Information Retrieval

Weeks 5. Recommender Systems

Yoon Suk Kang
School of Computer Science
Chungbuk National University

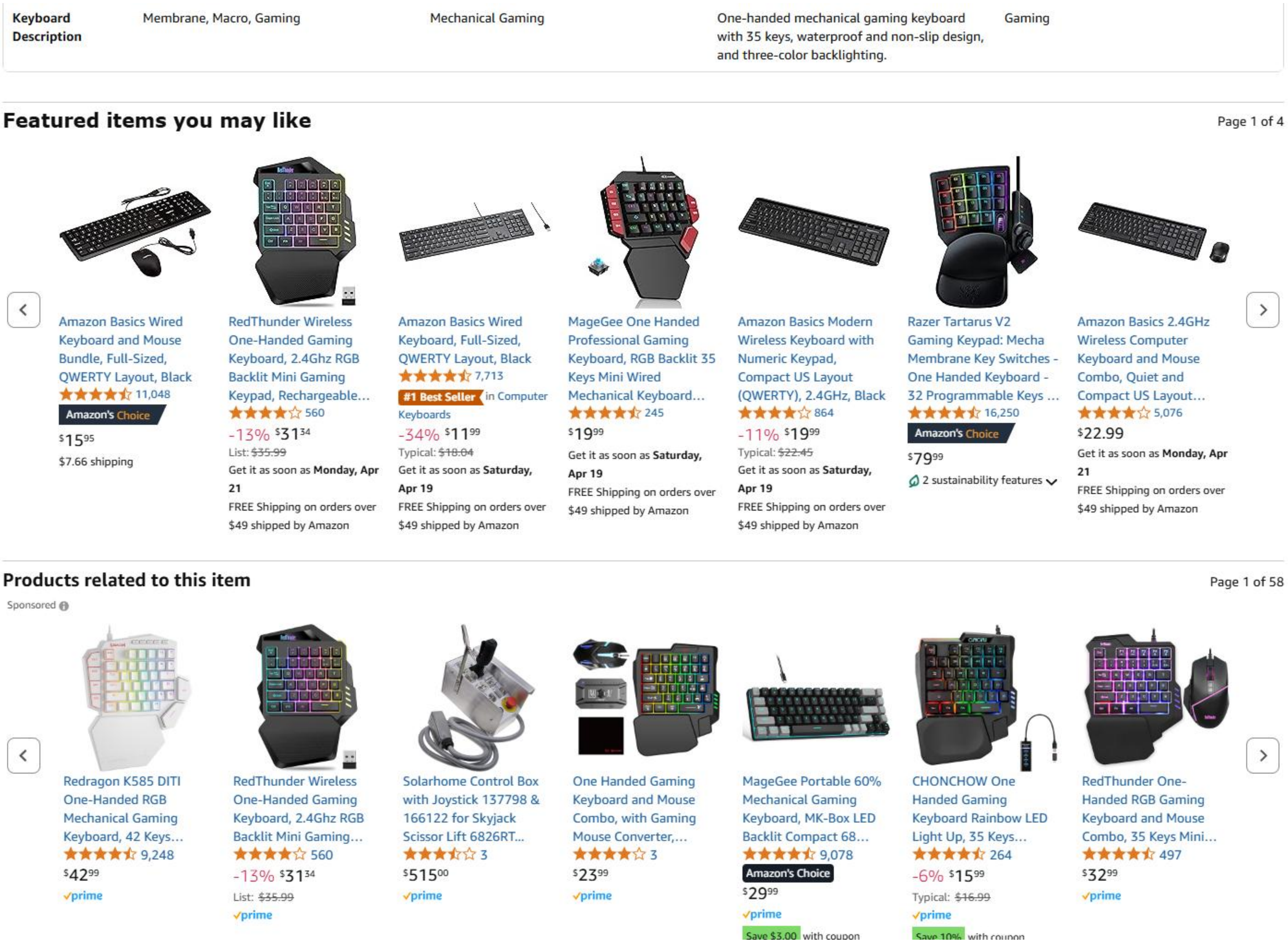
Outline

- ❑ Preliminary of Recommendation
- ❑ Recommender Systems Overview
- ❑ Data Imputation for Improving the Accuracy
- ❑ Summary

