Collaborative Filtering Methods in Recommendation System

Jihwan Lee

2022.03.18

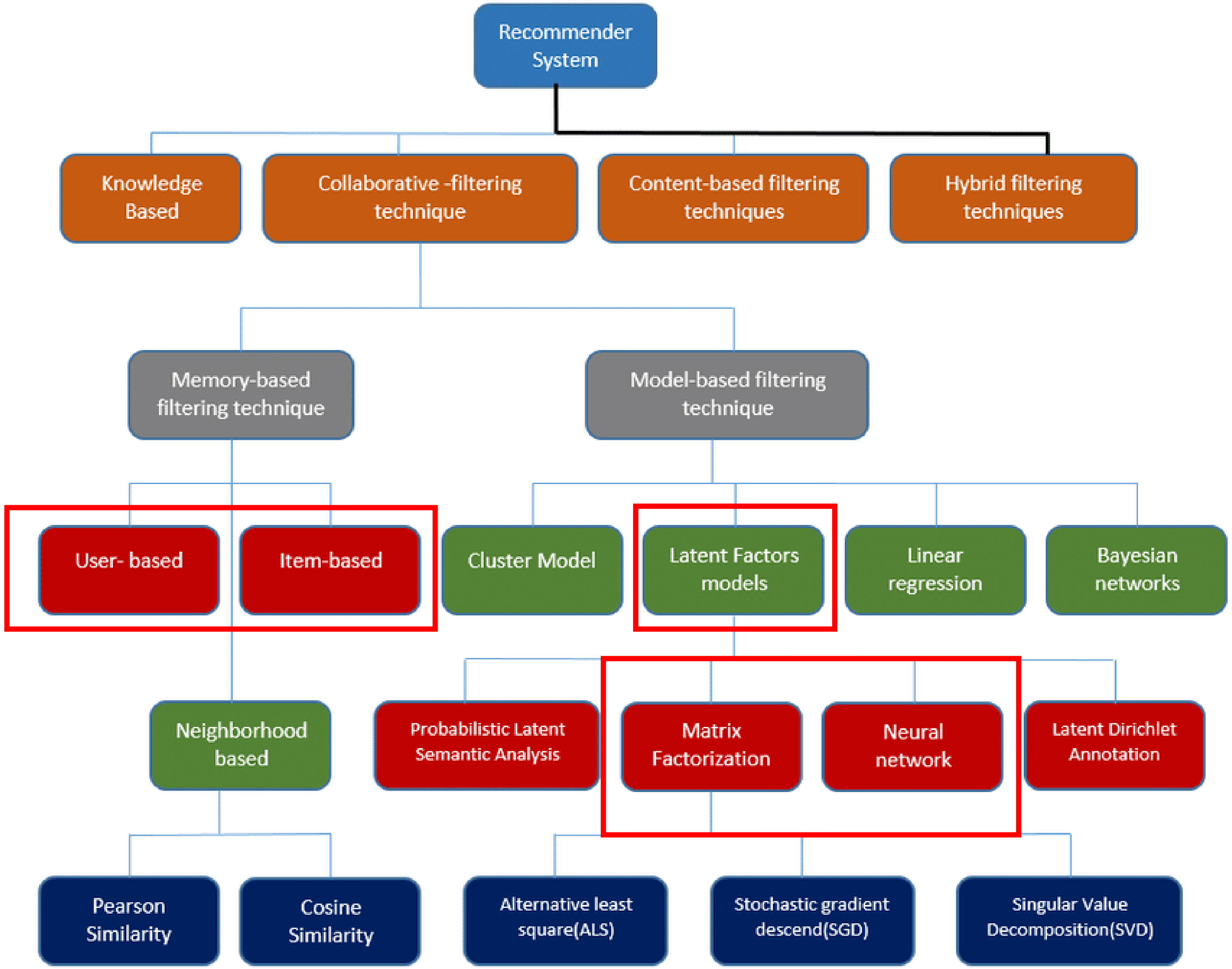
Contents

Recommender Systems
What is a recommendation system?

- A recommendation system refers to a system that shows information (movies, music, etc.) that a specific user may be interested in.
- Click prediction, Top-k recommendation, cold-start problem,….

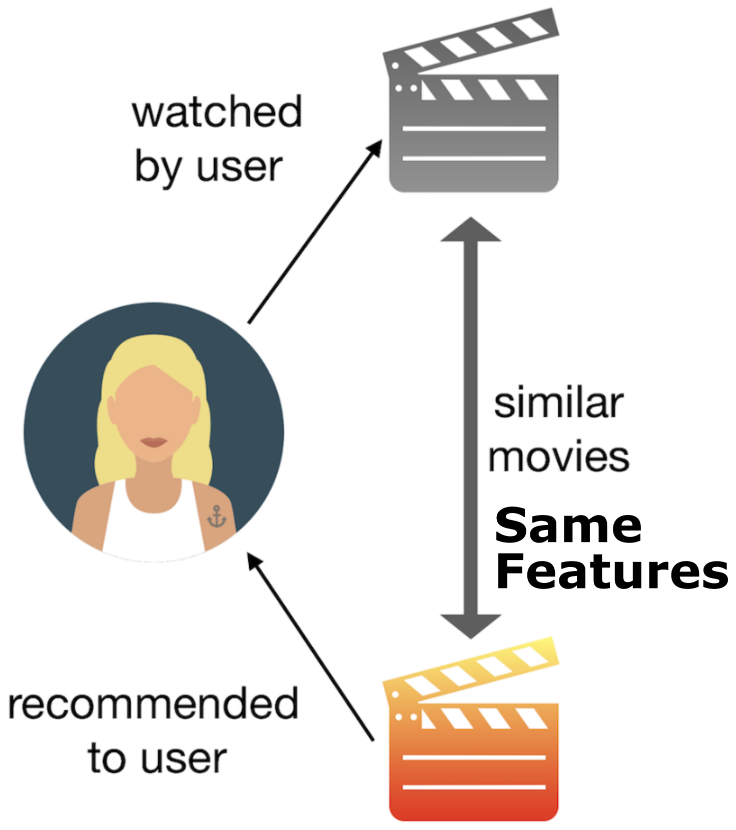
Recommender Systems

Taxonomy



Traditional Methods
Content-Based Filtering

- **Content-Based algorithms** aim to suggest items or product which are alike to that items that user enjoyed in past or is looking at in the present-day.
- **Advantages**
  [1] Other user's data not required.
  [2] No data sparsity as well as cold start.
- **Disadvantages**
  [1] Content analysis is essential to define the item features.
  [2] The excellence of the product can't be estimated.
  The likeness calculation is incomplete to the product description.



[fig 01] content-based filtering

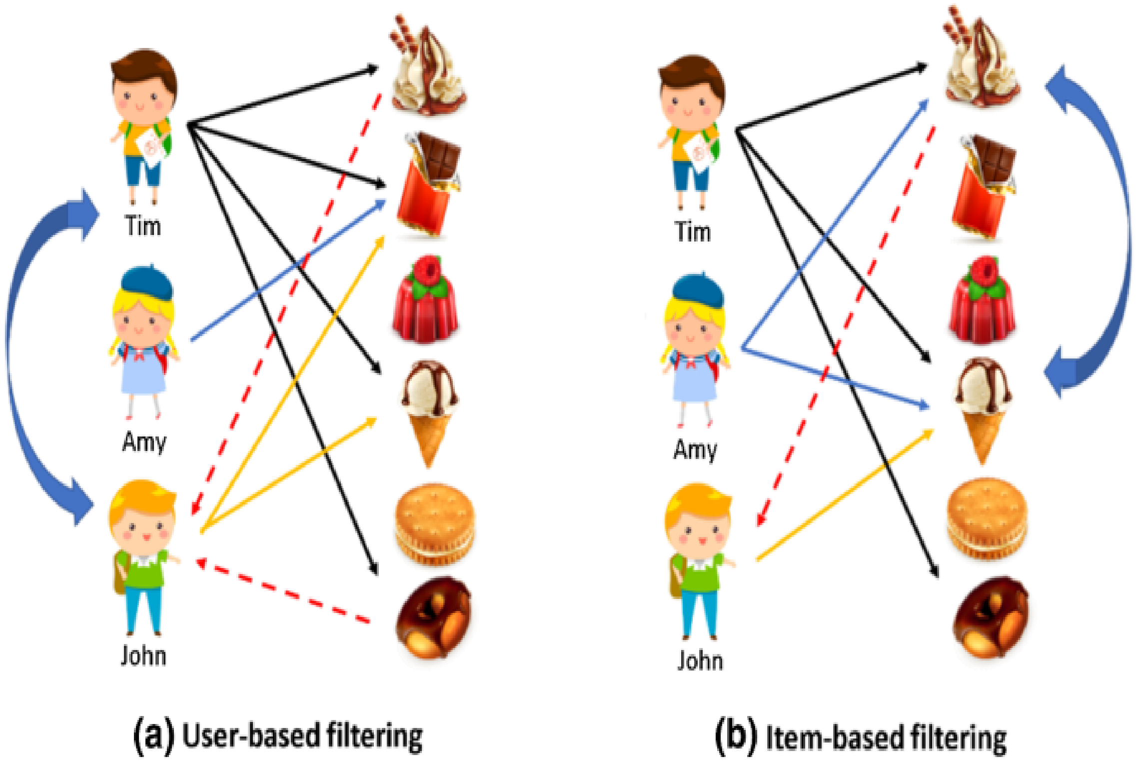
Das, Debashis, Laxman Sahoo, and Sujoy Datta. "A survey on recommendation system." *International Journal of Computer Applications* 160.7 (2017).

Traditional Methods

Memory Based Collaborative Filtering

- **Collaborative filtering** is based on the fact that relationships exist between products and people's interests.

- Collaborative filtering has two approaches:
  [1] User-Based Collaborative Filtering
  Based on user's neighborhood
  [2] Item-Based Collaborative Filtering
  Based on items' similarity



[fig 02] collaborative filtering

Das, Debashis, Laxman Sahoo, and Sujoy Datta. "A survey on recommendation system." *International Journal of Computer Applications* 160.7 (2017).

Traditional Methods

Memory Based Collaborative Filtering

User id	Item id	Rating
User 1	Item A	4
User 1	Item C	3
User 2	Item A	3
User 2	Item B	2
User 3	Item D	5

→

	Item A	Item B	Item C	Item 4
User 1	4		3	
User 2	3	3		
User 3				5

Das, Debashis, Laxman Sahoo, and Sujoy Datta. "A survey on recommendation system." *International Journal of Computer Applications* 160.7 (2017).

Traditional Methods

User-Based Collaborative Filtering

	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 = 0.70
sim = 0
sim = −0.79

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2}}$$

Pearson Correlation

Das, Debashis, Laxman Sahoo, and Sujoy Datta. "A survey on recommendation system." *International Journal of Computer Applications* 160.7 (2017).

Traditional Methods

Item-Based Collaborative Filtering

	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

$$similarity(i,j) = \frac{\sum_u^U r_{(u,i)} r_{(u,j)}}{\sqrt{\sum_u^U r_{(u,i)}^2}\sqrt{\sum_u^U r_{(u,j)}^2}}$$

Das, Debashis, Laxman Sahoo, and Sujoy Datta. "A survey on recommendation system." *International Journal of Computer Applications* 160.7 (2017).

