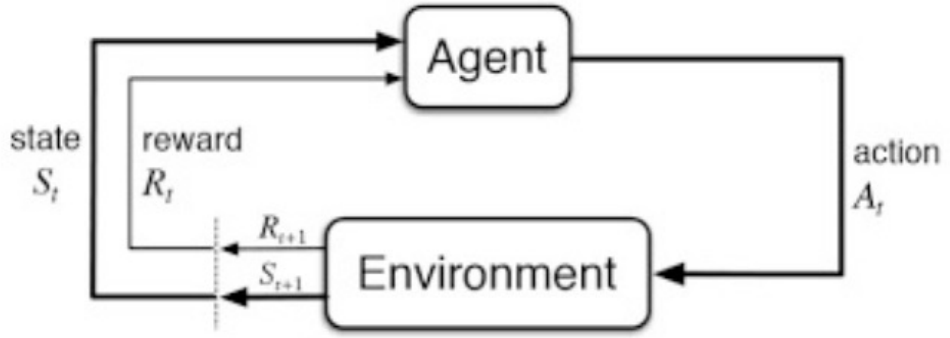
目标：选择合适的动作（Actions），使得回报（Rewards）最大化



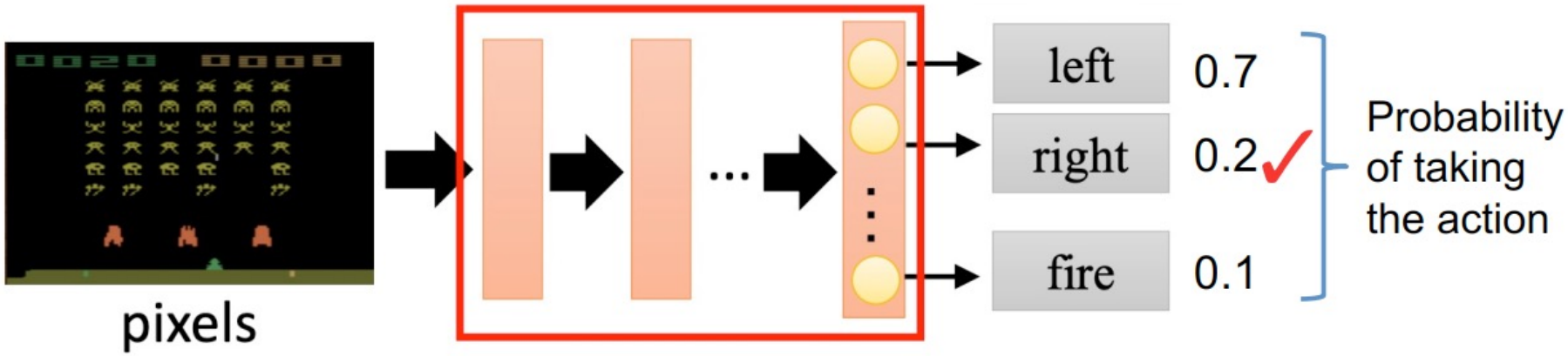
基于价值（Value-Based）的强化学习



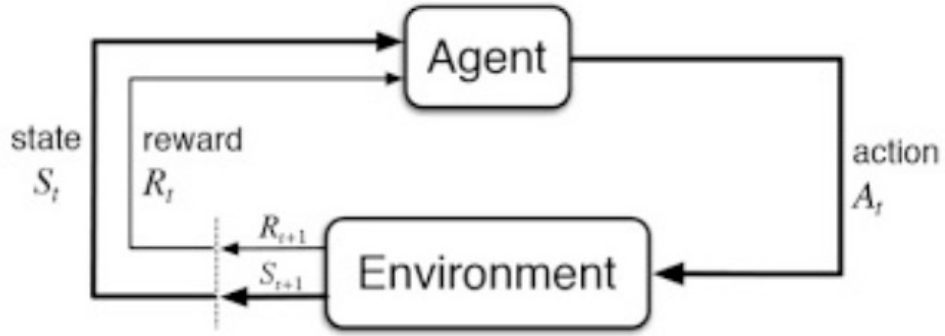
目标：选择合适的动作（Actions），使得回报（Rewards）最大化



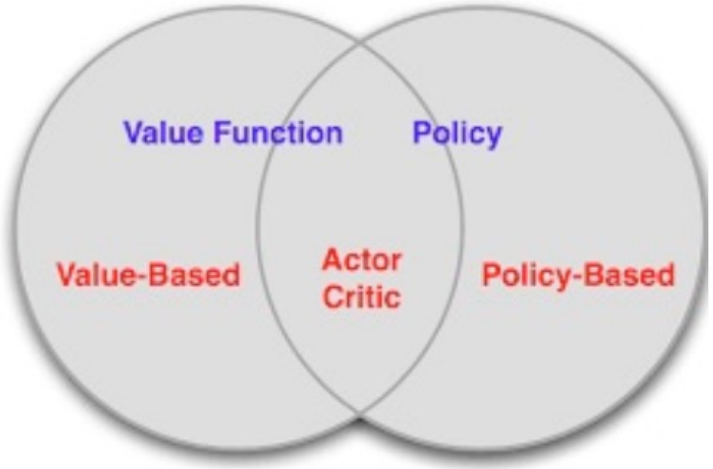
基于策略（Policy-Based）的强化学习



目标：选择合适的动作（Actions），使得回报（Rewards）最大化

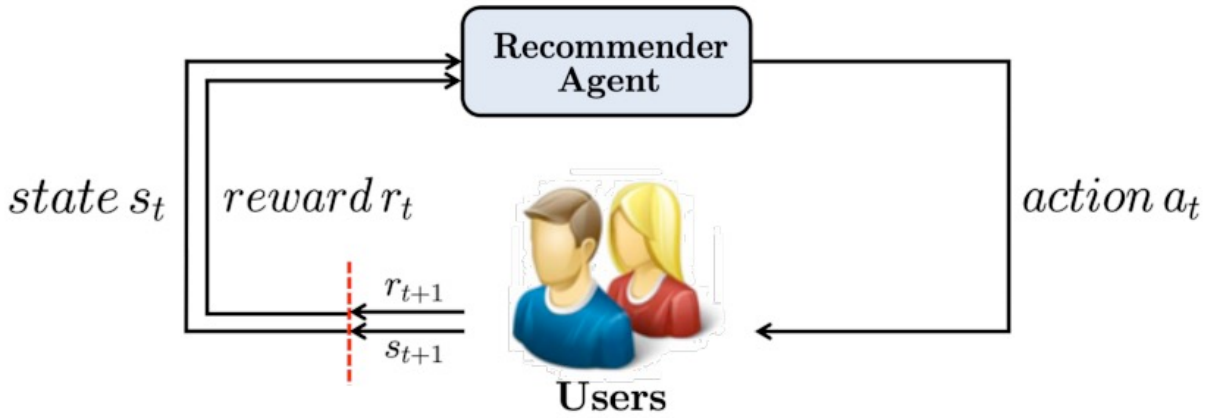


Actor-Critic 结构



强化学习 + 推荐系统？

在用户与项目/物品交互的过程中，持续更新推荐策略？

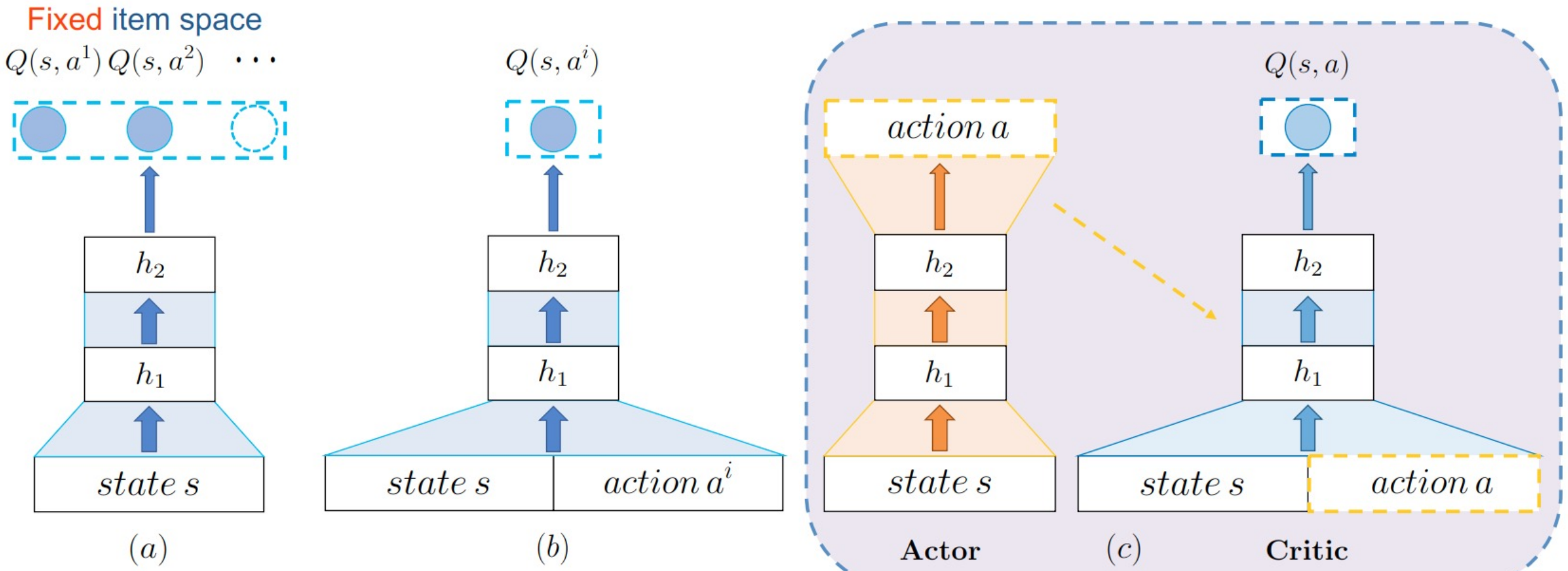


是否能够最大化长期收益？

强化学习 + 推荐系统

- 系统向用户推荐一页物品列表
- 用户提供对该商品列表的实时反馈，系统更新推荐策略
- 系统推荐新的商品列表

Actor-Critic 网络



$$Q^*(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q^*(s',a') | s,a\right]$$