Preliminary of Recommendation

- **Information Explosion in the era of internet**
  - 10K+ movies in Netflix
  - 12M products in Amazon (350m on Marketplace)
  - 70M+ music tracks in Spotify
  - 10B+ videos on YouTube
  - 200B+ pins (images) in Pinterest
- **Personalized recommendation**
  - Suggesting a small number of interesting items for each user
  - It is critical for users to effectively explore the content of their interest

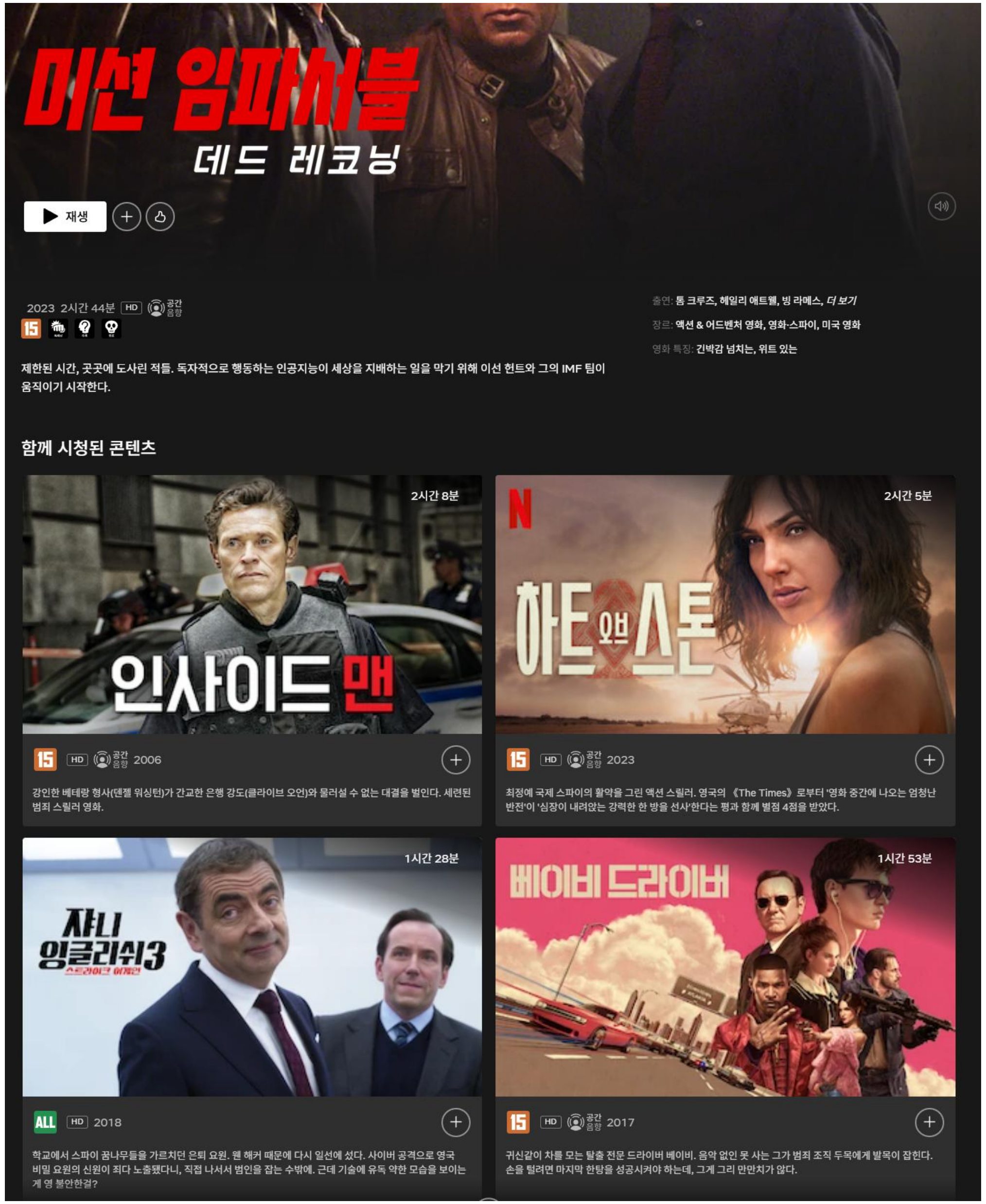
Amazon



Keyboard Description
Membrane, Macro, Gaming
Mechanical Gaming
One-handed mechanical gaming keyboard with 35 keys, waterproof and non-slip design, and three-color backlighting.
Gaming
Featured items you may like
Page 1 of 4
Amazon Basics Wired Keyboard and Mouse Bundle, Full-Sized, QWERTY Layout, Black
11,048
Amazon's Choice
$15.95
$7.66 shipping
RedThunder Wireless One-Handed Gaming Keyboard, 2.4Ghz RGB Backlit Mini Gaming Keypad, Rechargeable…
560
-13% $31.34
List: $35.99
Get it as soon as Monday, Apr 21
FREE Shipping on orders over $49 shipped by Amazon
Amazon Basics Wired Keyboard, Full-Sized, QWERTY Layout, Black
7,713
#1 Best Seller in Computer Keyboards
-34% $11.99
Typical: $18.04
Get it as soon as Saturday, Apr 19
FREE Shipping on orders over $49 shipped by Amazon
MageGee One Handed Professional Gaming Keyboard, RGB Backlit 35 Keys Mini Wired Mechanical Keyboard…
245
$19.99
Get it as soon as Saturday, Apr 19
FREE Shipping on orders over $49 shipped by Amazon
Amazon Basics Modern Wireless Keyboard with Numeric Keypad, Compact US Layout (QWERTY), 2.4GHz, Black
864
-11% $19.99
Typical: $22.45
Get it as soon as Saturday, Apr 19
FREE Shipping on orders over $49 shipped by Amazon
Razer Tartarus V2 Gaming Keypad: Mecha Membrane Key Switches - One Handed Keyboard - 32 Programmable Keys …
16,250
Amazon's Choice
$79.99
2 sustainability features
Amazon Basics 2.4GHz Wireless Computer Keyboard and Mouse Combo, Quiet and Compact US Layout…
5,076
$22.99
Get it as soon as Monday, Apr 21
FREE Shipping on orders over $49 shipped by Amazon
Products related to this item
Page 1 of 58
Sponsored
Redragon K585 DITI One-Handed RGB Mechanical Gaming Keyboard, 42 Keys…
9,248
$42.99
prime
RedThunder Wireless One-Handed Gaming Keyboard, 2.4Ghz RGB Backlit Mini Gaming…
560
-13% $31.34
List: $35.99
prime
Solarhome Control Box with Joystick 137798 & 166122 for Skyjack Scissor Lift 6826RT…
3
$515.00
prime
One Handed Gaming Keyboard and Mouse Combo, with Gaming Mouse Converter,…
3
$23.99
prime
MageGee Portable 60% Mechanical Gaming Keyboard, MK-Box LED Backlit Compact 68…
9,078
Amazon's Choice
$29.99
prime
Save $3.00 with coupon
CHONCHOW One Handed Gaming Keyboard Rainbow LED Light Up, 35 Keys…
264
-6% $15.99
Typical: $16.99
prime
Save 10% with coupon
RedThunder One-Handed RGB Gaming Keyboard and Mouse Combo, 35 Keys Mini…
497
$32.99
prime