Traditional Methods

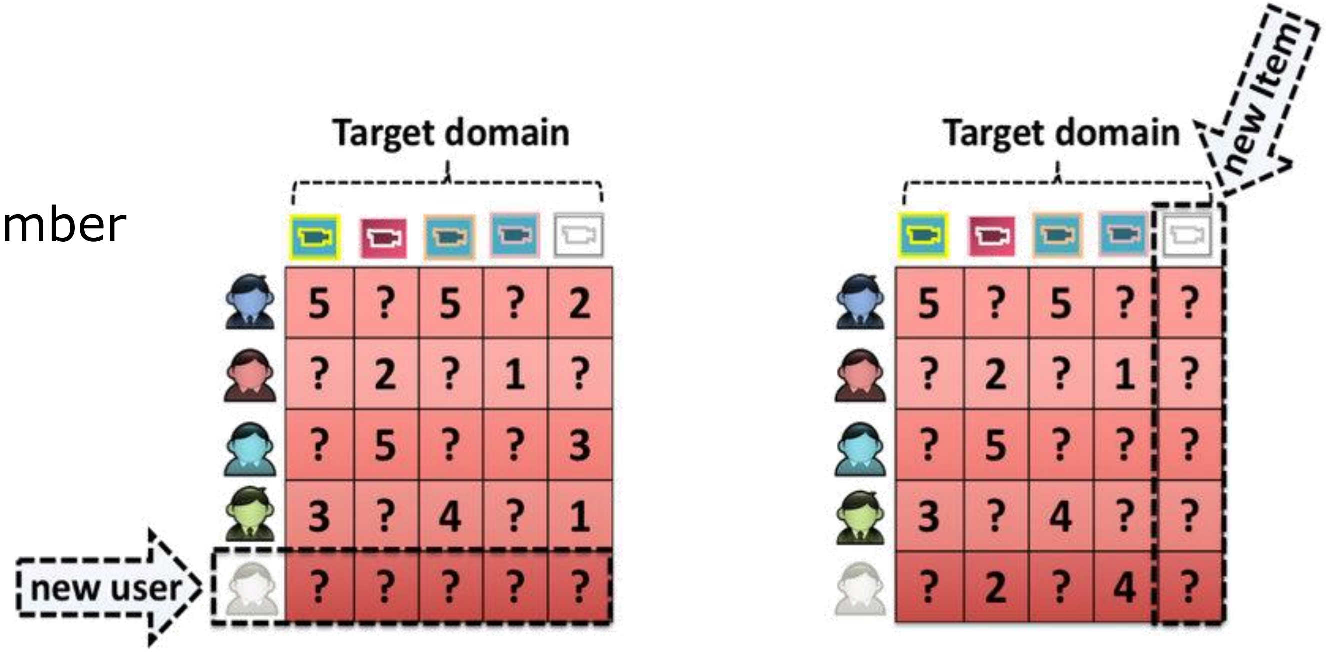
Problems of collaborative filtering

- **Data Sparsity**

  users in general rate only a limited a number of items
- **Cold Start**

  Difficulty in recommendation to new users or new items
- **Scalability**

  Increase in number of users of items



[fig 03] Cold Start Problem

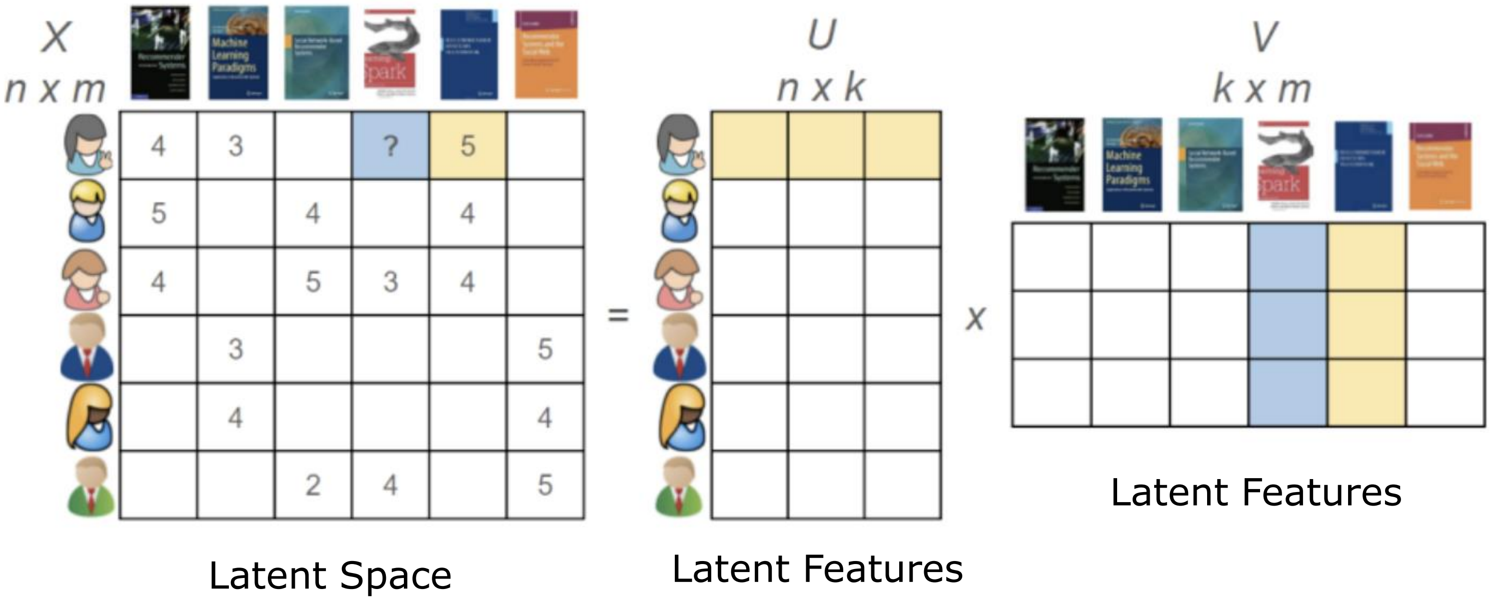
Das, Debashis, Laxman Sahoo, and Sujoy Datta. "A survey on recommendation system." *International Journal of Computer Applications* 160.7 (2017).

Traditional Methods

Collaborative Filtering: Matrix Factorization (Model-Based)

- Matrix Factorization (MF) algorithms work by decomposing the user-item interaction matrix into the product of two dimensionality rectangular matrices.

- This MF model enables the integration of additional information.



$$\hat{r}_{ui} = q_i^T p_u \qquad \min_{q^*, p^*} \sum_{(u,i) \in \kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\| q_i \|^2 + \| p_u \|^2)$$

Overfitting

Das, Debashis, Laxman Sahoo, and Sujoy Datta. "A survey on recommendation system." *International Journal of Computer Applications* 160.7 (2017).

Recommendation System
Problems of Matrix Factorization

- NCF model emphasizes the matrix factorization limitation caused by using an inner product.

- The most limitation of Matrix Factorization caused by the use of a simple and fixed inner product is that estimate complex user-item interactions in the low-dimensional latent space.

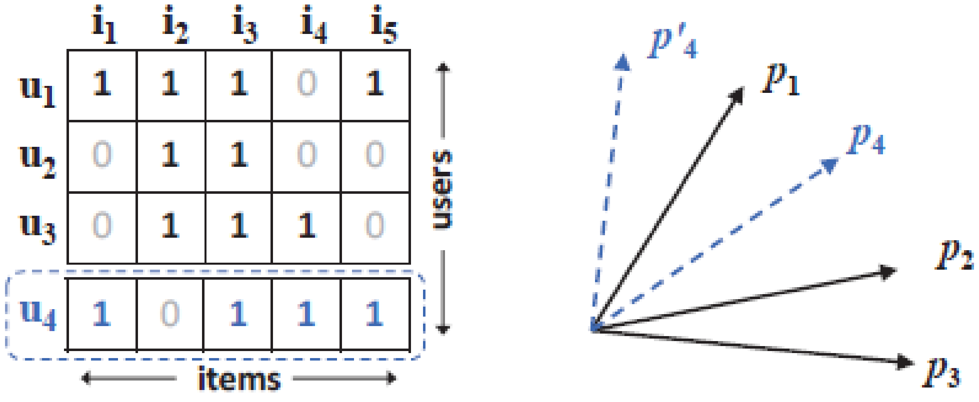
- Show the example (fig 04)

  new user u4 measure the similiarity with Jaccard coefficient (left).

  → s41(0.6) **>** s43(0.4) **>** s42(0.2)

  In the latent space (right), placing p4 closet to p1 makes p4 closer to p2 than p3, incurring a large ranking loss.

$$\hat{y}_{ui} = f(u, i|\mathbf{p}_u, \mathbf{q}_i) = \mathbf{p}_u^T \mathbf{q}_i = \sum_{k=1}^{K} p_{uk} q_{ik}$$



[fig 04]Limitation of matrix factorization

- **Jaccard coefficient similarity**
  Let Ru be the set of items that user u has interacted with, then the Jaccard similarity of users i and j is defined as

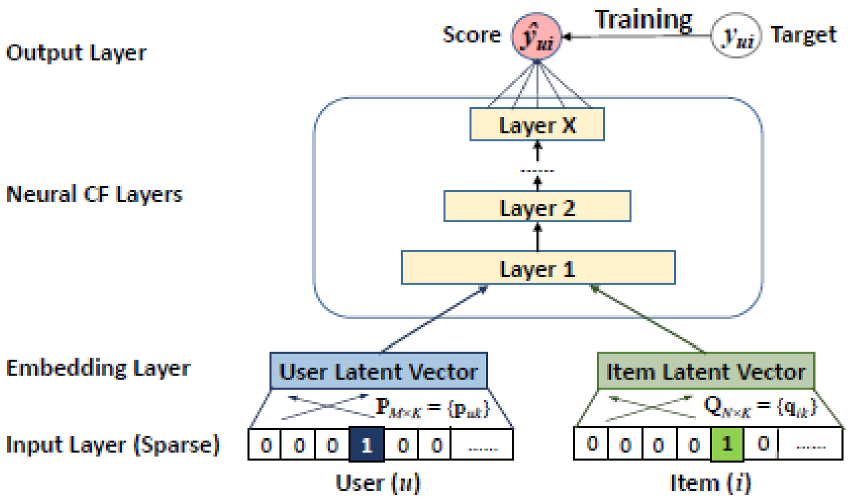
$$s_{ij} = \frac{|\mathcal{R}_i| \cap |\mathcal{R}_j|}{|\mathcal{R}_i| \cup |\mathcal{R}_j|}$$

He, Xiangnan, et al. "Neural collaborative filtering." *Proceedings of the 26th international conference on world wide web*. 2017.