max → enumerating all possible items

$$Q(s,a) = \mathbb{E}_{s'}\left[r + \gamma Q(s',a') | s,a\right]$$

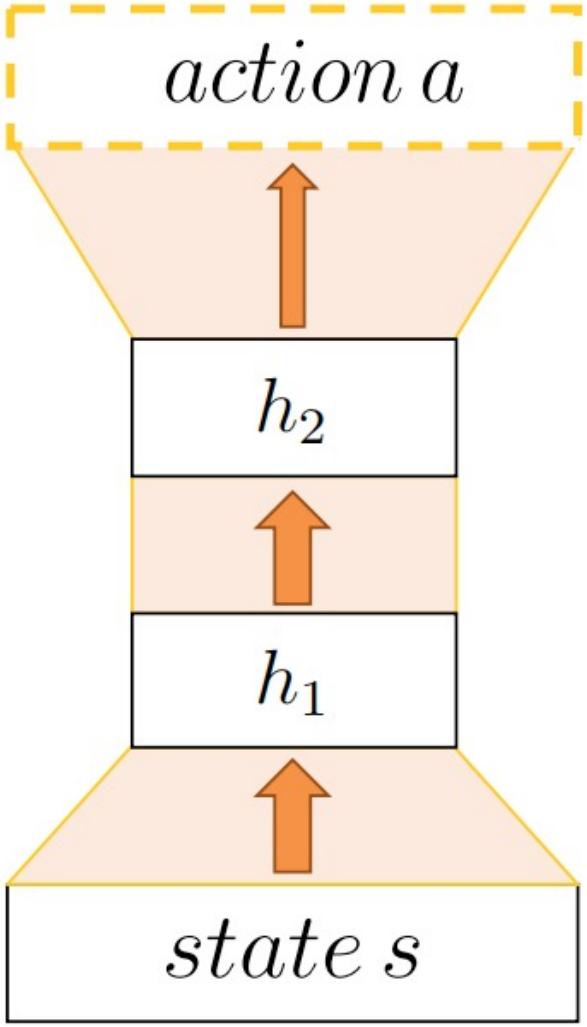
Actor 的设计

目标：

- 根据用户的浏览历史记录，生成一页新的推荐列表

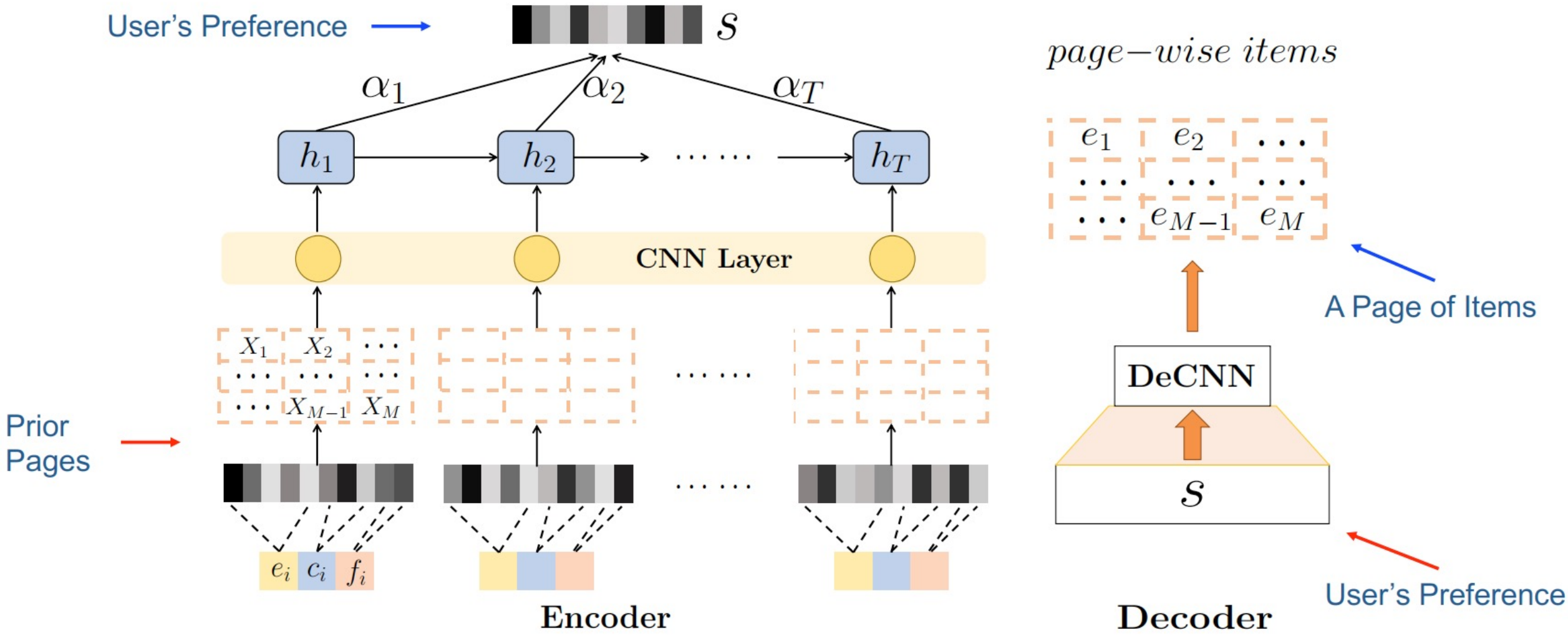
挑战：

- 从用户的实时反馈中获取信息
- 推荐相互补充的商品列表
- 将推荐列表按照特定排列，显示在页面中



Actor

Actor 结构


User's Preference
s
α_1
α_2
α_T
h_1
h_2
h_T
CNN Layer
X_1 X_2
X_{M-1} X_M
Prior Pages
e_i c_i f_i
Encoder
page−wise items
e_1 e_2
e_{M-1} e_M
A Page of Items
DeCNN
s
User's Preference
Decoder

Actor 结构

- Three types of information
  - e_i: item's identifier
  - c_i: item's category
  - f_i: user's feedback

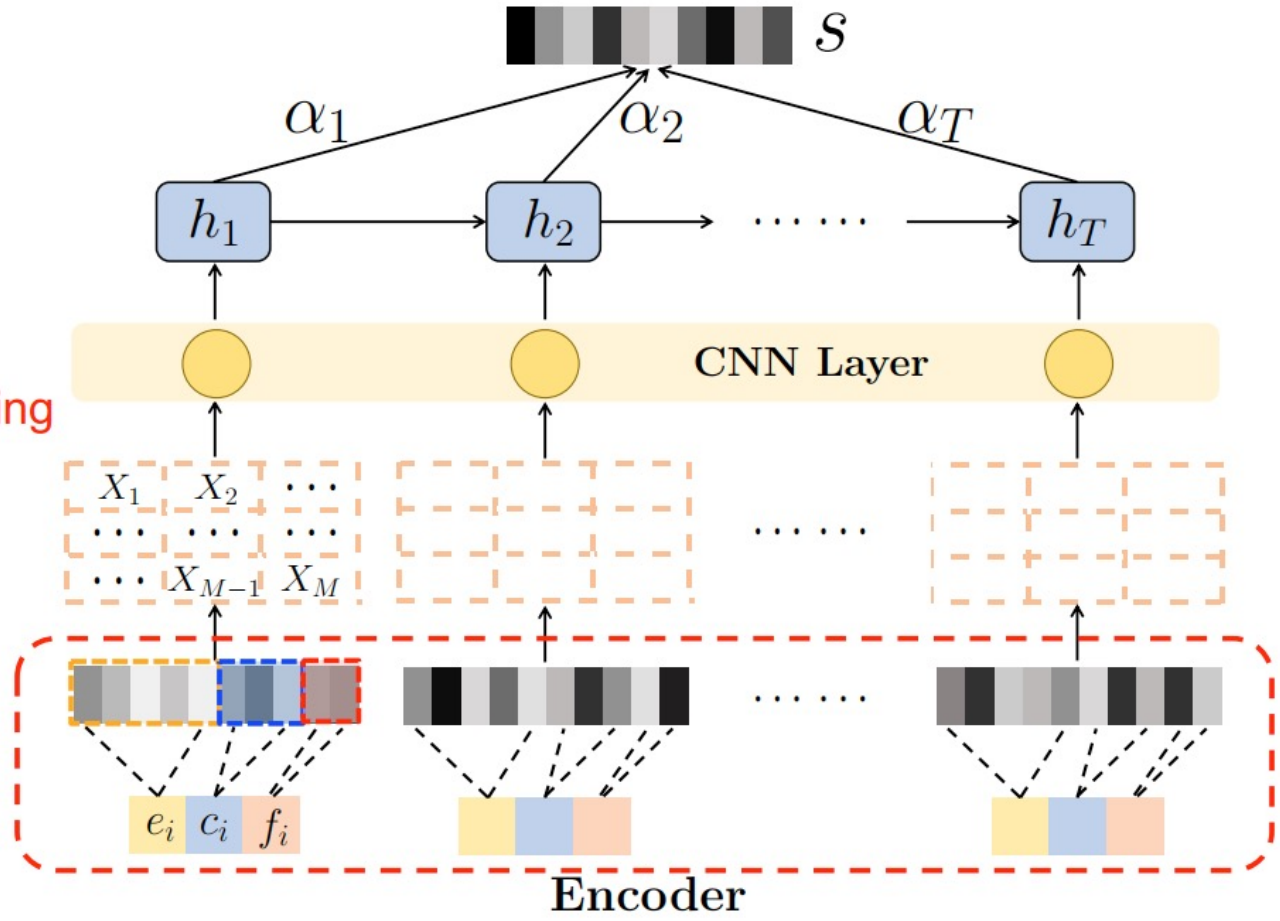
Item Embedding

Feedback Embedding

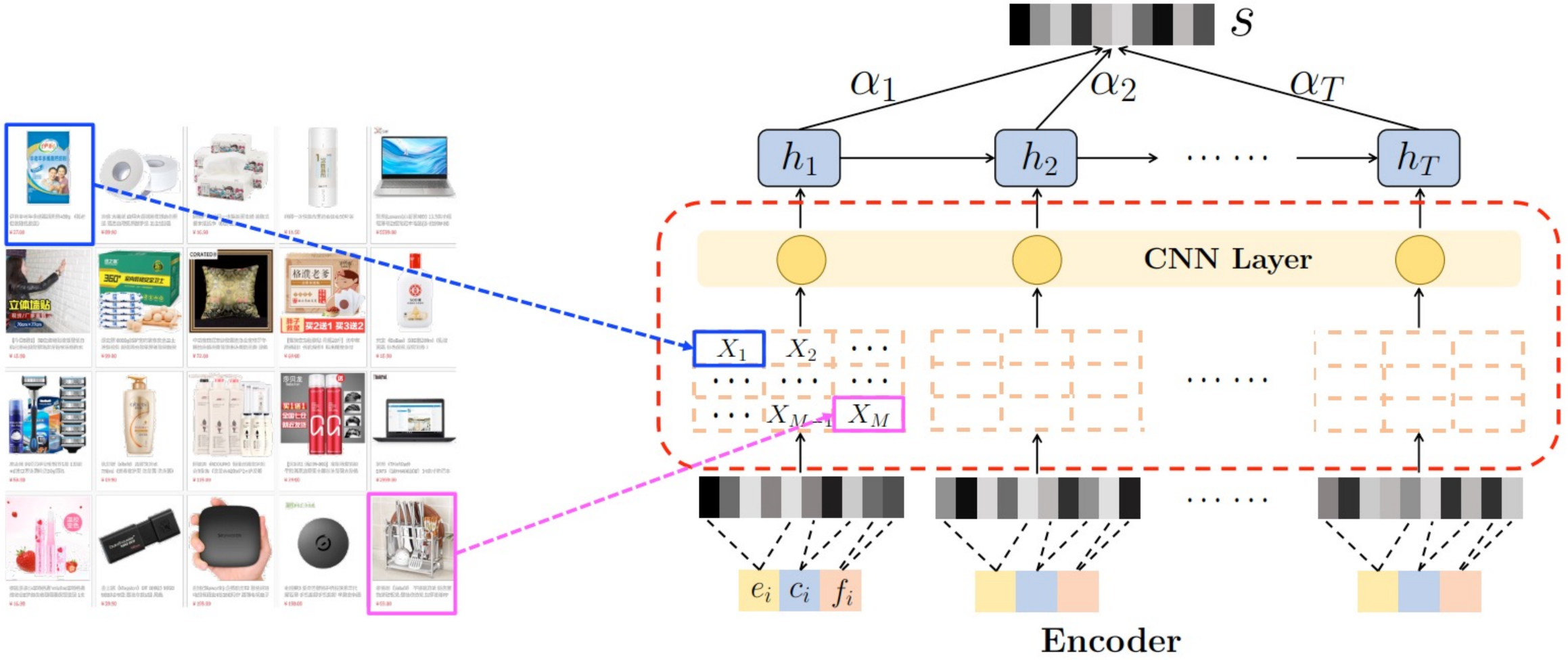
$$X_i = concat(E_i, C_i, F_i)$$
$$= \tanh\left(concat(W_E e_i + b_E, W_C c_i + b_C, W_F f_i + b_F)\right)$$