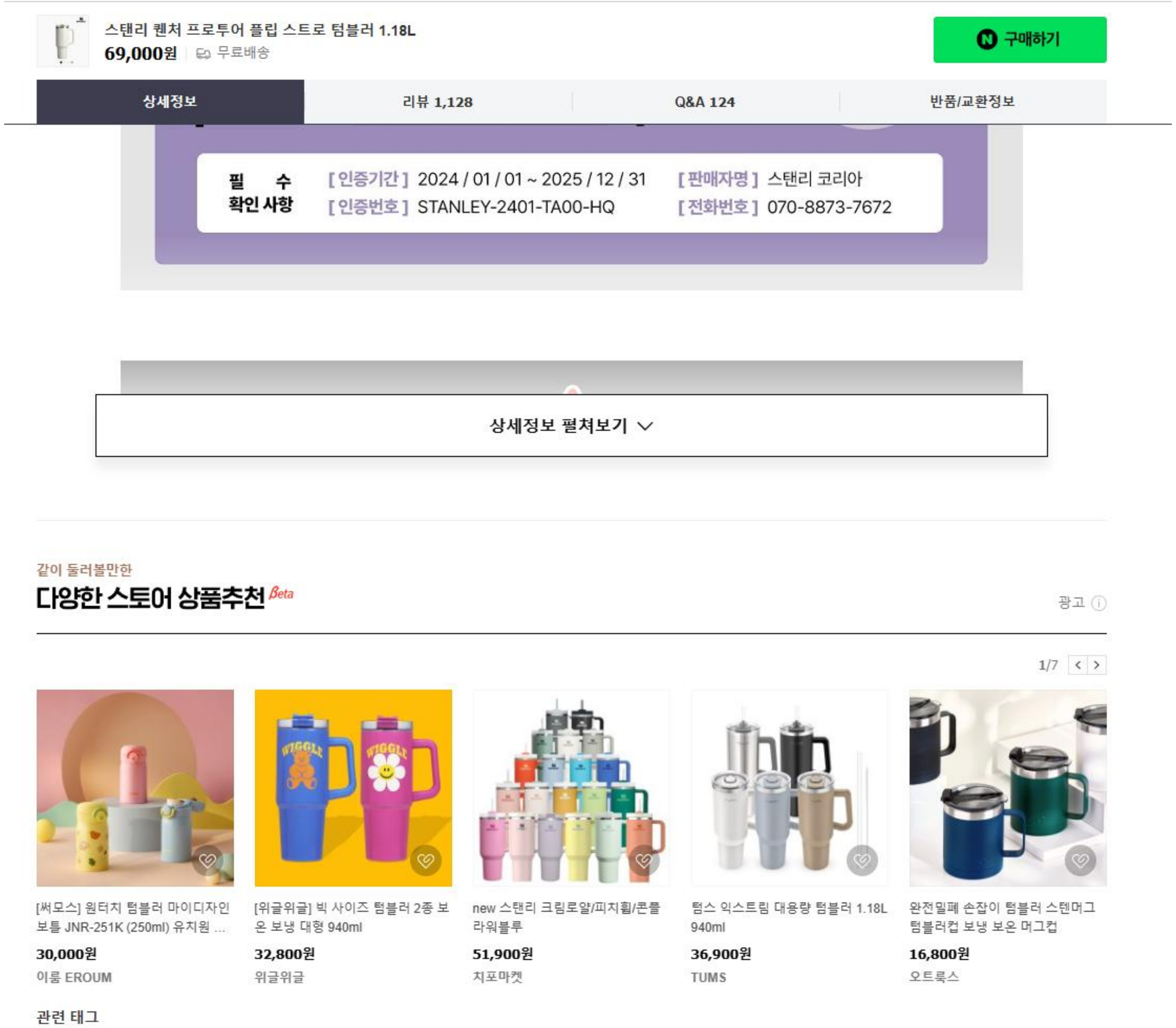
Netflix

레전드 드라마
올인
낭만닥터 김사부
시크릿가든
프라하의 연인
최신 등록
뿌리깊은 나무
닥터스
아련한 향수가 나를 부를 때
기묘한 이야기
STRANGER THINGS
드라이빙 미스 데이지
대부
타짜
신의 한 수 귀수편
코리아
즐거운 영어 세상 속으로!
하우스
퀸스 갬빗
더 크라운
THE CROWN
매니페스트
굿닥터
위쳐
내가 찜한 리스트
슈츠
비밀의 숲
브레이킹 배드
하우스 오브 카드
나르코스
NARCOS
지정생존자
바스터즈: 거친 녀석들과 비슷한 콘텐츠
트로이
알카트라즈 탈출
다크나이트
지오스톰
300
인셉션
한국 직장 TV 프로그램
친애하는 판사님께
최신 등록
정신병동에도 아침이 와요
사랑의 이해
의사요한
가족×멜로
밥 잘 사주는 예쁜 누나