Recommendation System

Deep Learning Based Methods - MLP

- **Neural Collaborative Filtering (NCF)**
- NCF explore the central theme of how to utilize DNNs to model noisy implicit feedback signals.
- Binary classification: interaction → 1, no interaction → 0
- Paper proposes three NCF methods
  [1] Generalized Matrix Factorization (GMF)
  [2] Multi-Layer Perceptron (MLP)
  [3] Ensemble of GMP and MLP (NeuMF)

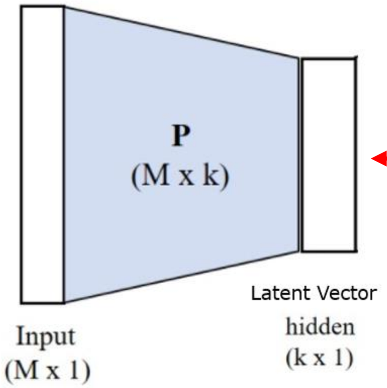


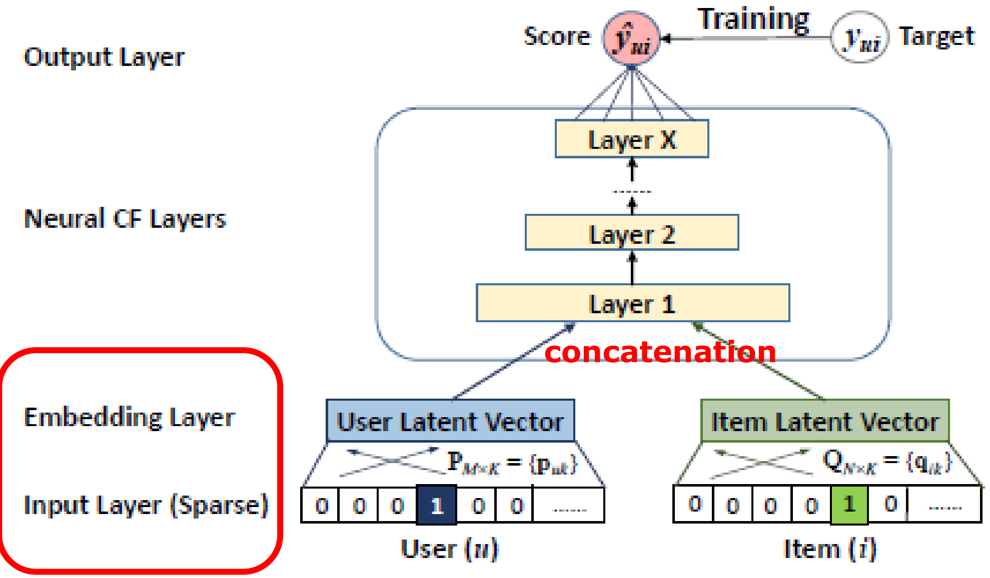
[fig 05] NCF framework

He, Xiangnan, et al. "Neural collaborative filtering." *Proceedings of the 26th international conference on world wide web*. 2017.

Recommendation System
Deep Learning Based Methods - MLP

- **Neural Collaborative Filtering (NCF)**
  - Input Layer: user (or item) one hot-encoding
  - Embedding Layer: Input Sparse Layer Mapping to K(<m) layer → User (or Item) Latent Vector
  - Neural CF Layers: Fully-Connected Layer





[fig 06] NCF framework

  - Output Layers

$$\hat{y}_{u,i} = f(P^T v_u^U, Q^T v_i^I | P, Q, \Theta_f) = \phi_{out}(\phi_X(\ldots\phi_2(\phi_1(P^T v_u^U, Q^T v_i^I))\ldots)), \quad 0 \leq \hat{y}_{u,i} \leq 1$$

→ Predicted value means how user u and item I interact on each other.

He, Xiangnan, et al. "Neural collaborative filtering." *Proceedings of the 26th international conference on world wide web*. 2017.

Recommendation System

Deep Learning Based Methods - MLP

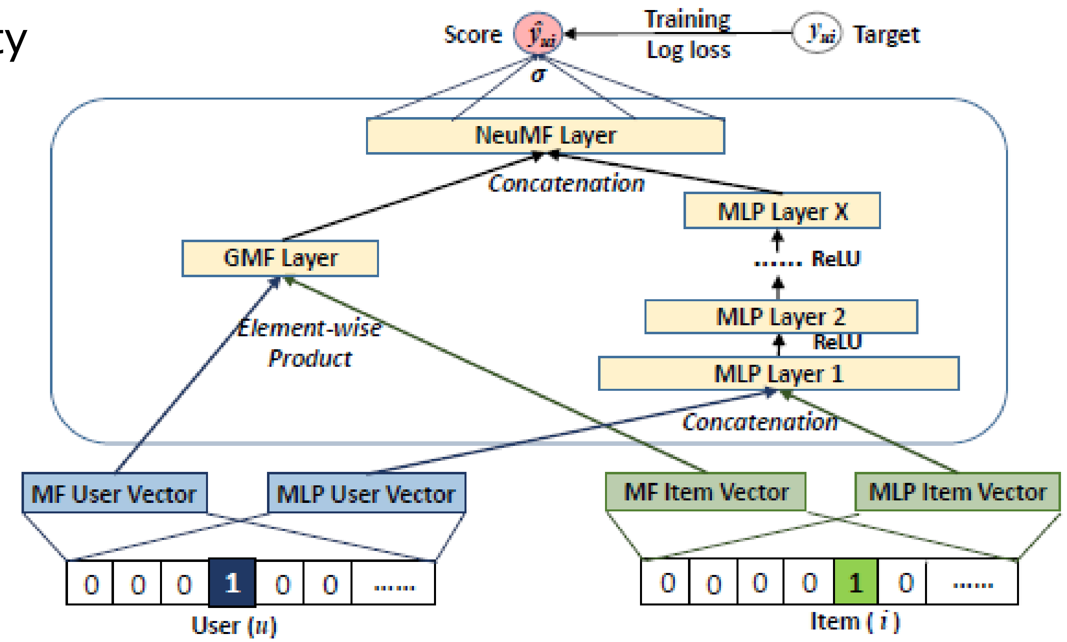
- **Neural Matrix Factorization: GMP + MLP**

- It unifies the strengths of linearity of MF and non-linearity of MLP for modeling the user-item latent structures.

$$\phi^{GMF} = \mathbf{p}_u^G \odot \mathbf{q}_i^G,$$

$$\phi^{MLP} = a_L(\mathbf{W}_L^T(a_{L-1}(...a_2(\mathbf{W}_2^T \begin{bmatrix} \mathbf{p}_u^M \\ \mathbf{q}_i^M \end{bmatrix} + \mathbf{b}_2)...)) + \mathbf{b}_L),$$

$$\hat{y}_{ui} = \sigma(\mathbf{h}^T \begin{bmatrix} \phi^{GMF} \\ \phi^{MLP} \end{bmatrix}),$$



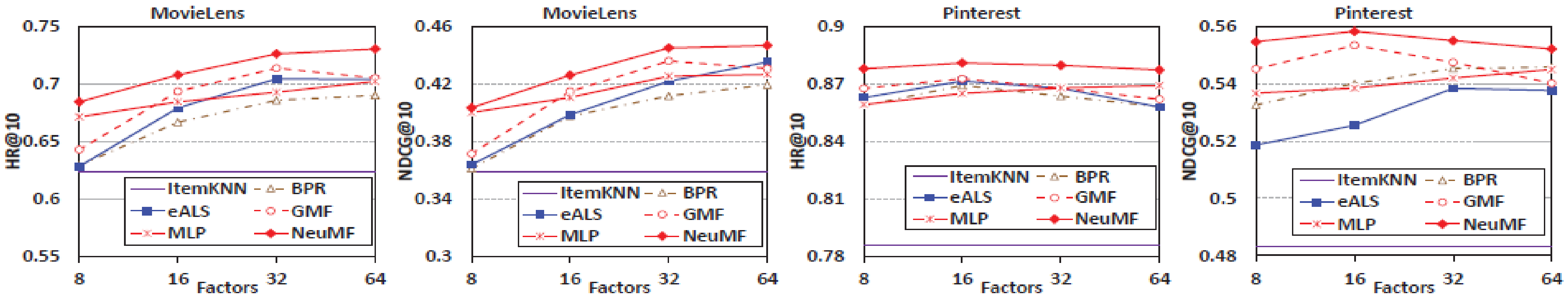
[fig 07] NeuMF Framework

He, Xiangnan, et al. "Neural collaborative filtering." *Proceedings of the 26th international conference on world wide web*. 2017.

Recommendation System

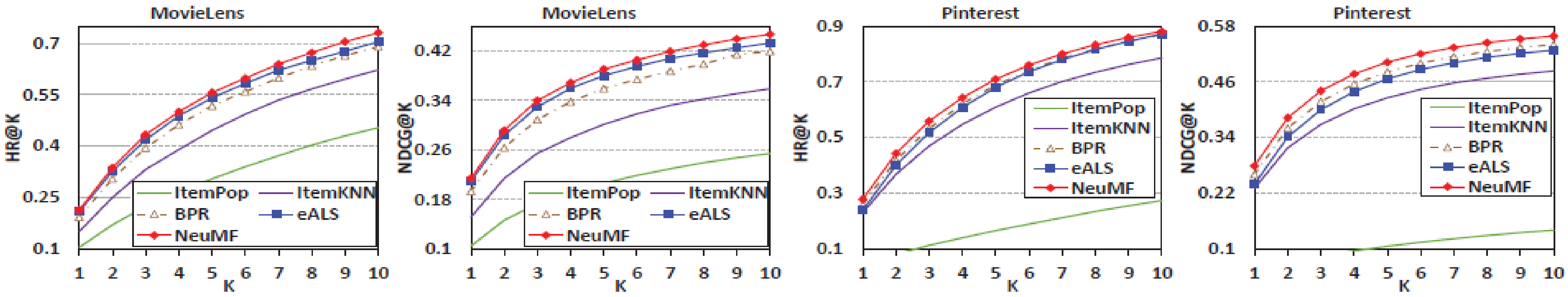
Deep Learning Based Methods - MLP

- **Experiments**



(a) MovieLens — HR@10 (b) MovieLens — NDCG@10 (c) Pinterest — HR@10 (d) Pinterest — NDCG@10

Figure 4: Performance of HR@10 and NDCG@10 *w.r.t.* the number of predictive factors on the two datasets.



(a) MovieLens — HR@K (b) MovieLens — NDCG@K (c) Pinterest — HR@K (d) Pinterest — NDCG@K

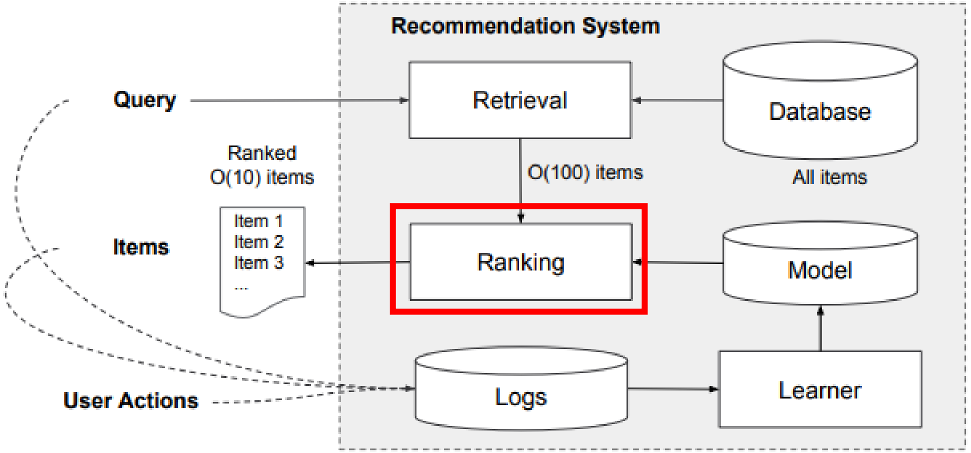
Figure 5: Evaluation of Top-K item recommendation where K ranges from 1 to 10 on the two datasets.