Identifier Embedding

Category Embedding



Page-Wise CNN 结构

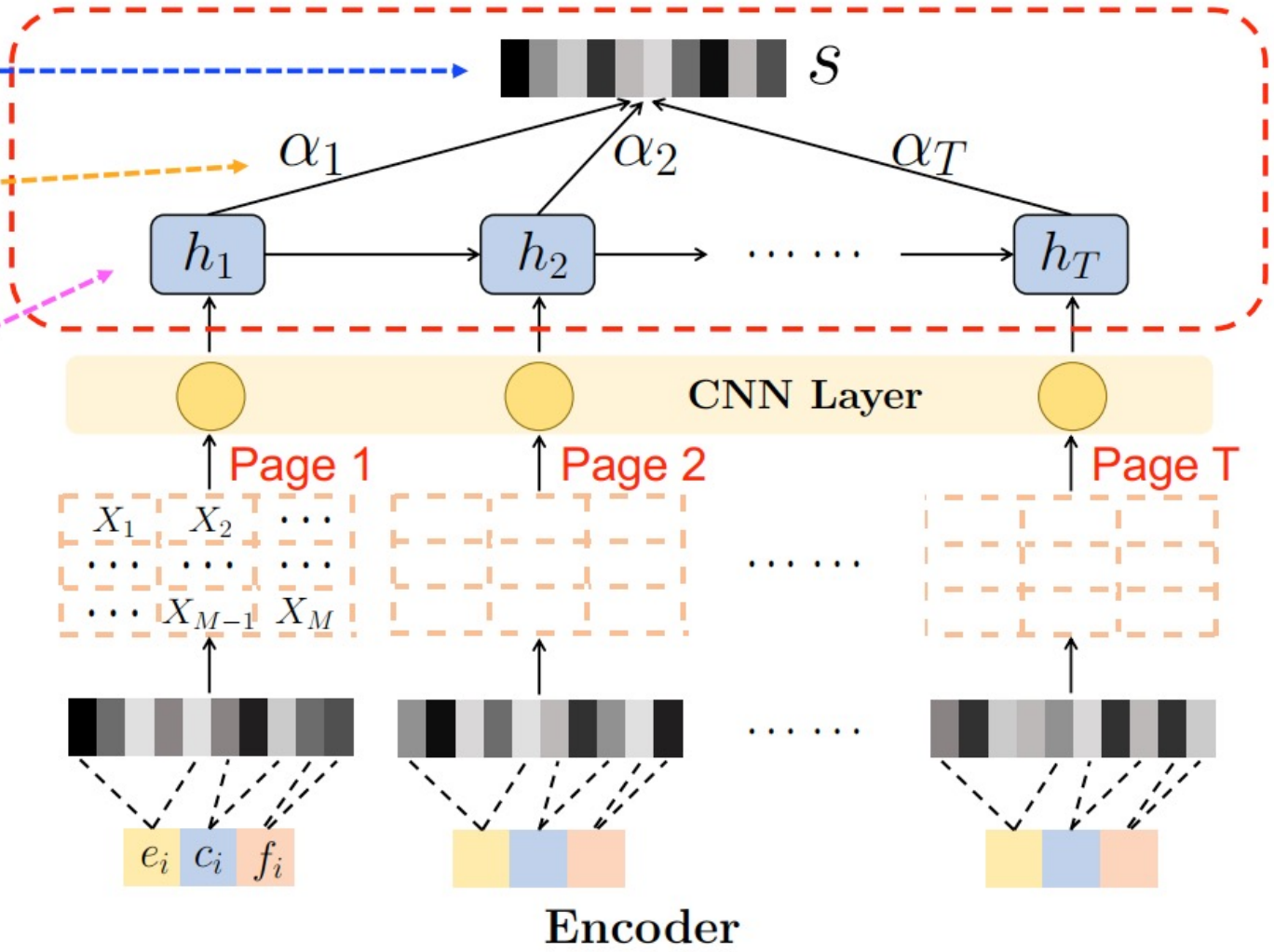


RNN + Attention

$$s = \sum_{t=1}^{T} \alpha_t h_t$$

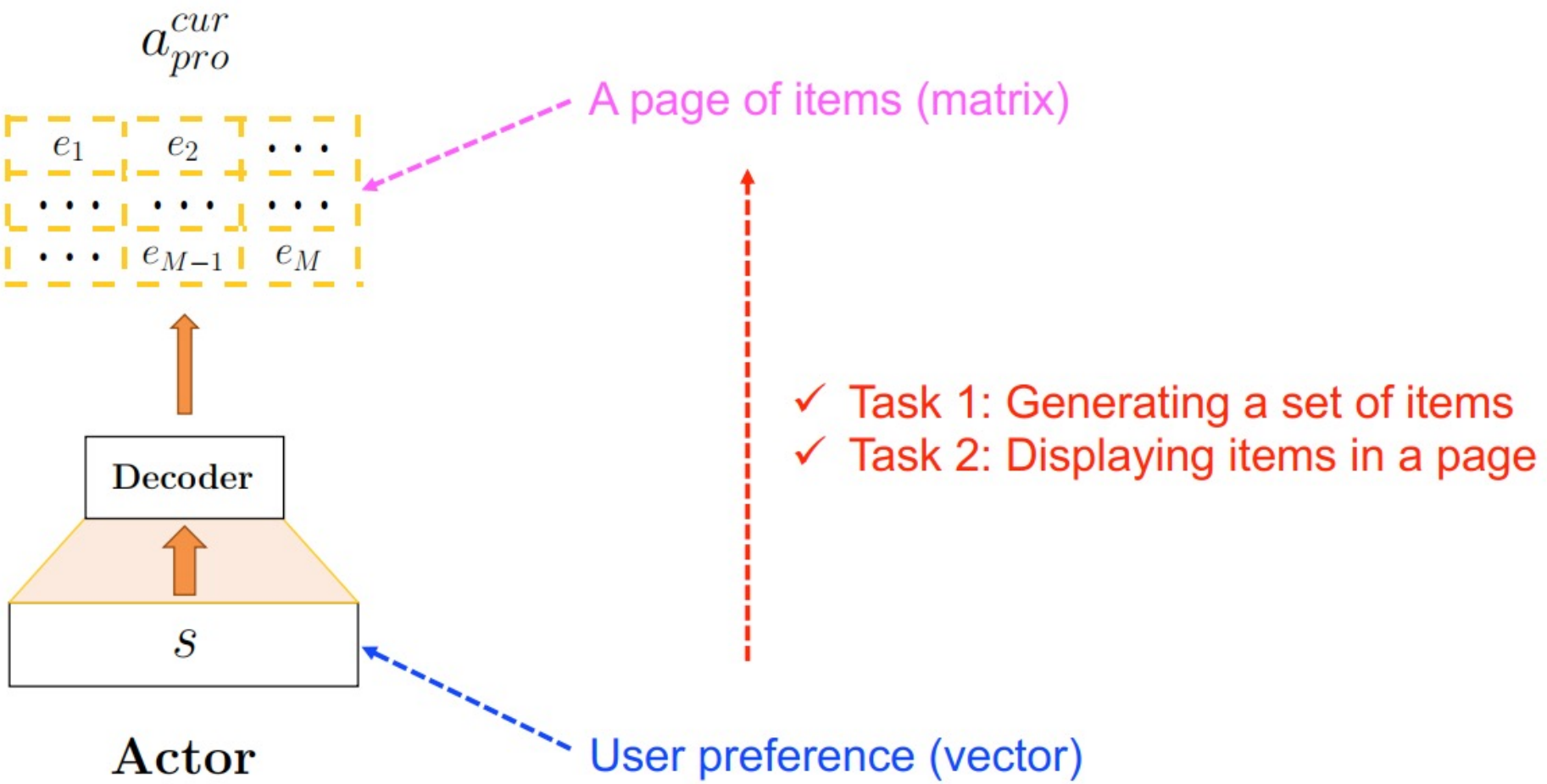
$$where\ \alpha_t = \frac{exp(W_\alpha h_t + b_\alpha)}{\sum_j exp(W_\alpha h_j + b_\alpha)}$$

$$z_t = \sigma(W_z E_t + U_z h_{t-1})$$
$$r_t = \sigma(W_r E_t + U_r h_{t-1})$$
$$h_t = (1 - z_t) h_{t-1} + z_t \hat{h}_t$$
$$\hat{h}_t = \tanh[W E_t + U(r_t \cdot h_{t-1})]$$