미션 임파서블
데드 레코닝
재생
2023 2시간 44분 HD 공간 음향
15
출연: 톰 크루즈, 헤일리 애트웰, 빙 라메스, 더 보기
장르: 액션 & 어드벤처 영화, 영화·스파이, 미국 영화
영화 특징: 긴박감 넘치는, 위트 있는
제한된 시간, 곳곳에 도사린 적들. 독자적으로 행동하는 인공지능이 세상을 지배하는 일을 막기 위해 이선 헌트와 그의 IMF 팀이 움직이기 시작한다.
함께 시청된 콘텐츠
2시간 8분
인사이드 맨
15 HD 공간 음향 2006
강인한 베테랑 형사(덴젤 워싱턴)가 간교한 은행 강도(클라이브 오언)와 물러설 수 없는 대결을 벌인다. 세련된 범죄 스릴러 영화.
2시간 5분
하트 오브 스톤
15 HD 공간 음향 2023
최정예 국제 스파이의 활약을 그린 액션 스릴러. 영국의 《The Times》로부터 '영화 중간에 나오는 엄청난 반전'이 '심장이 내려앉는 강력한 한 방을 선사'한다는 평과 함께 별점 4점을 받았다.
1시간 28분
쟈니 잉글리쉬3
ALL HD 2018
학교에서 스파이 꿈나무들을 가르치던 은퇴 요원. 웬 해커 때문에 다시 일선에 섰다. 사이버 공격으로 영국 비밀 요원의 신원이 죄다 노출됐다니, 직접 나서서 범인을 잡는 수밖에. 근데 기술에 유독 약한 모습을 보이는 게 영 불안한걸?
1시간 53분
베이비 드라이버
15 HD 공간 음향 2017
귀신같이 차를 모는 탈출 전문 드라이버 베이비. 음악 없인 못 사는 그가 범죄 조직 두목에게 발목이 잡힌다. 손을 털려면 마지막 한탕을 성공시켜야 하는데, 그게 그리 만만치가 않다.