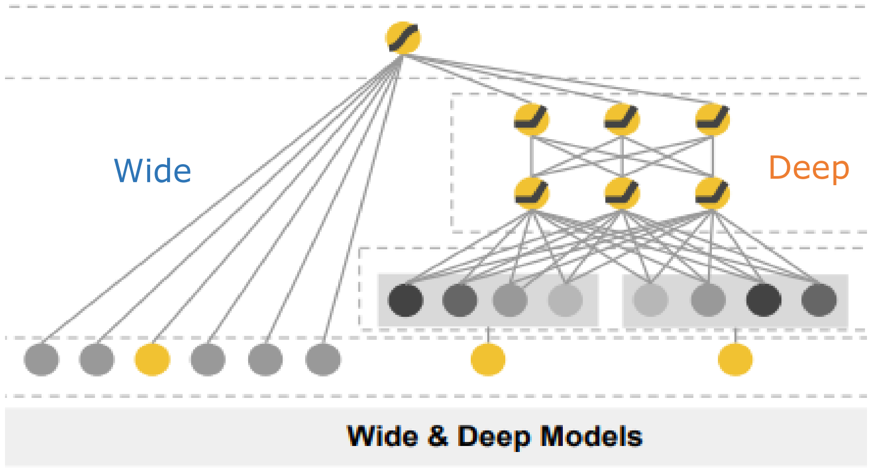
Recommendation System

Deep Learning Based Methods - MLP

- **Wide & Deep Learning**

- It was initially introduced for an app recommendation for Google play.
- Memorization: It can achieve through wide learning component (linear wide model)
- Generalization: Deep learning model can catch generalization by producing more general and abstract representations. (non-linear deep learning model) → That is wide and deep model.



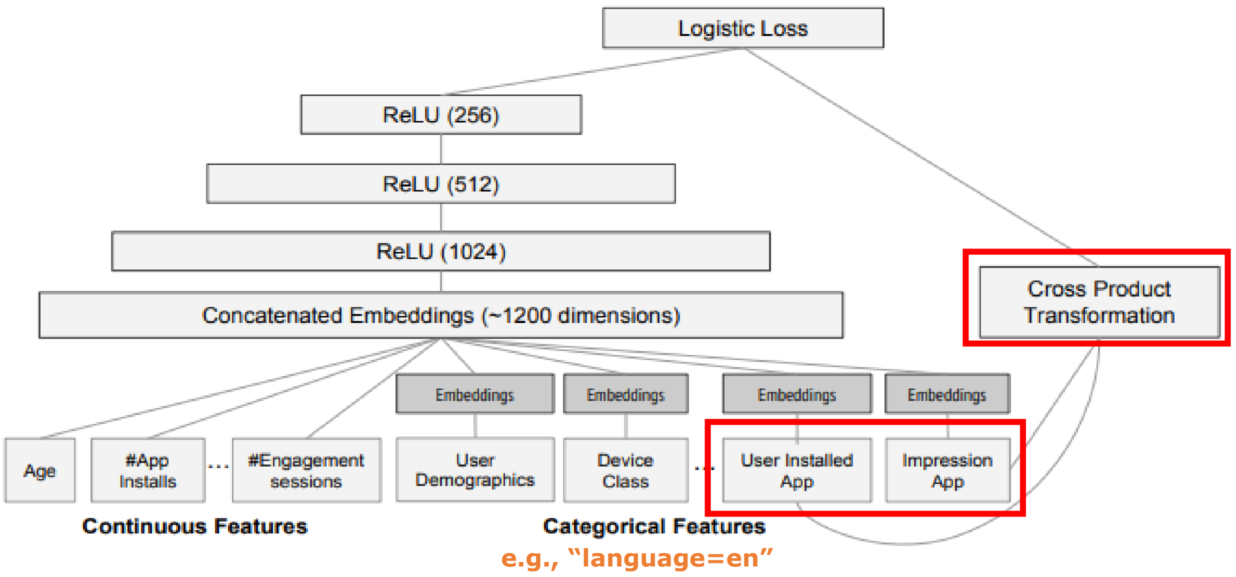


[fig 08] App pipeline & Wide Deep Models Architecture

Cheng, Heng-Tze, et al. "Wide & deep learning for recommender systems." *Proceedings of the 1st workshop on deep learning for recommender systems*. 2016.

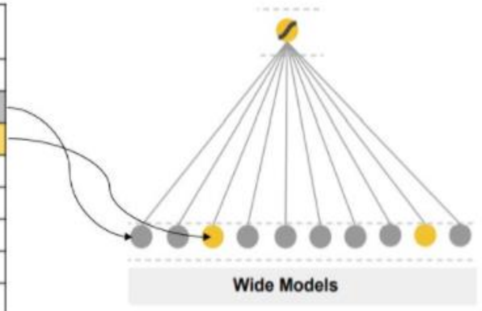
Recommendation System
Deep Learning Based Methods - MLP

- **Wide & Deep Learning – Wide component**



[fig 09] Wide learning components

Install	Impression	(Install, Impression)	Install x Impression
A	A	(1,1)	1
A	B	(1,0)	0
A	C	(1,1)	1
B	A	(1,1)	1
B	B	(1,0)	0
B	C	(1,1)	1
C	A	(0,1)	0
C	B	(0,0)	0
C	C	(0,1)	0



$$y = \mathbf{w}^T \mathbf{x} + b$$

$$\begin{aligned} where, \quad \mathbf{x} &= [x_1, x_2, \ldots, x_d] \\ \mathbf{w} &= [w_1, w_2, \ldots, w_d] \\ b &= \text{bias} \end{aligned}$$

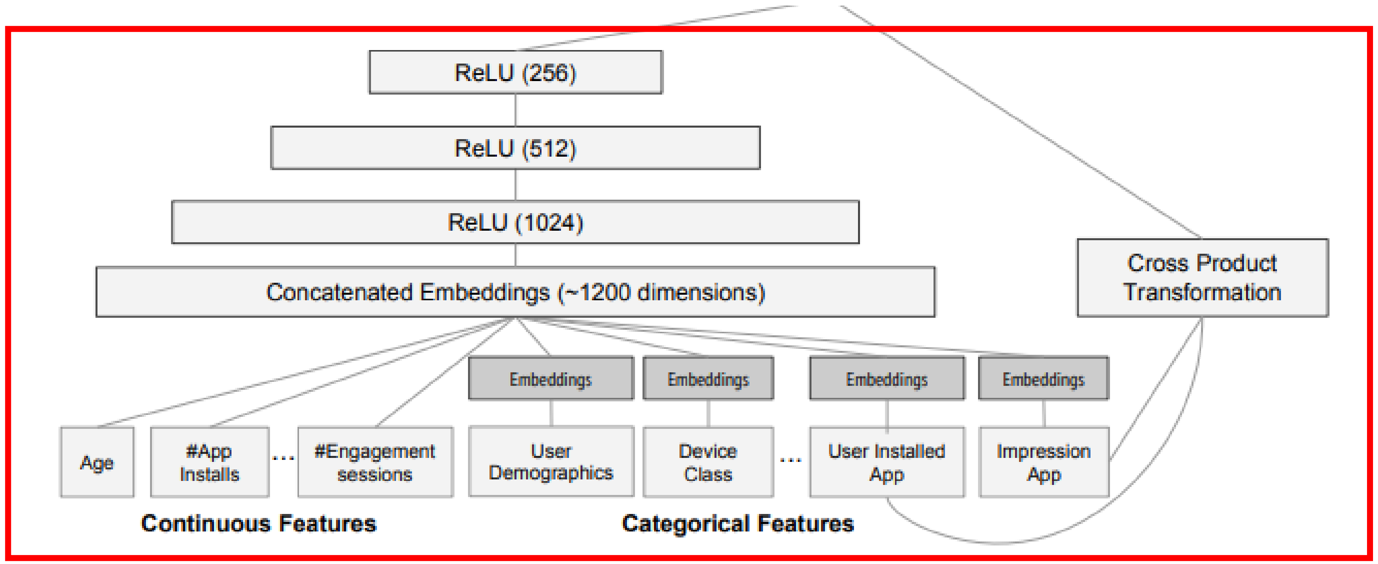
Cheng, Heng-Tze, et al. "Wide & deep learning for recommender systems." *Proceedings of the 1st workshop on deep learning for recommender systems*. 2016.

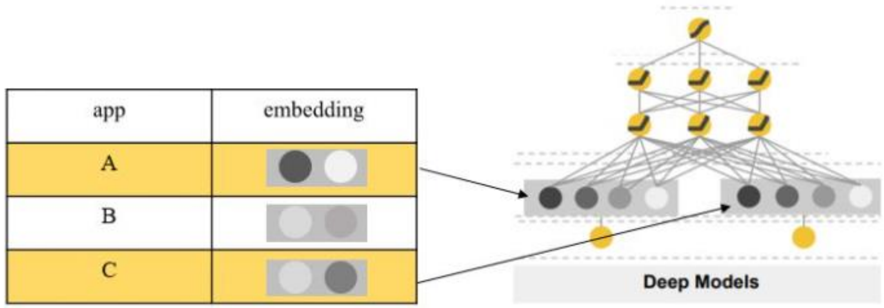
Recommendation System

Deep Learning Based Methods - MLP

- **Wide & Deep Learning – Deep component**

$$p(Y=1|\mathbf{x}) = \sigma(\mathbf{w}_{wide}^T[\mathbf{x}] + \mathbf{w}_{deep}^T a^{(l_f)} + b)$$





[fig 10] Wide component

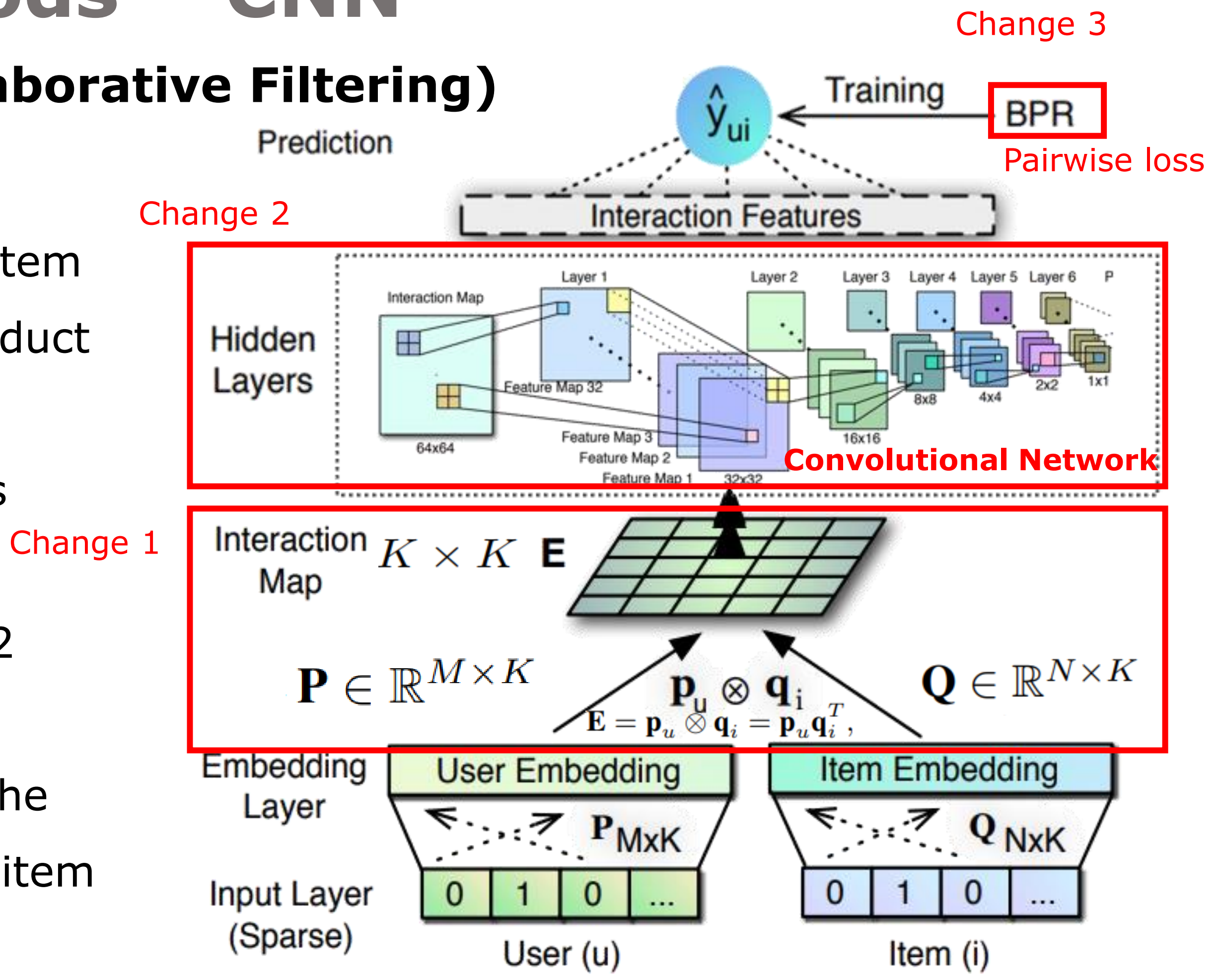
Cheng, Heng-Tze, et al. "Wide & deep learning for recommender systems." *Proceedings of the 1st workshop on deep learning for recommender systems*. 2016.