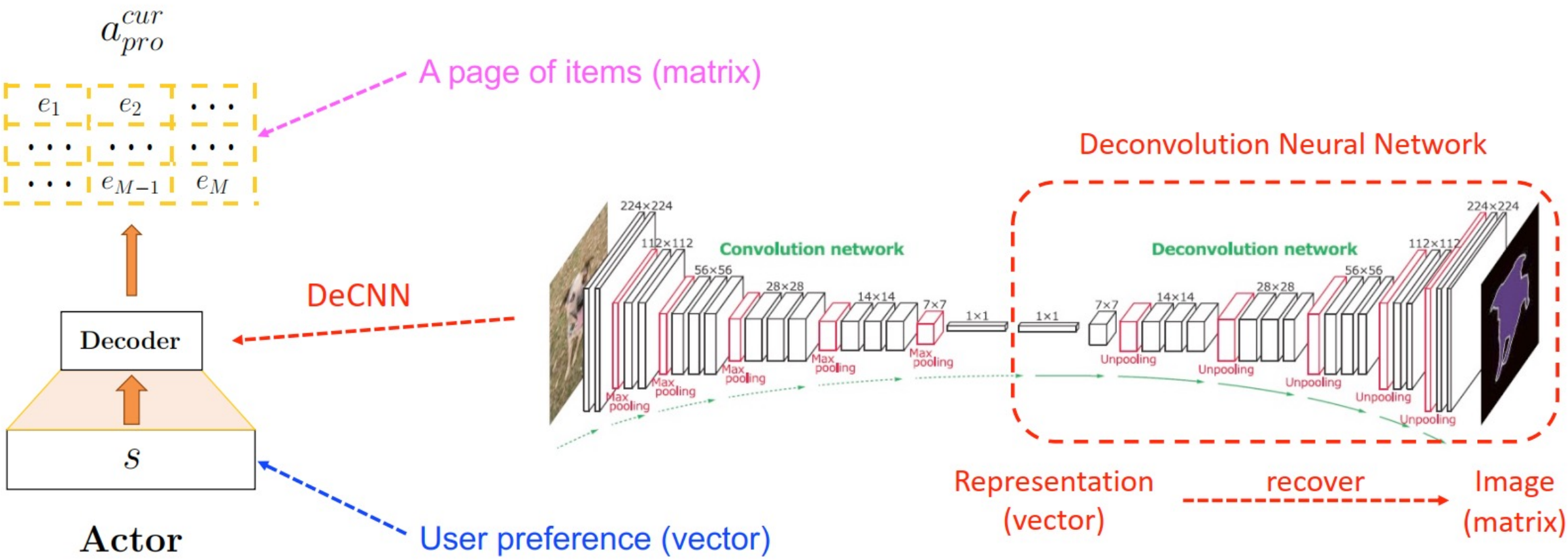
Decoder 的设计

目标：根据用户的偏好，生成一页新的推荐列表



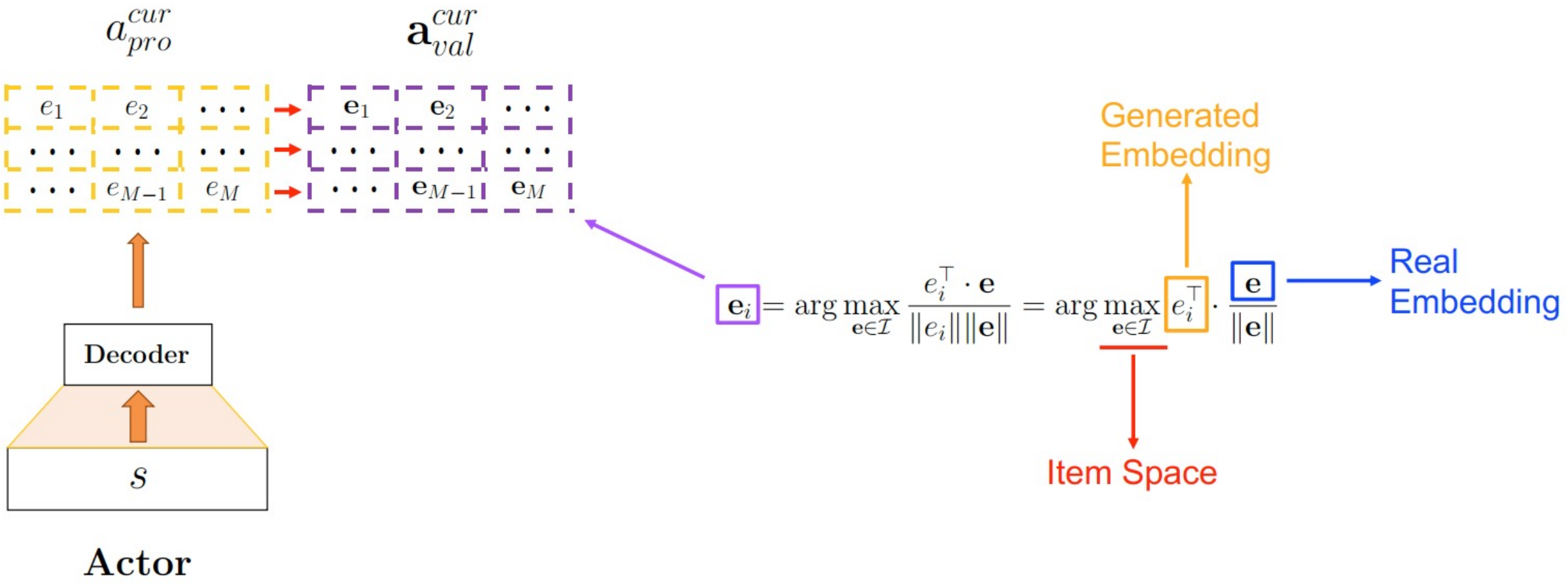
Decoder 的设计

目标：根据用户的偏好，生成一页新的推荐列表



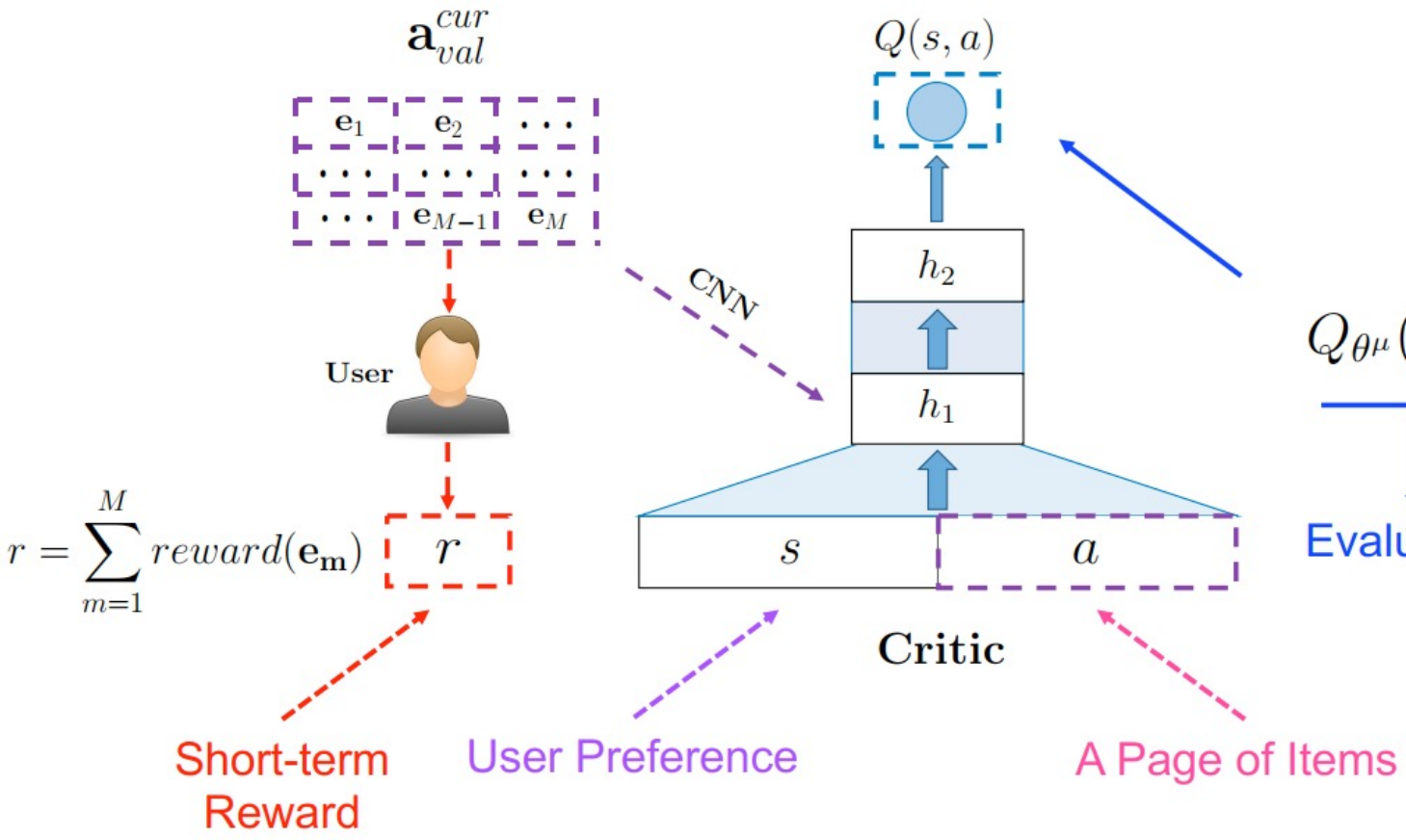
Decoder 的设计

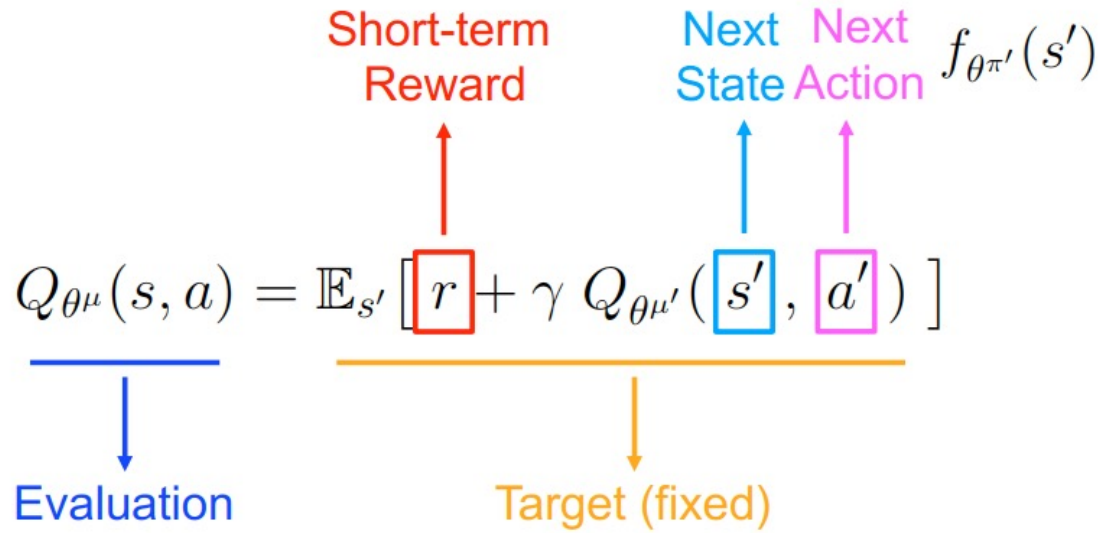
将Actor生成的向量表示转化为“项目”或“商品”的表示向量



Critic 的设计

目标：学习长期的价值函数





引入负面评价

用户行为并非单一：

- 正面评价：点击或者购买
- 负面评价：跳过或者离开页面

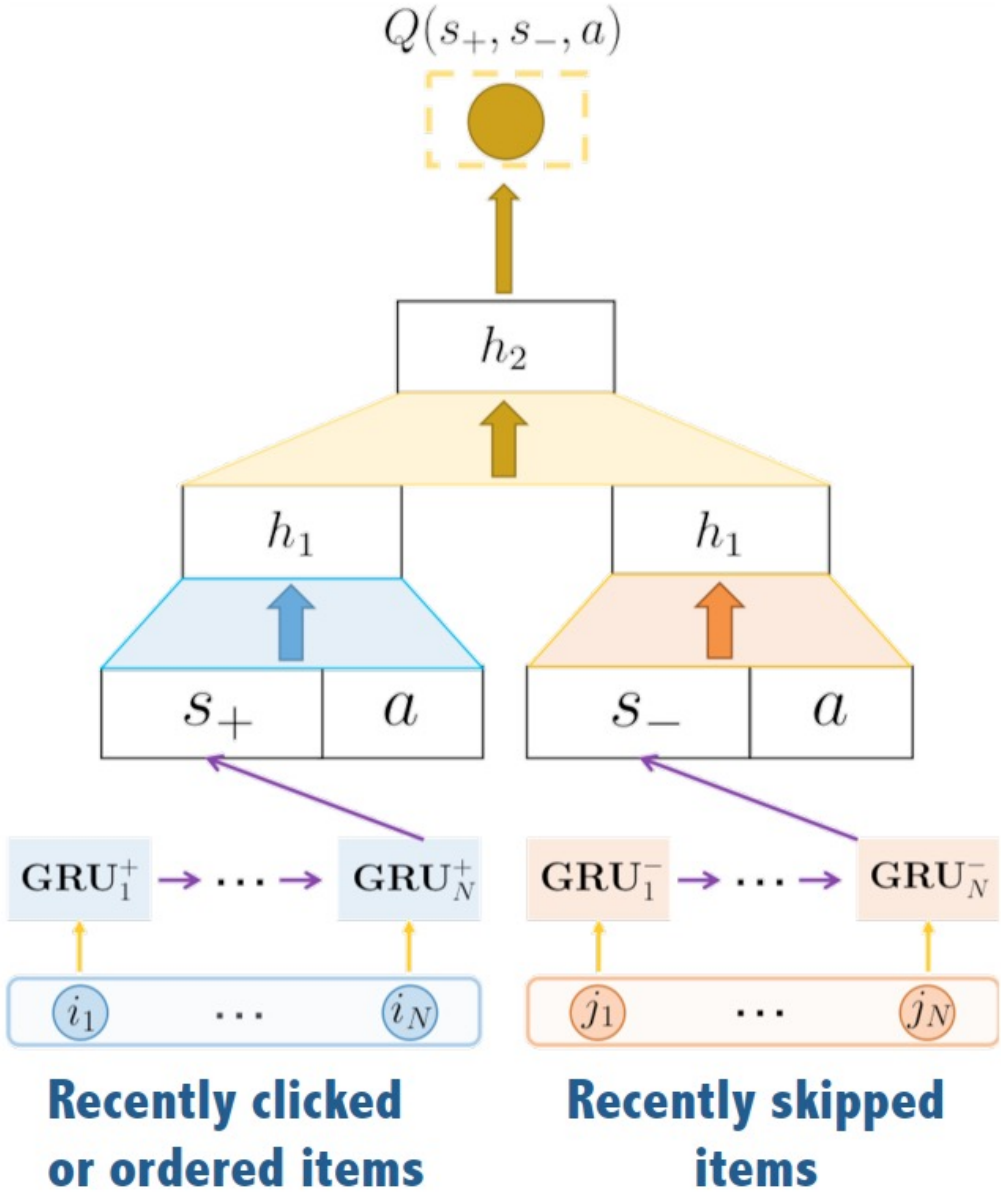
目标：

- 避免为用户推荐不合适的商品，造成用户流失

挑战：

- 对商品的负面评价可能掩盖掉正面评价
- 负面评价可能并非用户不喜欢该物品，只是当下不需要
- 错误的负面评价判别导致推荐算法出现偏差

设计新的 DQN 结构


$Q(s_+, s_-, a)$
h_2
h_1
h_1
s_+
a
s_-
a
$\text{GRU}_1^+ \rightarrow \cdots \rightarrow \text{GRU}_N^+$
$\text{GRU}_1^- \rightarrow \cdots \rightarrow \text{GRU}_N^-$
i_1 ... i_N
j_1 ... j_N
Recently clicked or ordered items
Recently skipped items