Naver Shopping

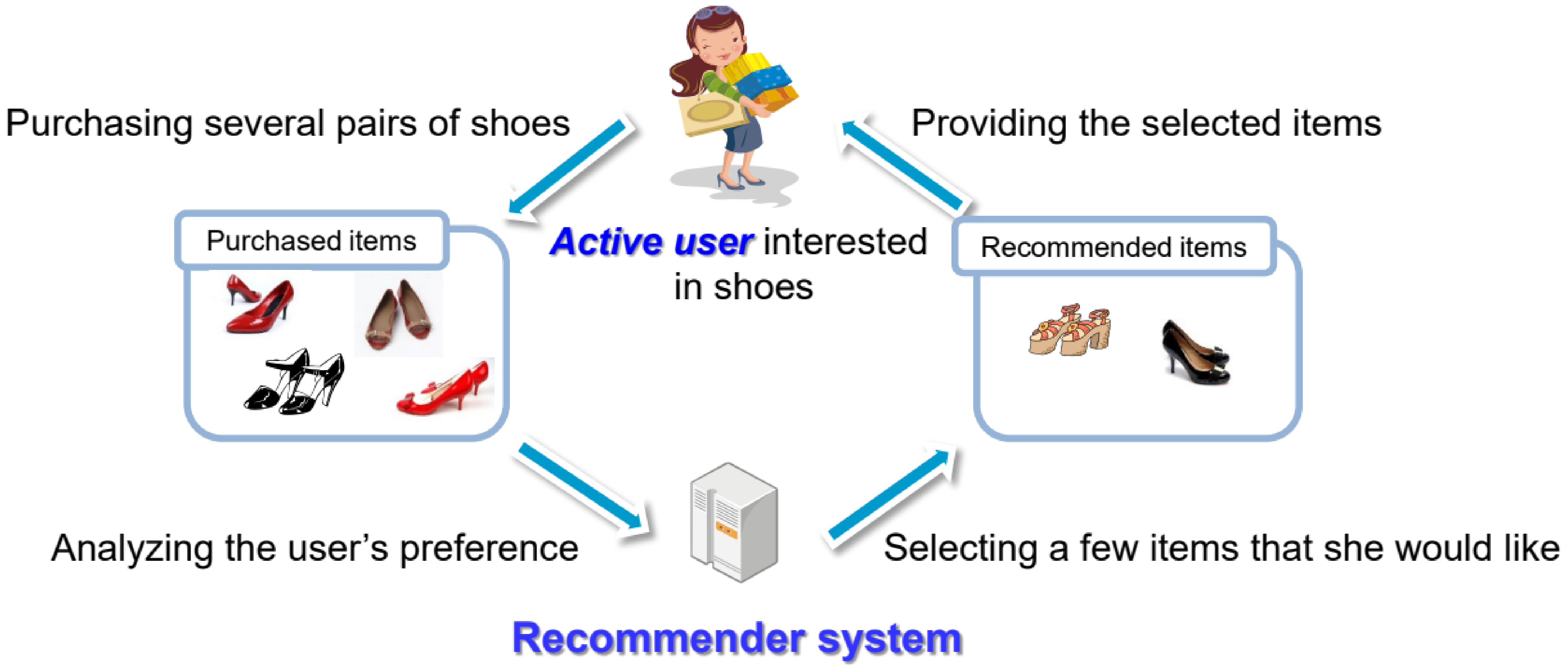


Outline

- ❑ Preliminary of Recommendation
- ❑ Recommender Systems Overview
- ❑ Data Imputation for Improving the Accuracy
- ❑ Summary

Recommender Systems

- Provide a user with a few items that she/he would like
  - Using her history of evaluating, purchasing, and browsing