Recommendation System

Deep Learning Based Methods – CNN

- **ONCF (Outer Product-based Neural Collaborative Filtering)**

  - This model is also called ConvNCF.
  - NeuMF conducts concatenation two vectors(user&item latent vector), but ConvNCF proposes to outer product operation to obtain the interaction map.
  - Hidden layer: Convolutional network with 6 layers → 64X64X1 feature map change to 1X1X32.
  - ConvNCF applies a linear projection on the 1X1X32 tensor to obtain the prediction. $\hat{y}_{ui} = \mathbf{w}^T \mathbf{g}$
  - ConvNCF is the first work that uses CNN to learn the interaction function between user embedding and item embedding.



[fig 11] ONCF (ConvNCF) Framework

He, Xiangnan, et al. "Outer product-based neural collaborative filtering." *arXiv preprint arXiv:1808.03912* (2018).

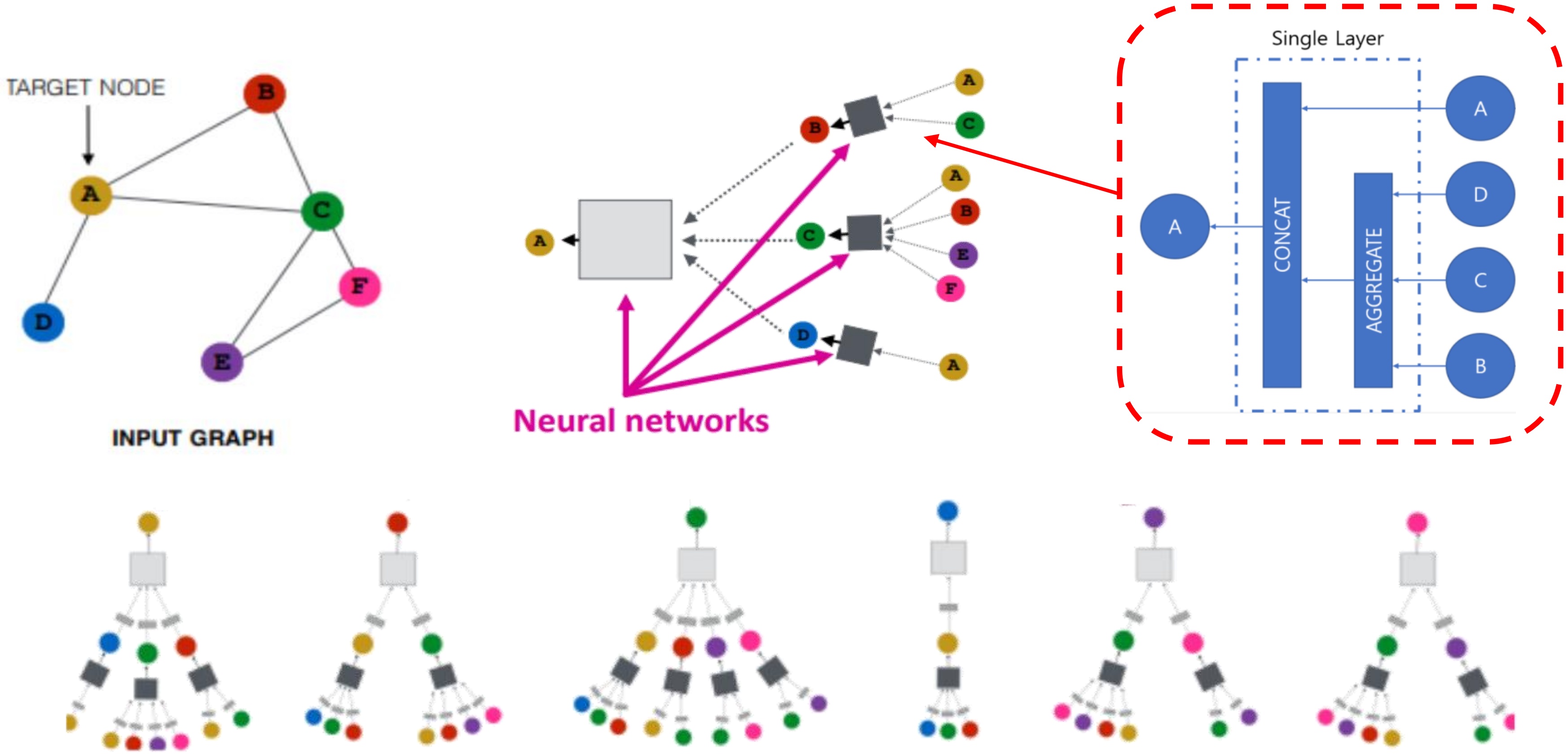
Recommendation System
Deep Learning Based Methods – CNN

- **ONCF Experiments**

	Gowalla						Yelp						
	HR@k			**NDCG@k**			**HR@k**			**NDCG@k**			RI
	$k=5$	$k=10$	$k=20$	$k=5$	$k=10$	$k=20$	$k=5$	$k=10$	$k=20$	$k=5$	$k=10$	$k=20$	
ItemPop	0.2003	0.2785	0.3739	0.1099	0.1350	0.1591	0.0710	0.1147	0.1732	0.0365	0.0505	0.0652	+227.6%
MF-BPR	0.6284	0.7480	0.8422	0.4825	0.5214	0.5454	0.1752	0.2817	0.4203	0.1104	0.1447	0.1796	+9.5%
MLP	0.6359	0.7590	0.8535	0.4802	0.5202	0.5443	0.1766	0.2831	0.4203	0.1103	0.1446	0.1792	+9.2%
JRL	0.6685	0.7747	0.8561	0.5270	0.5615	0.5821	0.1858	0.2922	0.4343	0.1177	0.1519	0.1877	+3.9%
NeuMF	0.6744	0.7793	0.8602	0.5319	0.5660	0.5865	0.1881	0.2958	0.4385	0.1189	0.1536	0.1895	+3.0%
ConvNCF	**0.6914***	**0.7936***	**0.8695***	**0.5494***	**0.5826***	**0.6019***	**0.1978***	**0.3086***	**0.4430***	**0.1243***	**0.1600***	**0.1939***	-

He, Xiangnan, et al. "Outer product-based neural collaborative filtering." *arXiv preprint arXiv:1808.03912* (2018).

Graph Neural Network

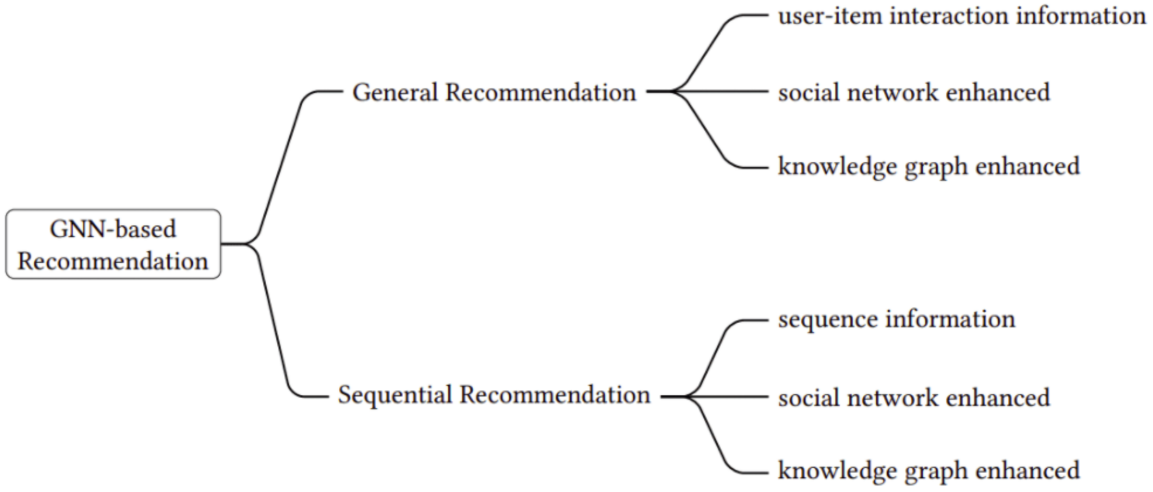


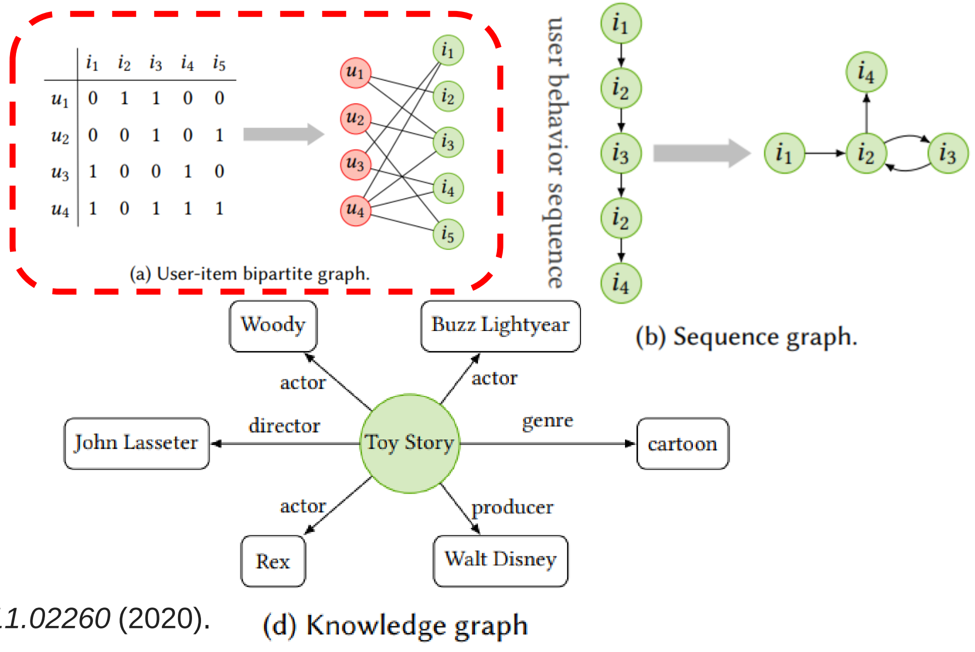
- Each node is characterized by having a different neural network structure.
- Node Aggregation methods are various. (e.g. Mean-pooling, Normalization,…)

Wu, Shiwen, et al. "Graph neural networks in recommender systems: a survey." *arXiv preprint arXiv:2011.02260* (2020).