提纲

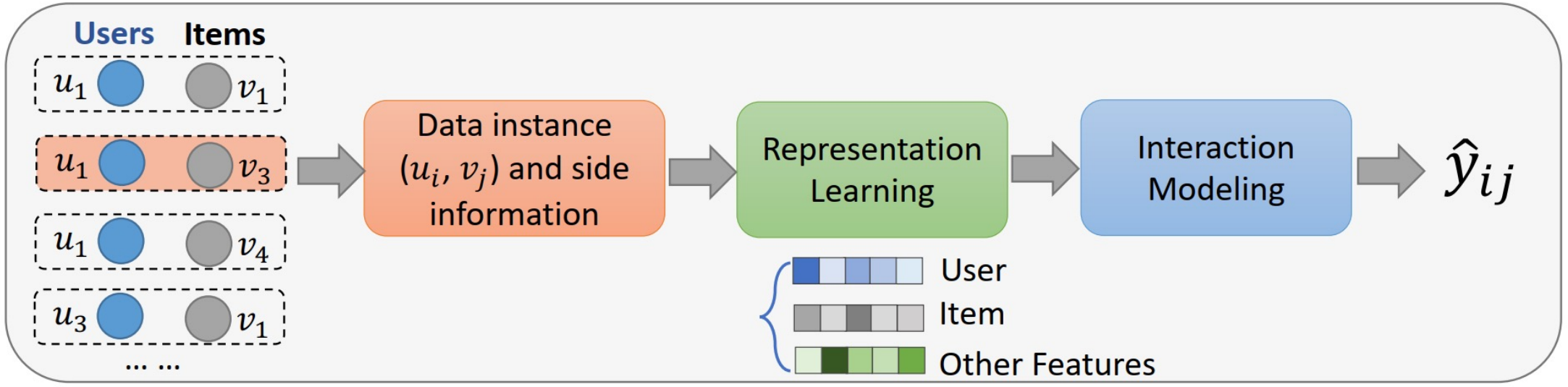
1 推荐系统概述

2 DNN推荐算法

3 RL推荐算法

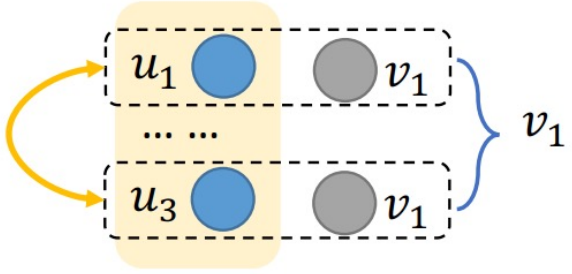
4 GNN推荐算法

常规推荐算法

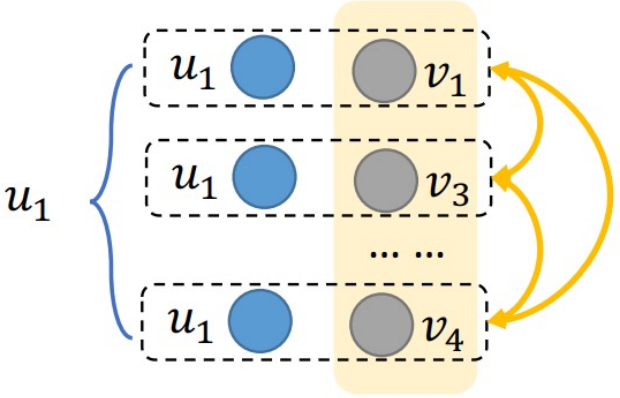


信息孤岛问题！

- **忽略了用户之间、项目之间存在的内在联系（高阶语义关系）**



Behavior similarity among users/items

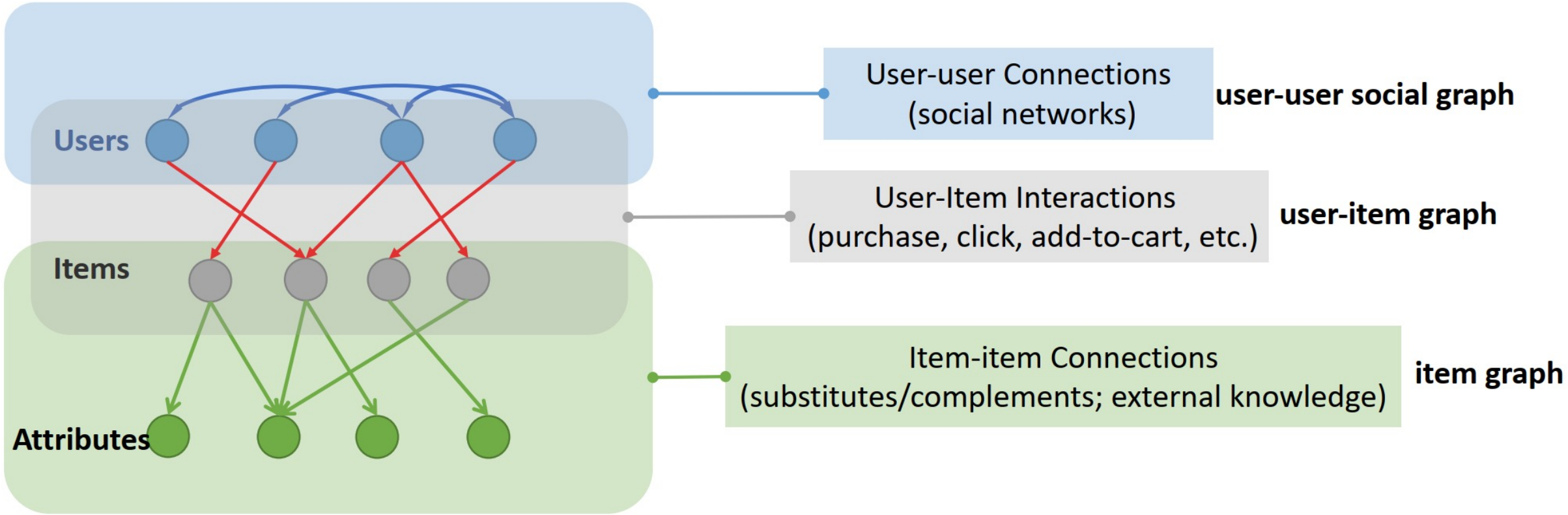


Learning and Reasoning on Graph for Recommendation, WSDM 2020

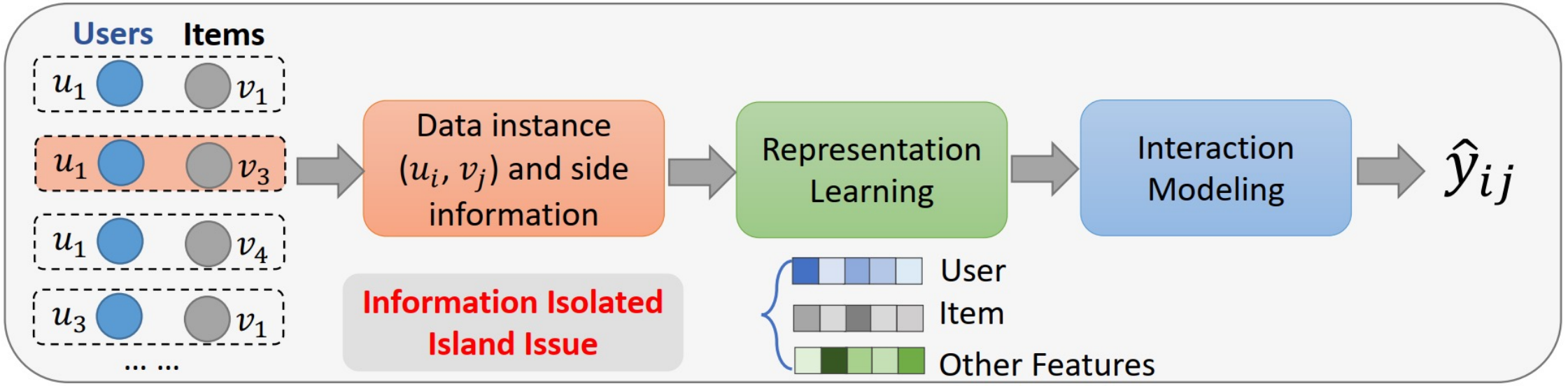
社会系统存在天然的图结构

绝大部分推荐系统数据都可以用图结构进行表达：

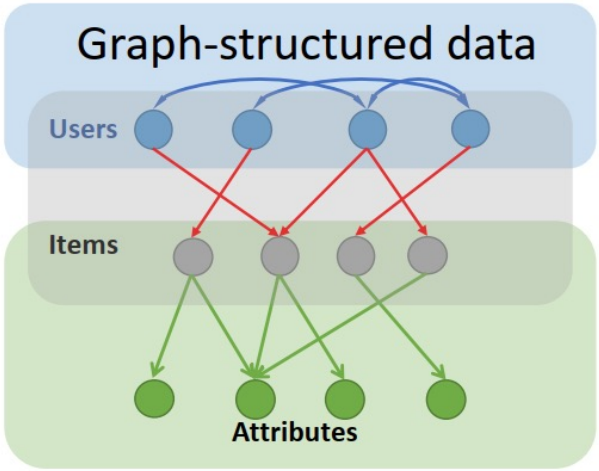
- 电商、内容分享、社交网络。。。



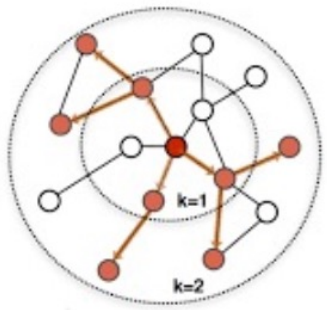
图关系引入推荐系统



Explore & Exploit Relations among Instances



Graph Neural Networks (GNNs)