Classification of Recommender Systems

❑ Content-based approach

- Recommending those items that have *similar contents* to those of the active user's favorite items

❑ Collaborative filtering (CF) approach

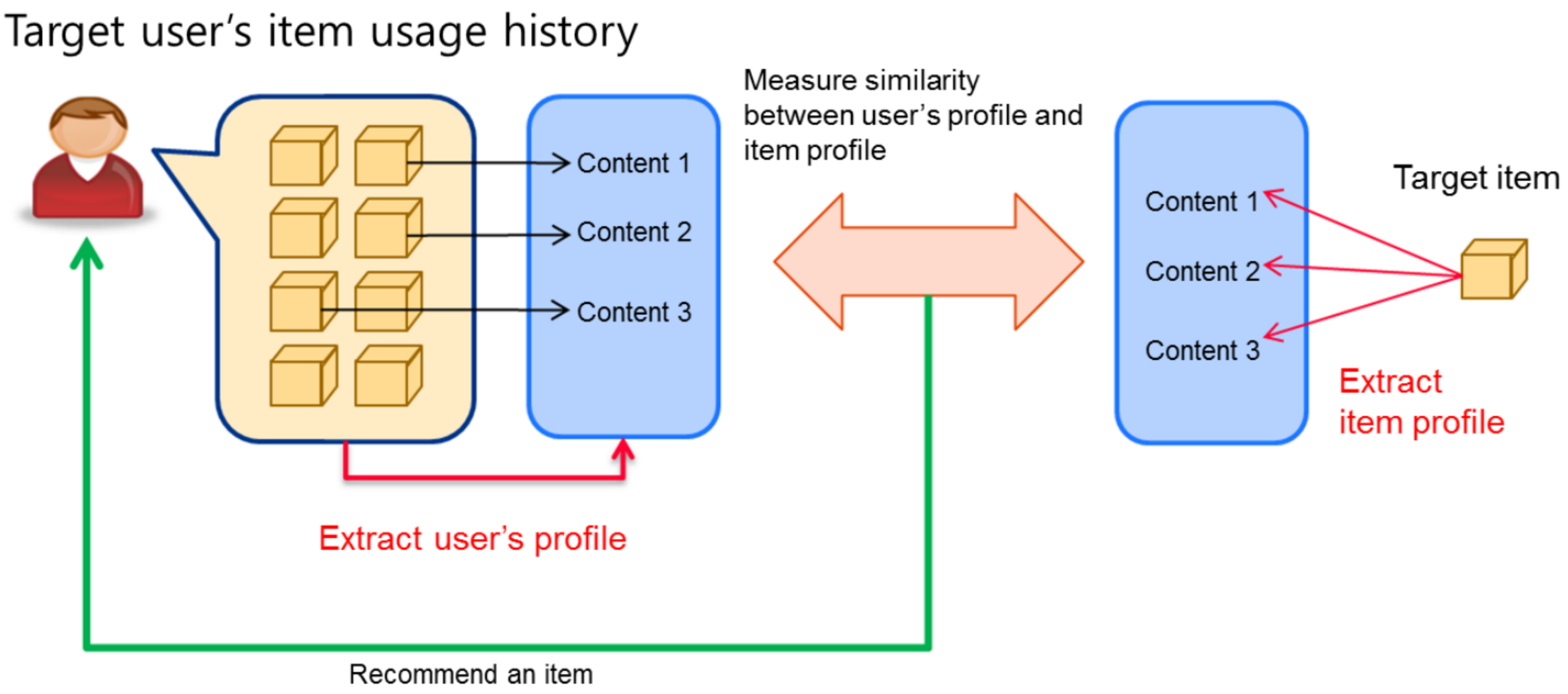
- Recommending items *related high by neighbors* who have preferences similar to that of the active user