Graph Neural Networks for Recommendations

- Graph neural networks can be used to model feature interactions and generate high-quality embeddings for all users and items.
- GNNs can be used for different types of recommendations. General recommendations disregard the notion of time and deals with user-item interactions but sequential recommendation seeks to capture transitional patterns in the user's behavior.
- The knowledge graph can explain why user was recommended the item. (explainable AI)



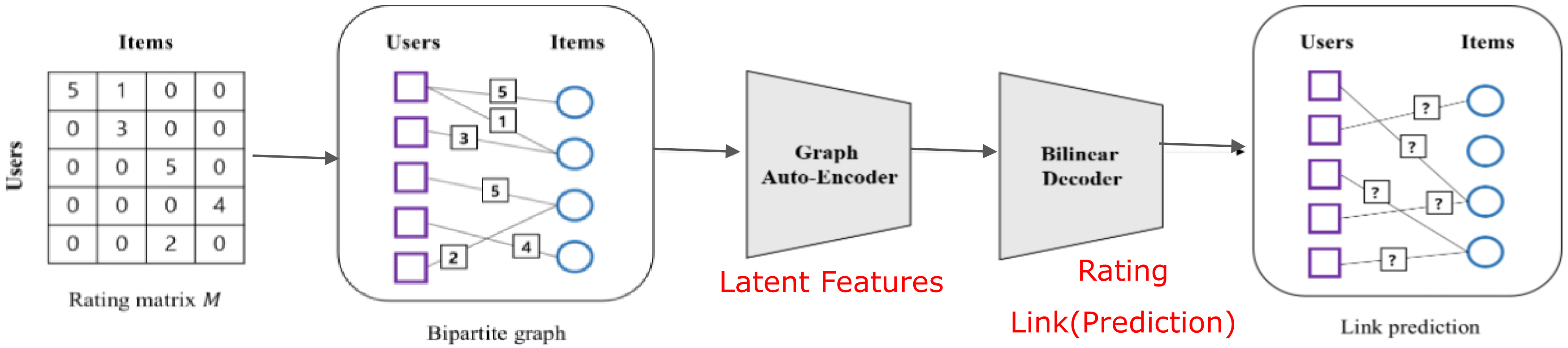


Wu, Shiwen, et al. "Graph neural networks in recommender systems: a survey." *arXiv preprint arXiv:2011.02260* (2020).

Recommendation System

Deep Learning Based Methods – CNN + GNN

- **Graph Convolutional Matrix Completion (GC-MC)**



- **Graph Auto-Encoder**: The auto-encoder produces latent features of user and item nodes through a form of message passing on the bipartite interaction graph.
- **Bilinear Decoder**: Latent user and item representations are used to reconstruct the rating links through a bilinear decoder. (Link prediction)

Berg, Rianne van den, Thomas N. Kipf, and Max Welling. "Graph convolutional matrix completion." *arXiv preprint arXiv:1706.02263* (2017).

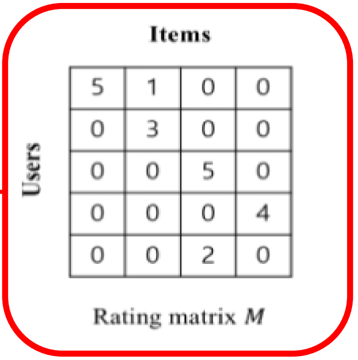
Recommendation System

Deep Learning Based Methods – CNN + GNN

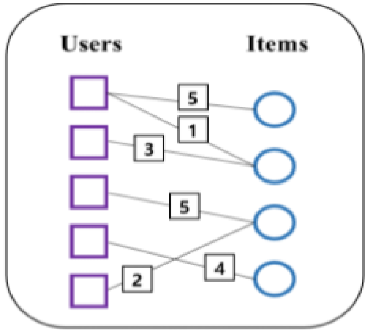
- **Graph Convolutional Matrix Completion (GC-MC)**

  - Step 1) User-Item Rating Matrix

$$M_{N_u \times N_v} = \begin{bmatrix} & & \\ & M_{ij} & \\ & & \end{bmatrix}, \quad M_{ij} = \begin{cases} r & \text{observed,} \\ 0 & \text{unobserved} \end{cases}$$



Rating matrix M



Bipartite graph

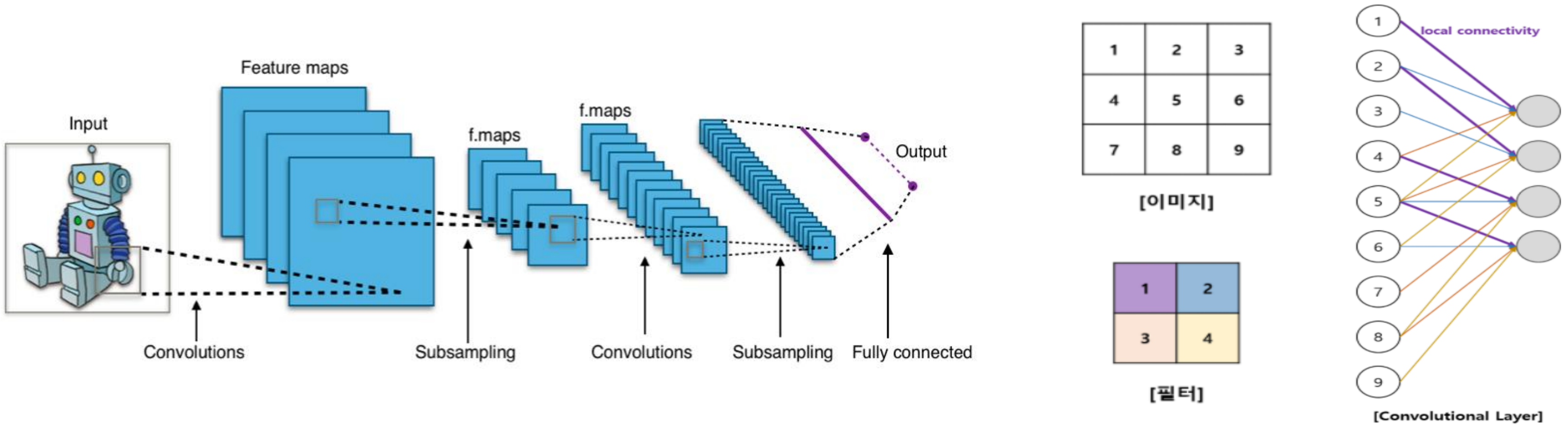
- Creating user-item rating matrix M: row users and column items
- Matrix encode either an observed rating from a set of discrete possible rating values, or unobserved rating fill in 0.
- Users & Items : Node Rating value: edge

Berg, Rianne van den, Thomas N. Kipf, and Max Welling. "Graph convolutional matrix completion." *arXiv preprint arXiv:1706.02263* (2017).

Recommendation System

Deep Learning Based Methods – CNN + GNN

- **Graph Convolutional Matrix Completion (GC-MC)**
  - **Step 2) Graph auto-encoders :** graph auto-encoder produces latent features of user and item nodes with graph convolutional network (GCN).
  - Graph Convolutional Network also has two characteristics: abstract feature + weight sharing



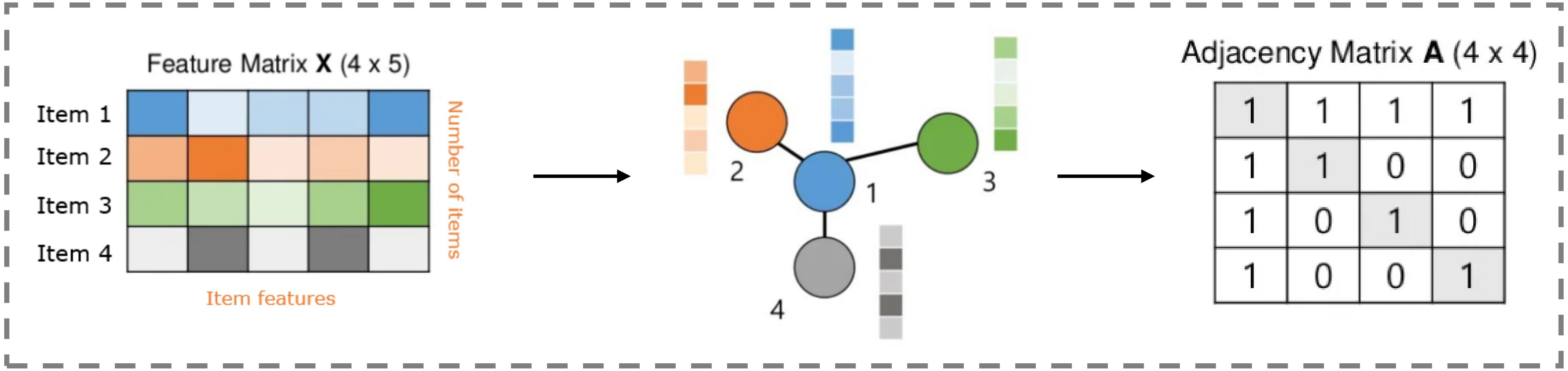
[fig 12] Convolutional Neural Network characteristics

Berg, Rianne van den, Thomas N. Kipf, and Max Welling. "Graph convolutional matrix completion." *arXiv preprint arXiv:1706.02263* (2017).

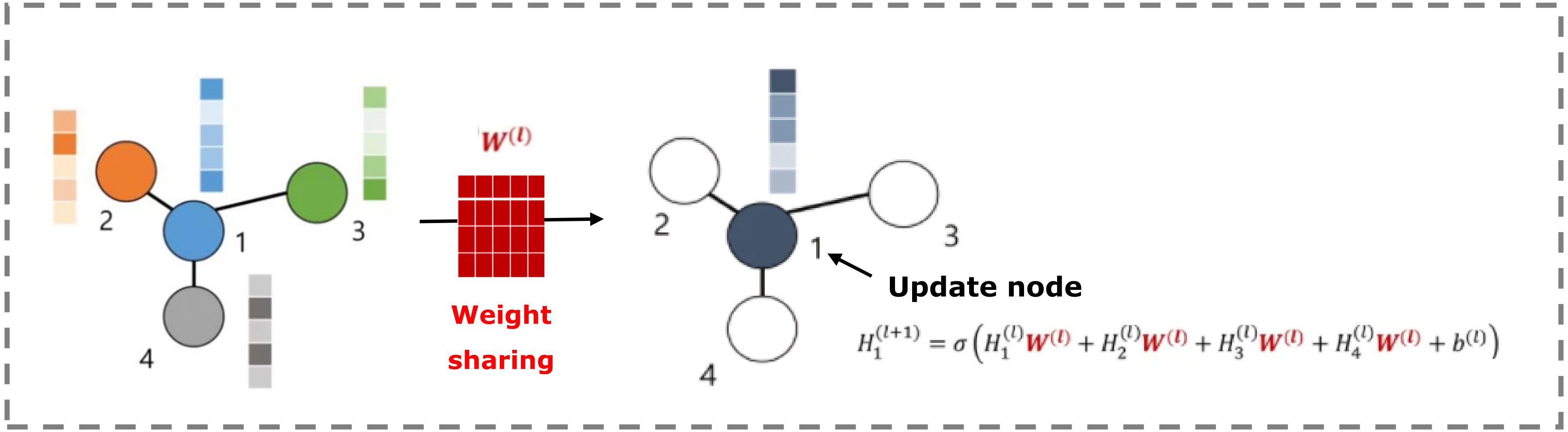
Graph Convolutional Network (GCN)