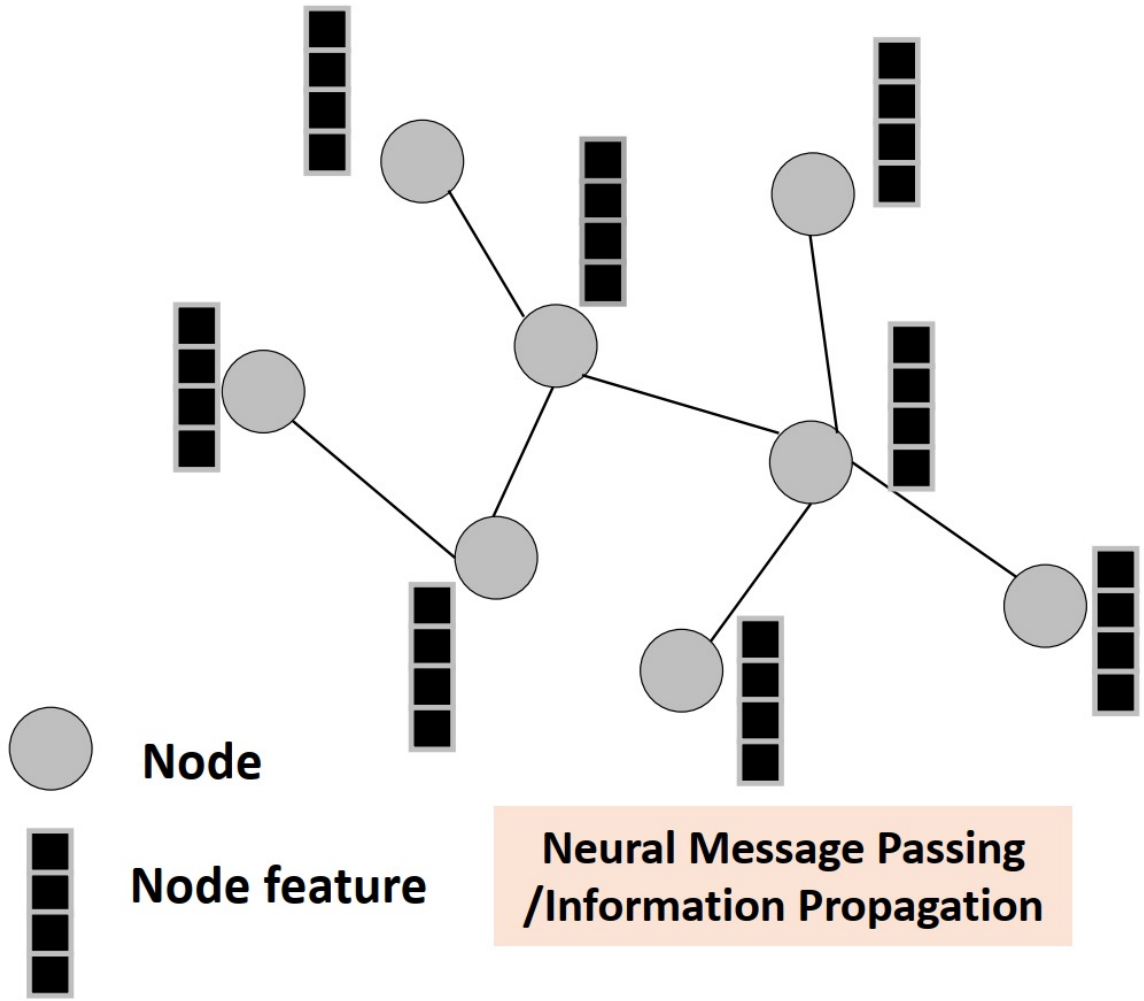


图神经网络（Graph Neural Networks, GNN）

Key idea： 利用神经网络聚合“网络/图”中的局部邻域信息

- 建模节点（Node）的邻域结构信息
- 聚合邻域信息
- 更新节点的表示向量

GNN可以天然地集成图结构数据中的节点特征及局部邻域拓扑结构。



Inductive Representation Learning on Large Graphs, NeuIPS, 2017.

基本思路：对邻域节点信息求平均，并利用神经网络更新表示向量

$$\mathbf{h}_v^0 = \mathbf{x}_v$$

Initial 0-th layer embeddings are equal to node v's features

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_1^k \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{\sqrt{|N(u)|}} + \mathbf{W}_2^k \mathbf{h}_v^{k-1}\right)$$

Non-linearity (e.g., ReLU or tanh)

trainable matrices (i.e., what we learn)

Previous layer embedding of node v

k-th layer embedding of node v

Average of neighbor's previous layer embeddings

$$\mathbf{z}_v = \mathbf{h}_v^k$$

Embedding after k layers of neighborhood aggregation.

Semi-supervised Classification with Graph Convolutional Network, ICLR, 2017.

- **Simple neighborhood aggregation:**

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_1^k \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{\sqrt{|N(u)|}} + \mathbf{W}_2^k \mathbf{h}_v^{k-1}\right)$$

- **GraphSAGE:**

$$\mathbf{h}_v^k = \sigma\left([\mathbf{W}_1^k \cdot \mathrm{AGG}\left(\{\mathbf{h}_u^{k-1}, \forall_u \in N(u)\}\right), \mathbf{W}_2^k \cdot \mathbf{h}_v^k]\right)$$

Generalized Aggregation: mean, pooling, LSTM

- **GAT:**

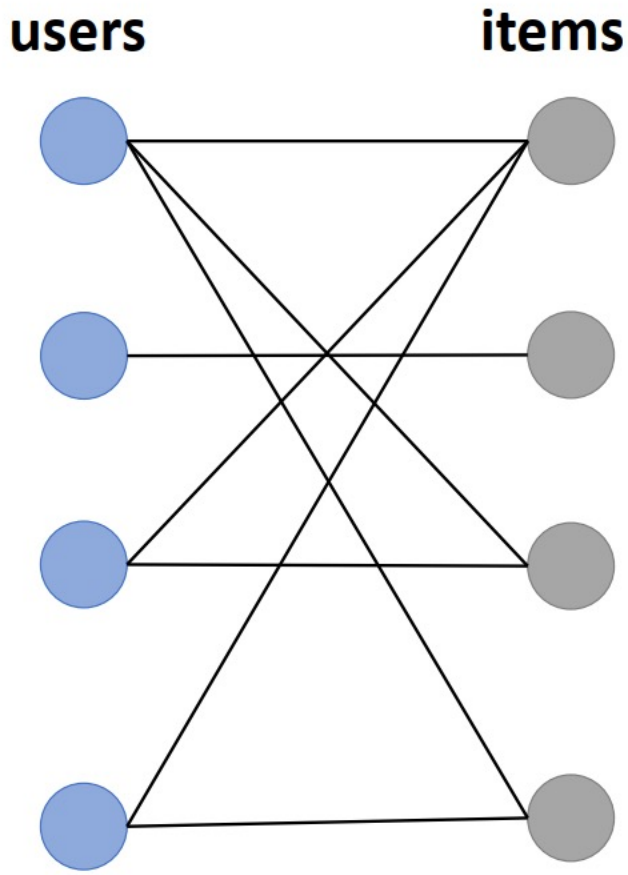
$$\mathbf{h}_v^k = \sigma\left(\sum_{u \in N(v)} \alpha_{v,u} \mathbf{W}^k \mathbf{h}_u^{k-1}\right)$$

Learned attention weights

Inductive Representation Learning on Large Graphs, NeuIPS, 2017.
Graph Attention Networks, ICLR, 2018

users \ items				
	1	0	1	1
	0	1	0	0
	1	1	0	0
	1	0	0	1

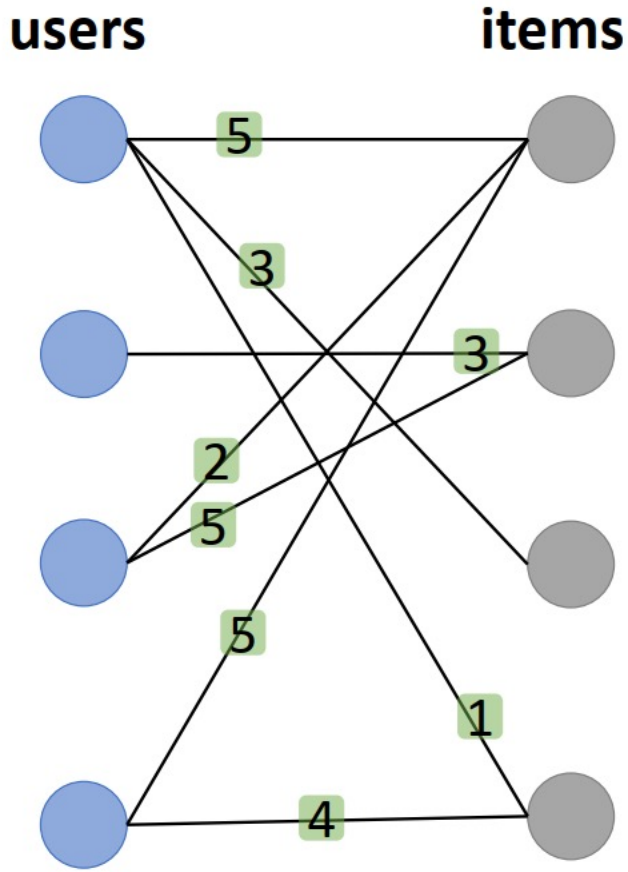
0/1 Interaction matrix



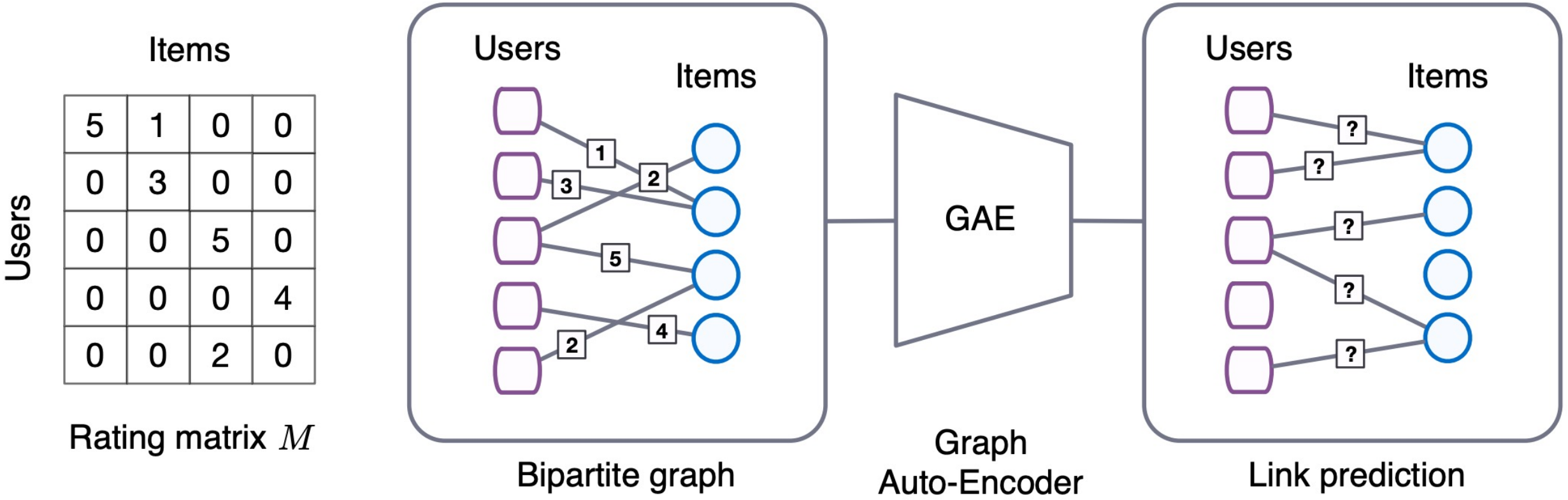
Bipartite Graph

users \ items				
	5	0	3	1
	0	3	0	0
	2	5	0	0
	5	0	0	4

Weighted interaction matrix

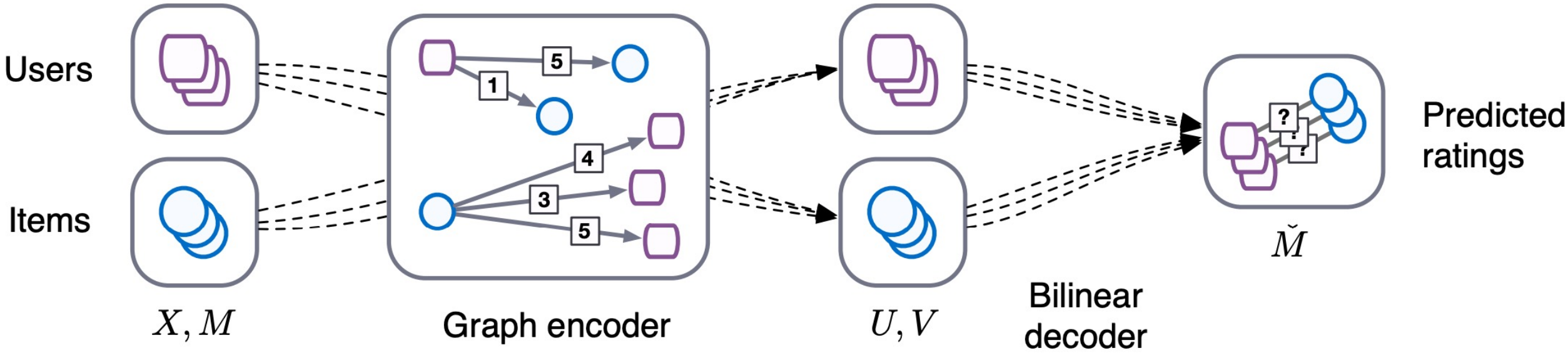


Bipartite Graph



部分边信息 → 节点表示学习 → 边信息预测

Graph Convolutional Matrix Completion. 2017.