Graph Construction

$$H = \psi(X, A) = \sigma(AXW)$$



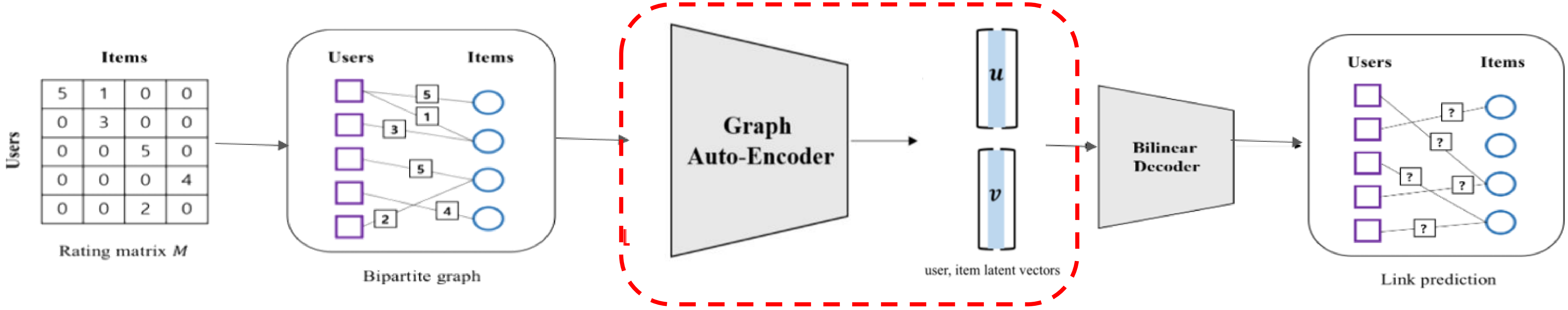
Node Update



Recommendation System

Deep Learning Based Methods – CNN + GNN

- **Graph Convolutional Matrix Completion (GC-MC) - Encoder**



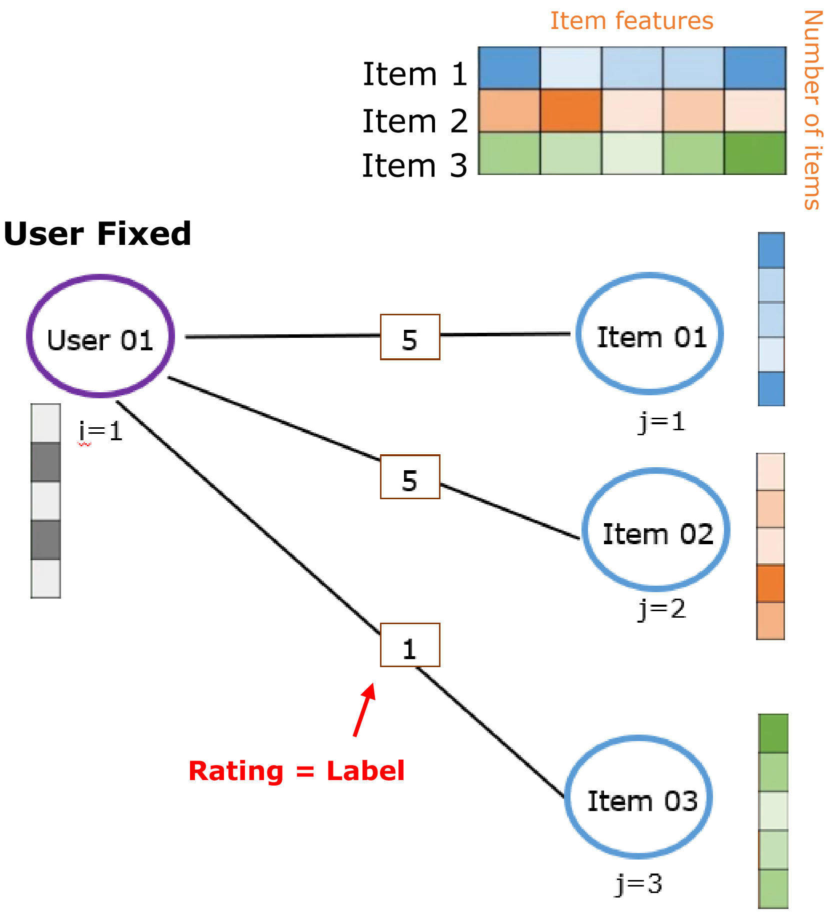
- **Graph Auto-Encoder**: The auto-encoder produces latent features of user and item nodes through a form of message passing on the bipartite interaction graph.

Berg, Rianne van den, Thomas N. Kipf, and Max Welling. "Graph convolutional matrix completion." *arXiv preprint arXiv:1706.02263* (2017).

Graph Convolutional Encoder Example

User latent vectors example

→ Message Passing, Hidden Layer, Transformation three step



- 1. Message Passing

$$\mu_{j\to i,r} := W_r x_j^T$$

(User i & Item j & rating edge r)

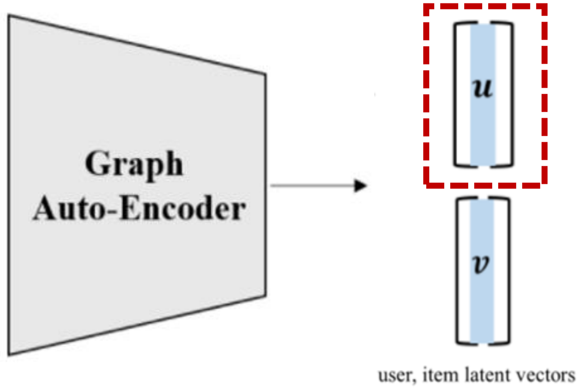
$$\sigma\left[\mathrm{accum}(\mu_{3\to 1,1}\ ,\mu_{2\to 1,5}\ ,\mu_{1\to 1,5}\ )\right]$$

- $\mu_{1\to 1,5} = W_{r=5} x_{j=1}^T = \begin{bmatrix} W_{11}^5 & W_{12}^5 & \cdots & W_{1D}^5 \\ W_{21}^5 & W_{22}^5 & \cdots & W_{2D}^5 \\ \cdots & \cdots & \cdots & \cdots \\ W_{E1}^5 & W_{E2}^5 & \cdots & W_{ED}^5 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ \cdots \\ x_{1D} \end{bmatrix} = \begin{bmatrix} s_{11}^5 \\ s_{12}^5 \\ \cdots \\ s_{1E}^5 \end{bmatrix}$

- $\mu_{2\to 1,5} = W_{r=5} x_{j=2}^T = \begin{bmatrix} W_{11}^5 & W_{12}^5 & \cdots & W_{1D}^5 \\ W_{21}^5 & W_{22}^5 & \cdots & W_{2D}^5 \\ \cdots & \cdots & \cdots & \cdots \\ W_{E1}^5 & W_{E2}^5 & \cdots & W_{ED}^5 \end{bmatrix} \begin{bmatrix} x_{21} \\ x_{22} \\ \cdots \\ x_{2D} \end{bmatrix} = \begin{bmatrix} s_{21}^5 \\ s_{22}^5 \\ \cdots \\ s_{2E}^5 \end{bmatrix}$

Identical Rating → Weight Sharing

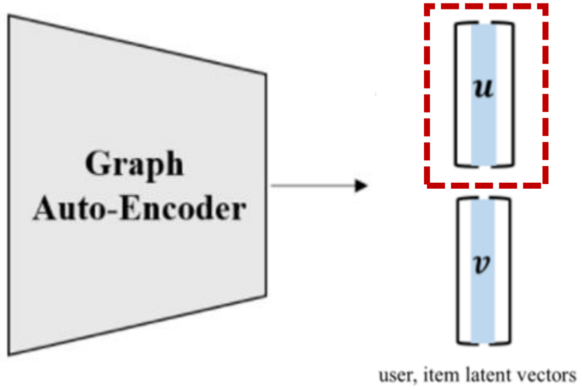
- $\mu_{3\to 1,1} = W_{r=1} x_{j=3}^T = \begin{bmatrix} W_{11}^1 & W_{12}^1 & \cdots & W_{1D}^1 \\ W_{21}^1 & W_{22}^1 & \cdots & W_{2D}^1 \\ \cdots & \cdots & \cdots & \cdots \\ W_{E1}^1 & W_{E2}^1 & \cdots & W_{ED}^1 \end{bmatrix} \begin{bmatrix} x_{31} \\ x_{32} \\ \cdots \\ x_{3D} \end{bmatrix} = \begin{bmatrix} s_{21}^1 \\ s_{22}^1 \\ \cdots \\ s_{2E}^1 \end{bmatrix}$



Berg, Rianne van den, Thomas N. Kipf, and Max Welling. "Graph convolutional matrix completion." *arXiv preprint arXiv:1706.02263* (2017).

Graph Convolutional Encoder Example

User latent vectors example



- **1.Message Passing**

$$\mu_{j\rightarrow i,r} := W_r x_j^T$$

(User i & Item j & rating edge r)

$$\sigma\left[\text{accum}(\mu_{3\rightarrow 1,1}, \mu_{2\rightarrow 1,5}, \mu_{1\rightarrow 1,5})\right]$$

- $\mu_{1\rightarrow 1,5} = W_{r=5} x_{j=1}^T = \begin{bmatrix} W^5_{11} & W^5_{12} & \cdots & W^5_{1D} \\ W^5_{21} & W^5_{22} & \cdots & W^5_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ W^5_{E1} & W^5_{E2} & \cdots & W^5_{ED} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ \cdots \\ x_{1D} \end{bmatrix} = \begin{bmatrix} s^5_{11} \\ s^5_{12} \\ \cdots \\ s^5_{1E} \end{bmatrix}$

- $\mu_{2\rightarrow 1,5} = W_{r=5} x_{j=2}^T = \begin{bmatrix} W^5_{11} & W^5_{12} & \cdots & W^5_{1D} \\ W^5_{21} & W^5_{22} & \cdots & W^5_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ W^5_{E1} & W^5_{E2} & \cdots & W^5_{ED} \end{bmatrix} \begin{bmatrix} x_{21} \\ x_{22} \\ \cdots \\ x_{2D} \end{bmatrix} = \begin{bmatrix} s^5_{21} \\ s^5_{22} \\ \cdots \\ s^5_{2E} \end{bmatrix}$

Identical Rating → Weight Sharing

- $\mu_{3\rightarrow 1,1} = W_{r=1} x_{j=3}^T = \begin{bmatrix} W^1_{11} & W^1_{12} & \cdots & W^1_{1D} \\ W^1_{21} & W^1_{22} & \cdots & W^1_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ W^1_{E1} & W^1_{E2} & \cdots & W^1_{ED} \end{bmatrix} \begin{bmatrix} x_{31} \\ x_{32} \\ \cdots \\ x_{3D} \end{bmatrix} = \begin{bmatrix} s^1_{21} \\ s^1_{22} \\ \cdots \\ s^1_{2E} \end{bmatrix}$

Normalization method

1. Left Normalization

$$\mu_{j\rightarrow i,r} = \frac{1}{|\mathcal{N}_i|} W_r x_j$$

2. Symmetric Normalization

$$\mu_{j\rightarrow i,r} = \frac{1}{|\mathcal{N}_i||\mathcal{N}_j|} W_r x_j$$

$$\sigma\left[\text{accum}(\mu_{3\rightarrow 1,1}, \mu_{2\rightarrow 1,5}, \mu_{1\rightarrow 1,5})\right]$$

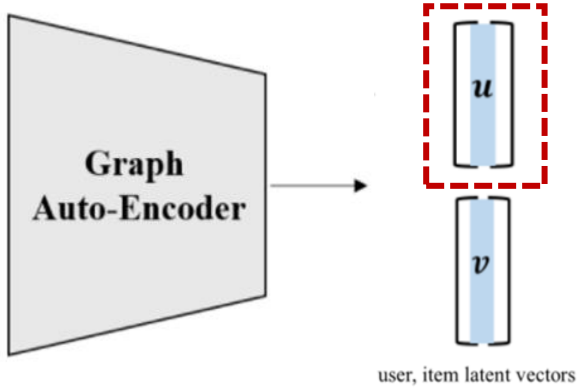
$$\text{accum}(.) = \text{sum}(.)$$

$$h_1 = \begin{bmatrix} \sigma(s^1_{21} + s^5_{11}) \\ \sigma(s^1_{22} + s^5_{12}) \\ \cdots \\ \sigma(s^1_{2E} + s^5_{1E}) \end{bmatrix}$$

Berg, Rianne van den, Thomas N. Kipf, and Max Welling. "Graph convolutional matrix completion." *arXiv preprint arXiv:1706.02263* (2017).