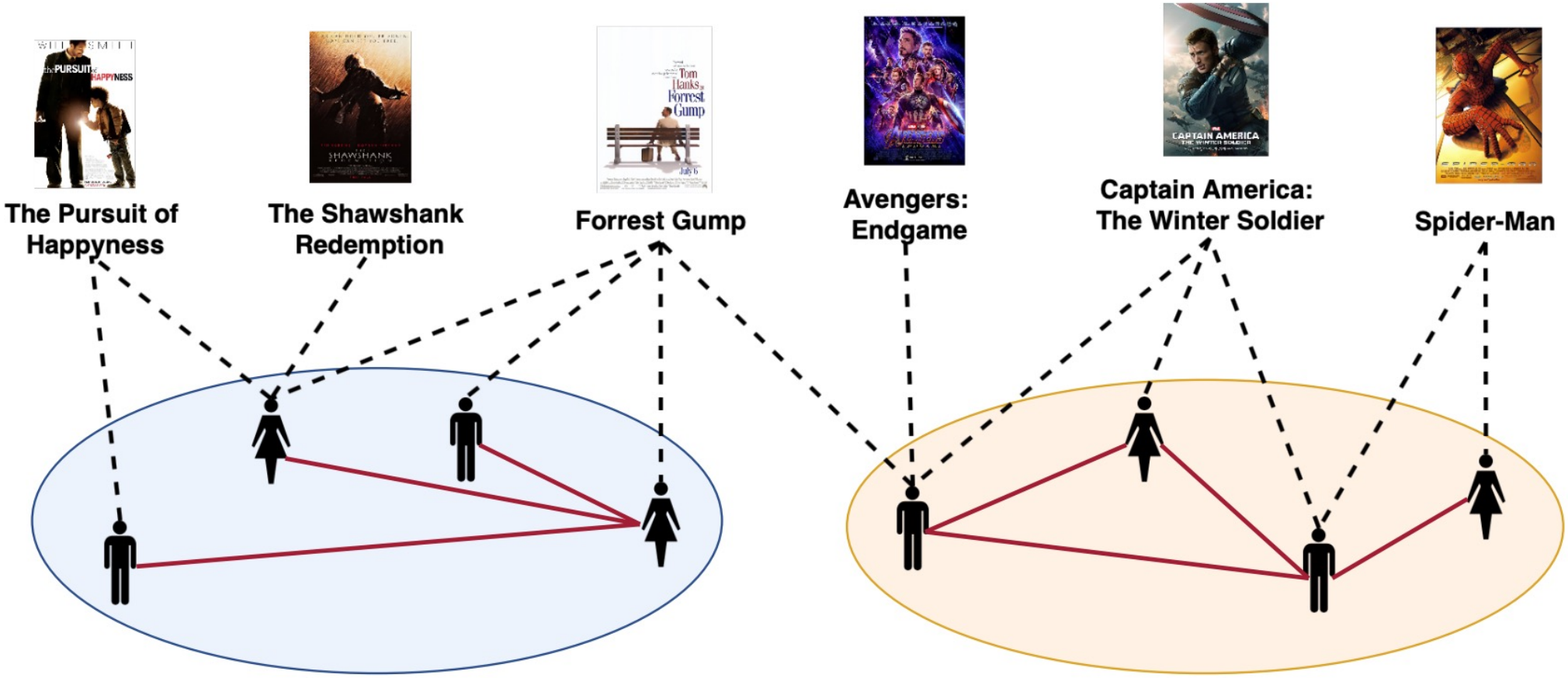


部分边信息 → 节点表示学习 → 边信息预测

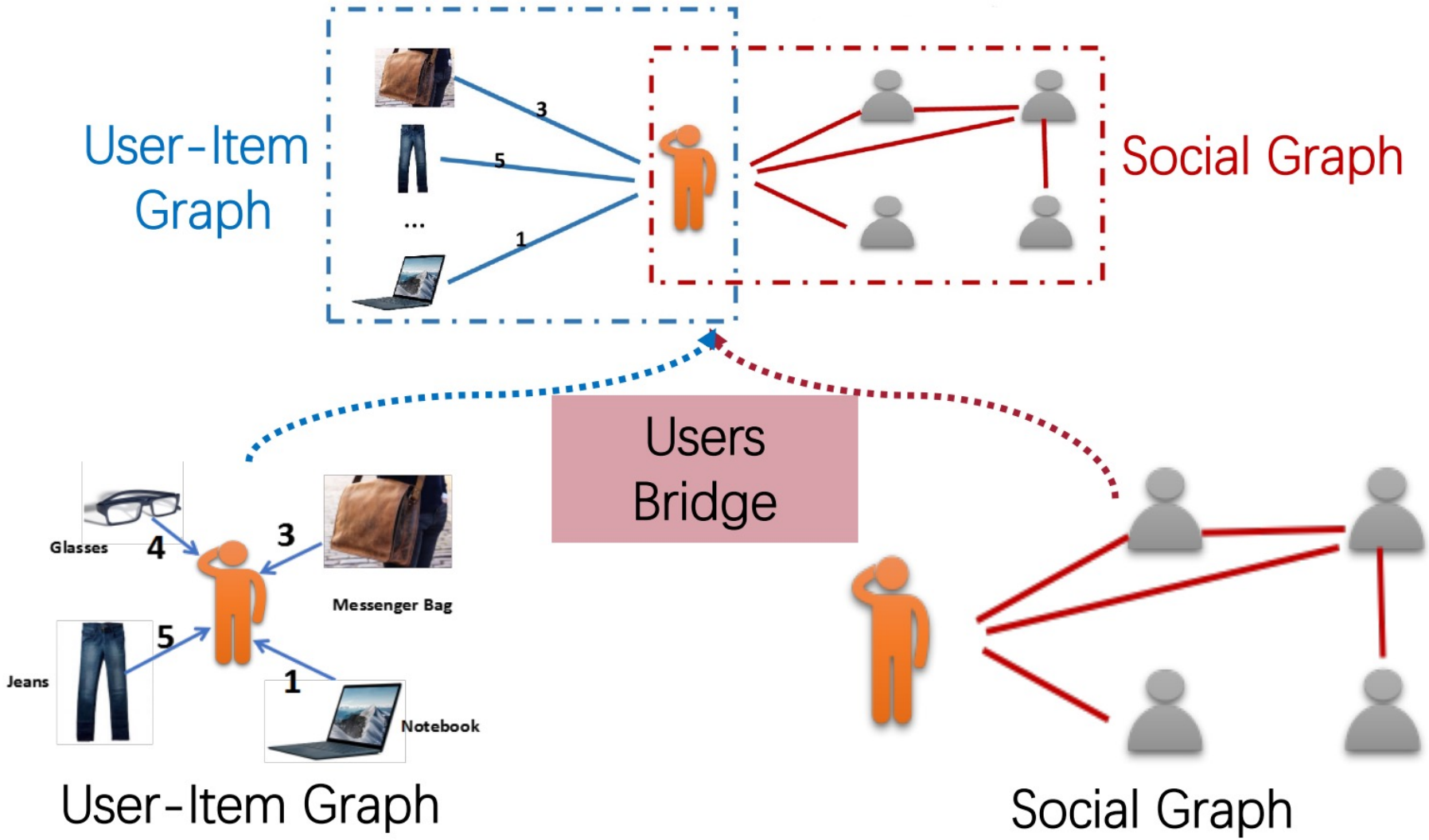
Graph Convolutional Matrix Completion. 2017.

增强用户辅助信息：社交网络

- Users' preferences are similar to or influenced by the people around them (nearer neighbours) [Tang et. al, 2013]



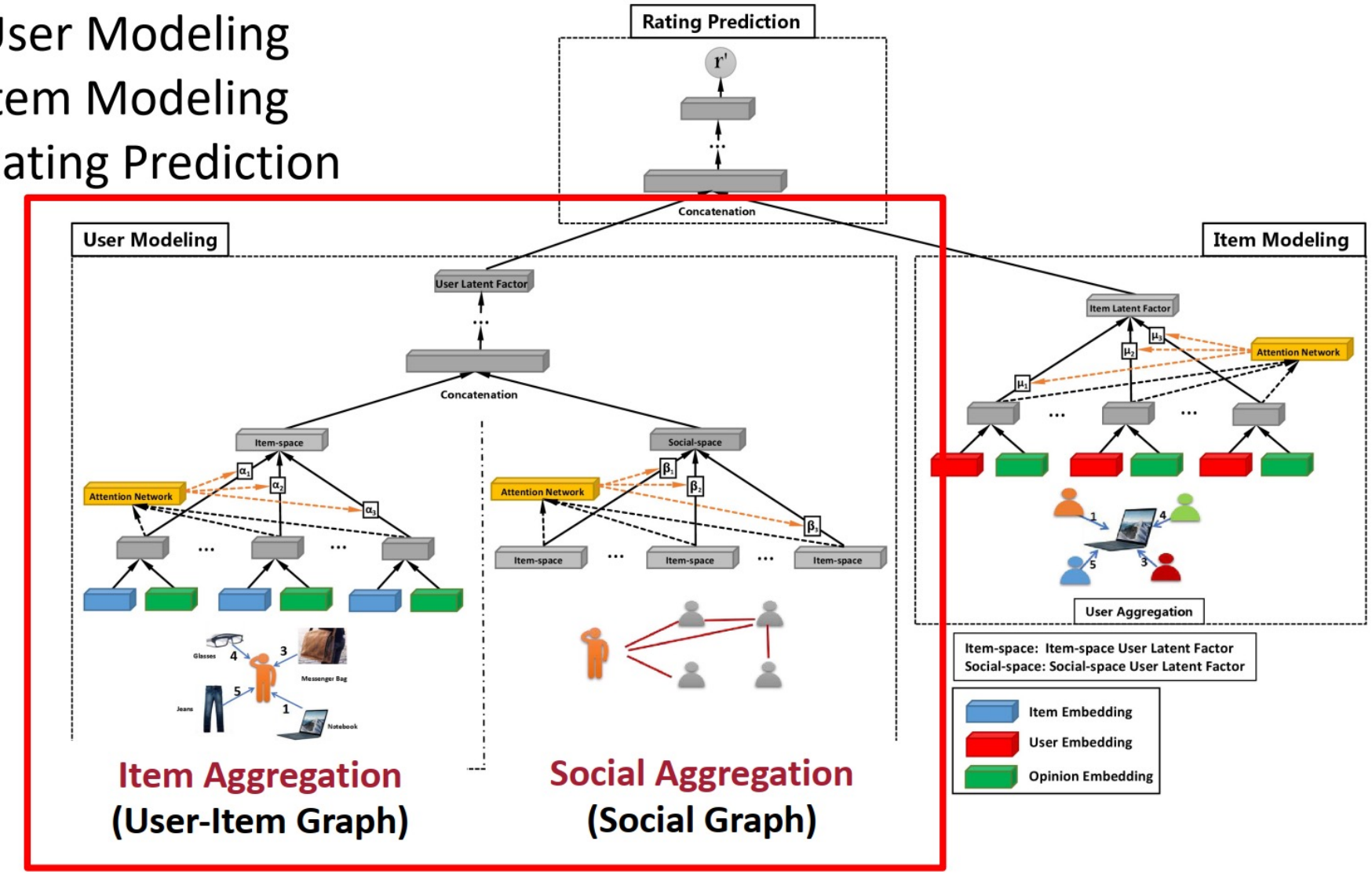
Graph Data in Social Recommendation



Graph Neural Networks for Social Recommendation. WWW 2019.