Graph Convolutional Encoder Example

User latent vectors example



- 2.Hidden Layer (Layer & weight dimension → hyperparameter)

$$h^{(1)}_{i(E_1\times 1)} = \sigma\left[\text{accum}\Big(\sum_{j\in\mathcal{N}_{i,1}} \mu^{(0)}_{j\to i,1}, \ldots, \sum_{j\in\mathcal{N}_{i,R}} \mu^{(0)}_{j\to i,R}\Big)\right], \quad \mu^{(0)}_{j\to i,r} = W^{(0)}_r x^T_j, \quad W^{(0)}_r \in R^{E_1\times D},\ x_j \in R^{1\times D}$$

$$h^{(2)}_{i(E_2\times 1)} = \sigma\left[\text{accum}\Big(\sum_{j\in\mathcal{N}_{i,1}} \mu^{(1)}_{j\to i,1}, \ldots, \sum_{j\in\mathcal{N}_{i,R}} \mu^{(1)}_{j\to i,R}\Big)\right], \quad \mu^{(1)}_{j\to i,r} = W^{(1)}_r h^{(1)}_i, \quad W^{(1)}_r \in R^{E_2\times E_1}$$

$$\ldots$$

$$h^{(l)}_{i(E_l\times 1)} = \sigma\left[\text{accum}\Big(\sum_{j\in\mathcal{N}_{i,1}} \mu^{(l-1)}_{j\to i,1}, \ldots, \sum_{j\in\mathcal{N}_{i,R}} \mu^{(l-1)}_{j\to i,R}\Big)\right], \quad \mu^{(l-1)}_{j\to i,r} = W^{(l-1)}_r h^{(l-1)}_i, \quad W^{(l-1)}_r \in R^{E_l\times E_{l-1}}$$

$E_2(E_2 < E_1)$

Abstract Feature

Different Rating → Different Weight Parameter

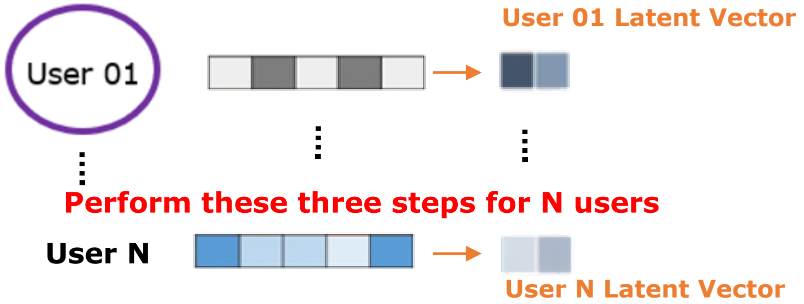
- 3.Transformation

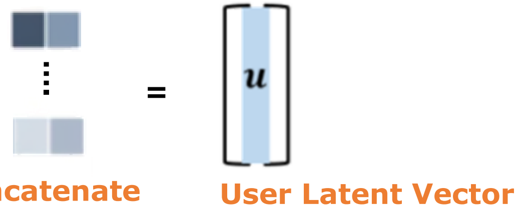
$$u^{(1)}_{i(E_1\times 1)} = \sigma(W^{(1)} h^{(1)}_i), \quad W^{(1)} \in R^{E_1\times E_1}$$

$$u^{(2)}_{i(E_2\times 1)} = \sigma(W^{(2)} h^{(2)}_i), \quad W^{(2)} \in R^{E_2\times E_2}$$

$$\ldots$$

$$u^{(l)}_{i(E_l\times 1)} = \sigma(W^{(l)} h^{(l)}_i), \quad W^{(l)} \in R^{E_l\times E_l}$$





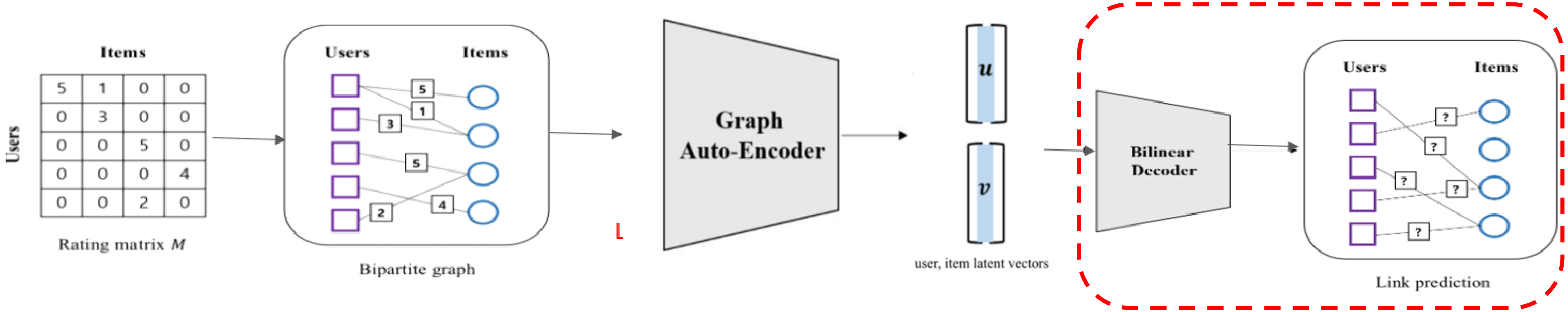
By multiplying all user i by the same weight W, all users are represented in the same space.

Berg, Rianne van den, Thomas N. Kipf, and Max Welling. "Graph convolutional matrix completion." *arXiv preprint arXiv:1706.02263* (2017).

Recommendation System

Deep Learning Based Methods – CNN + GNN

- **Graph Convolutional Matrix Completion (GC-MC) - Decoder**



- **Bilinear Decoder**: Latent user and item representations are used to reconstruct the rating links through a bilinear decoder. (Link prediction)

Berg, Rianne van den, Thomas N. Kipf, and Max Welling. "Graph convolutional matrix completion." *arXiv preprint arXiv:1706.02263* (2017).

GC-MC Bilinear Decoder & Model Training

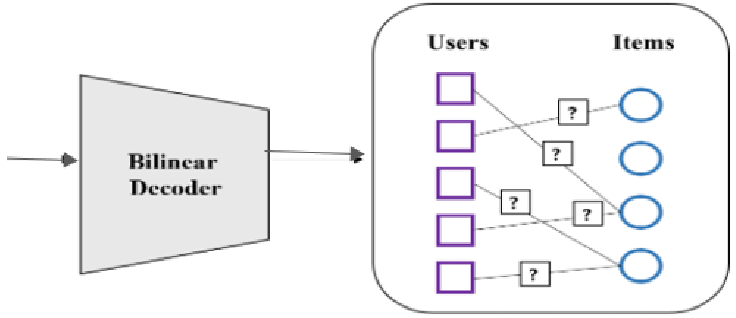
- Bilinear Decoder – Predict Link

$$\hat{M}_{N_u \times N_v} = g(U, V)$$

$$\hat{M}_{ij} = \sum_{r \in R} r\, p(\hat{M}_{ij} = r), \quad \text{where} \quad p(\hat{M}_{ij} = r) = \frac{e^{u_i^T Q_r v_j}}{\sum_{s \in R} e^{u_i^T Q_s v_j}}$$

Softmax

Rating of the user i for item j is predicted by weighted average of the rating.



Link prediction

- Model Training – Negative log likelihood

$$\mathcal{L} = - \sum_{i,j;\, \Omega_{i,j}=1} \sum_{r=1}^{R} I[r = M_{ij}] \log p(\hat{M}_{ij} = r), \quad \Omega \in \{0,1\}^{N_u \times N_v}$$

mask

To calculate the loss only for links with ratings, multiply the mask matrix filled with 1 for the observed element and 0 for the unobserved element.

Berg, Rianne van den, Thomas N. Kipf, and Max Welling. "Graph convolutional matrix completion." *arXiv preprint arXiv:1706.02263* (2017).

Recommendation System

Deep Learning Based Methods – CNN + GNN

- **GC-MC Experiments**

Model	Flixster	Douban	YahooMusic
GRALS	1.313/1.245	0.833	38.0
sRGCNN	1.179/0.926	0.801	22.4
GC-MC	**0.941/0.917**	**0.734**	**20.5**

Table 3: Average RMSE test set scores for 5 runs on Flixster, Douban, and YahooMusic, all of which include side information in the form of user and/or item graphs. We replicate the benchmark setting as in [22]. For Flixster, we show results for both user/item graphs (right number) and user graph only (left number). Baseline numbers are taken from [22].

GNN Model

Model	ML-1M	ML-10M
PMF [20]	0.883	–
I-RBM [26]	0.854	0.825
BiasMF [16]	0.845	0.803
NNMF [7]	0.843	–
LLORMA-Local [17]	0.833	0.782
I-AUTOREC [27]	0.831	0.782
CF-NADE [32]	**0.829**	**0.771**
GC-MC (Ours)	0.832	0.777

Table 4: Comparison of average test RMSE scores on five 90/10 training/test set splits (as in [32]) without the use of side information. Baseline scores are taken from [32]. For CF-NADE, we report the best-performing model variant. **MF Model**

Berg, Rianne van den, Thomas N. Kipf, and Max Welling. "Graph convolutional matrix completion." *arXiv preprint arXiv:1706.02263* (2017).

References

[1] Zhang, Shuai, et al. "Deep learning based recommender system: A survey and new perspectives." ACM Computing Surveys (CSUR) 52.1 (2019): 1-38.

[2] Das, Debashis, Laxman Sahoo, and Sujoy Datta. "A survey on recommendation system." *International Journal of Computer Applications* 160.7 (2017).

[3] MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS

(https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf)

[4] He, Xiangnan, et al. "Neural collaborative filtering." *Proceedings of the 26th international conference on world wide web*. 2017.

[5] Cheng, Heng-Tze, et al. "Wide & deep learning for recommender systems." *Proceedings of the 1st workshop on deep learning for recommender systems*. 2016.

[6] He, Xiangnan, et al. "Outer product-based neural collaborative filtering." *arXiv preprint arXiv:1808.03912* (2018).

[7] Du, Xiaoyu, et al. "Modeling embedding dimension correlations via convolutional neural collaborative filtering." *ACM Transactions on Information Systems (TOIS)* 37.4 (2019): 1-22.

[8] Zhu, Hangyu, and Maoting Gao. "Group recommendation method combining short-term interest and long-term preference." *Journal of Physics: Conference Series*. Vol. 2132. No. 1. IOP Publishing, 2021.

[9] Wu, Shiwen, et al. "Graph neural networks in recommender systems: a survey." *arXiv preprint arXiv:2011.02260* (2020).

[10] Scarselli, Franco, et al. "The graph neural network model." IEEE transactions on neural networks 20.1 (2008): 61-80.