Three Components:

- User Modeling
- Item Modeling
- Rating Prediction



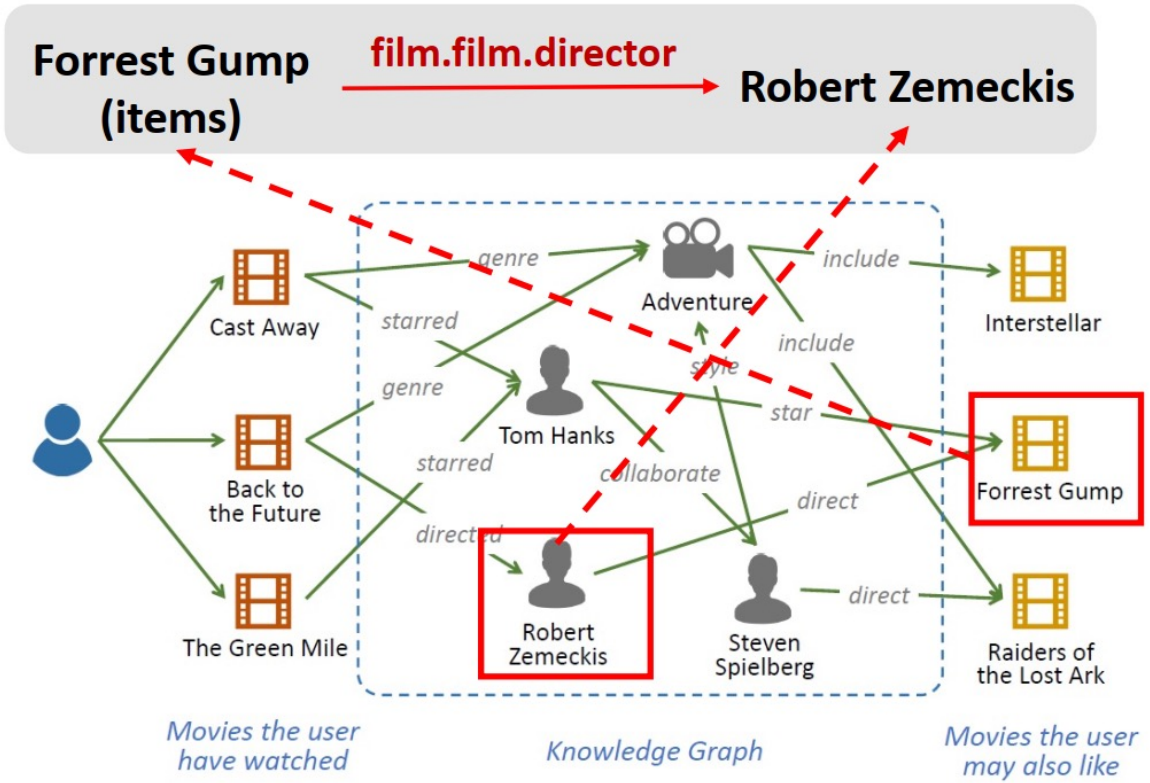
Graph Neural Networks for Social Recommendation. WWW 2019.

增强用户辅助信息：知识图谱（KG）

Heterogeneous Graph:

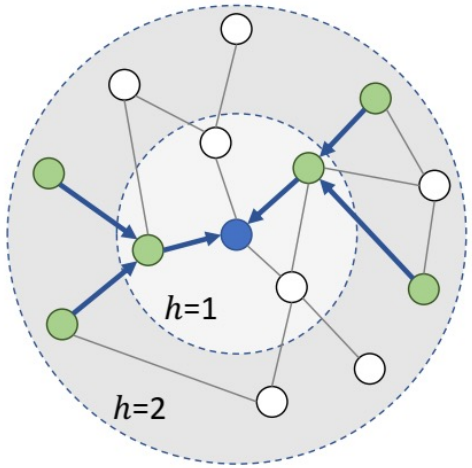
- Nodes: entities (Items)
- Edges: relations

Triples: (head, relation, tail)

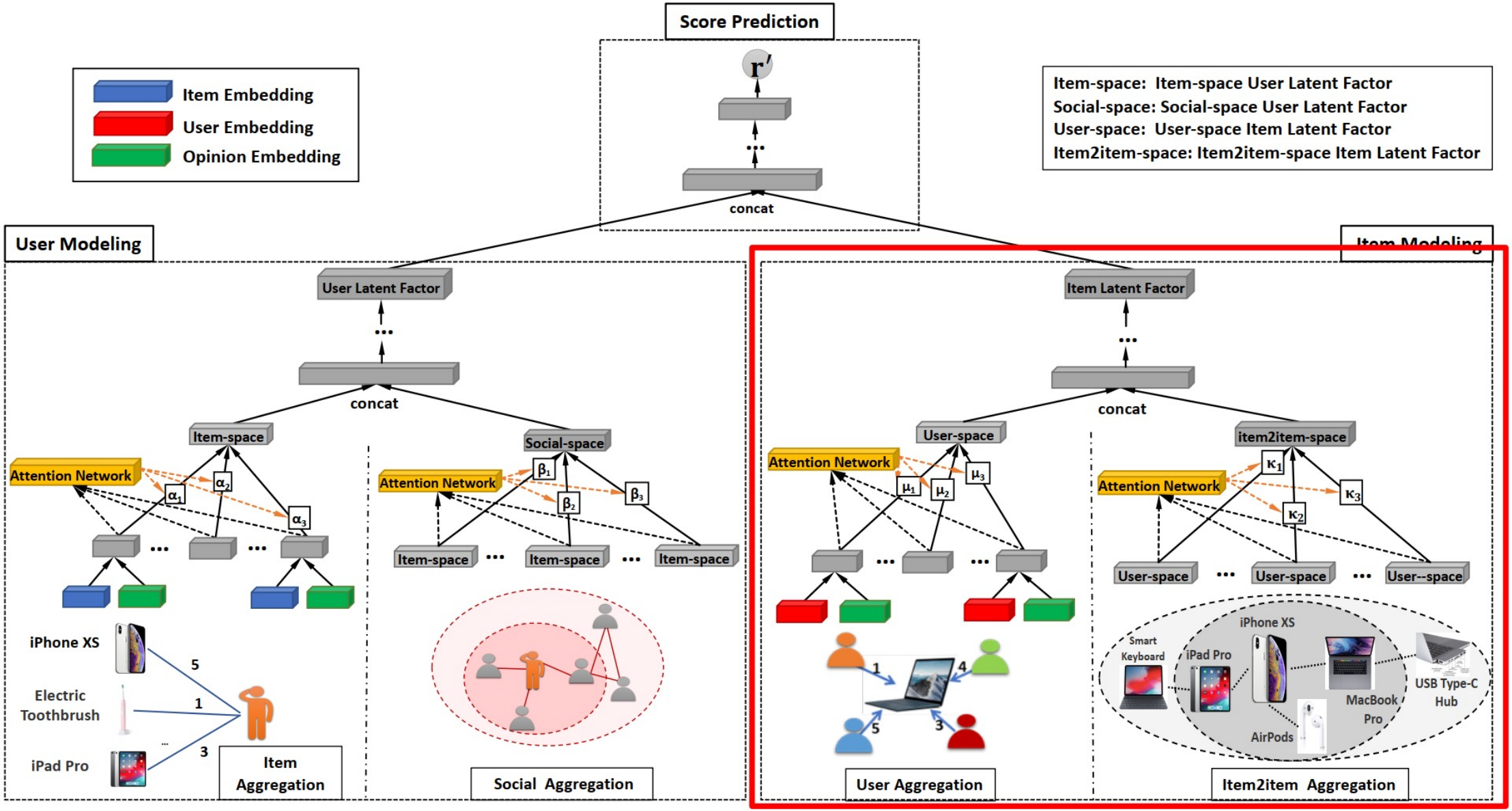


$$\hat{y}_{uv} = f(\mathbf{u}, \mathbf{v}^u)$$

GNNs?



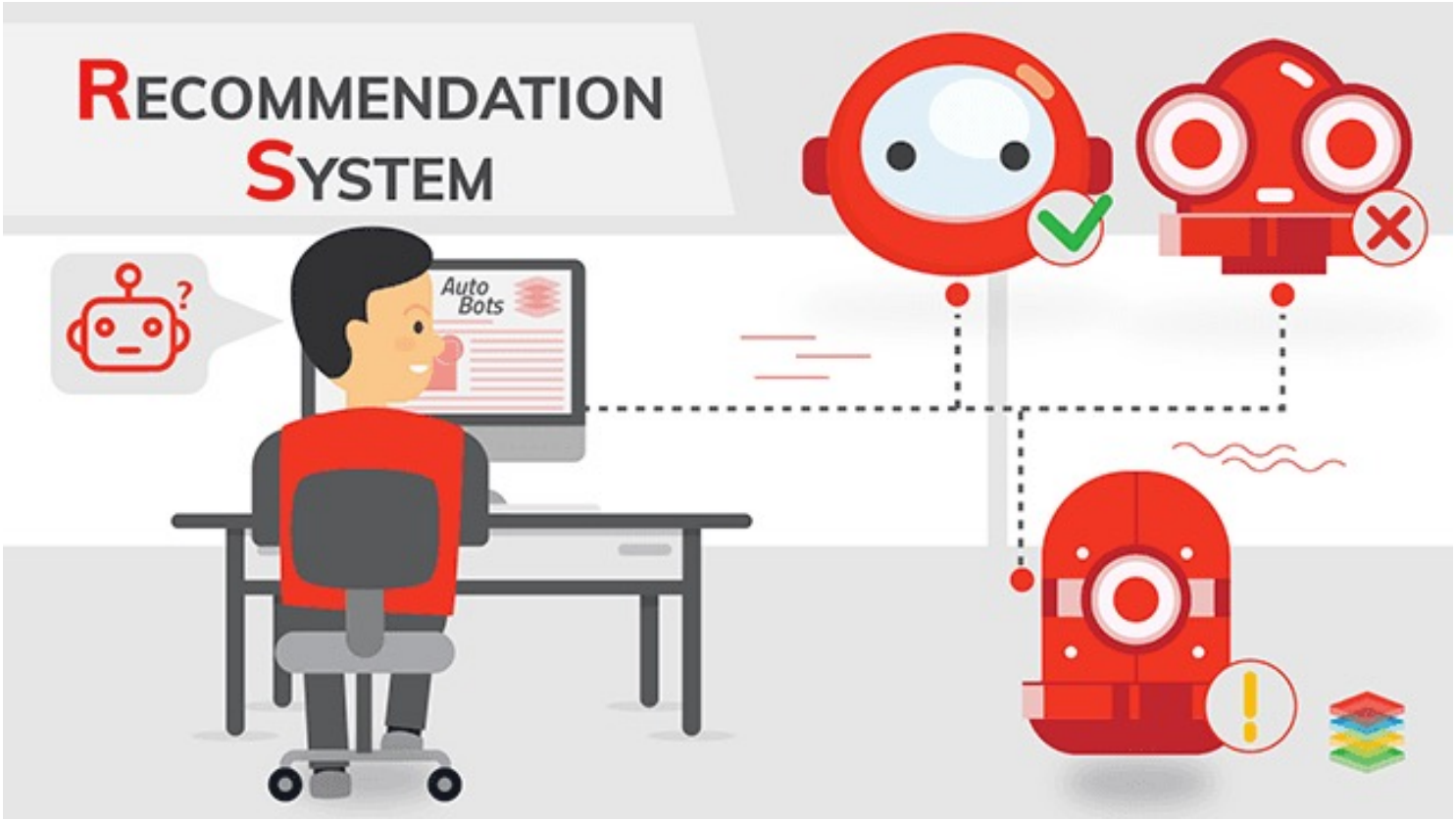
Knowledge Graph Convolutional Networks for Recommender Systems, WWW 2019.



A Graph Neural Network Framework for Social Recommendations, TKDE 2020
Graph Neural Networks for Social Recommendation, WWW 2019.

挑战：

- 多模态表示学习
- 动态异质图表示学习
- 决策行为（因果）学习



上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

人工智能研究院
Artificial Intelligence Institute

谢谢！

饮水思源 爱国荣校

http://humnetlab.berkeley.edu/~yxu/