❑ Hybrid approach

- Recommending items by *integrating the approaches above*

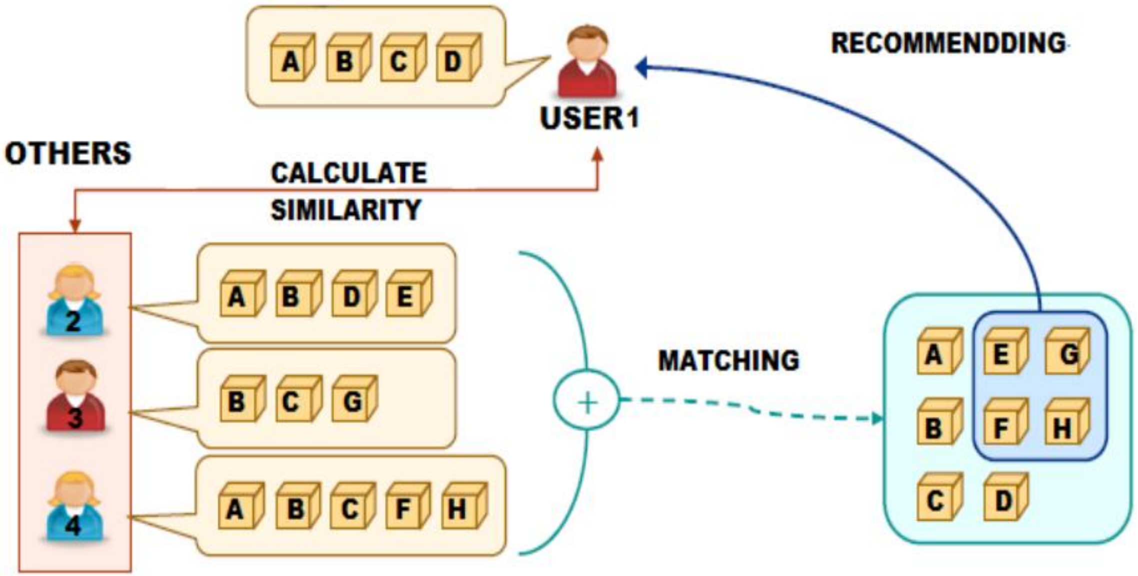
Content-Based Approach

- **Recommend items with similar contents to the profile of the active user**
  - Step 1: *Calculating similarity* between user's profile and item's profile
  - Step 2: Recommending *a few items with the similarities calculated high*



Collaborative Filtering (CF) Approach

- **Recommend items highly rated by users with preferences similar to the target user**
  - Step 1: *Finding a group of users* whose preferences are similar to that of a target user u
  - Step 2: *Estimating the rating of an item i for target user u* based on the ratings given to item i by u's neighbors
  - Step 3: Recommending *a few items with the ratings estimated high*



Heuristic-Based Method in CF

- Data: rating matrix

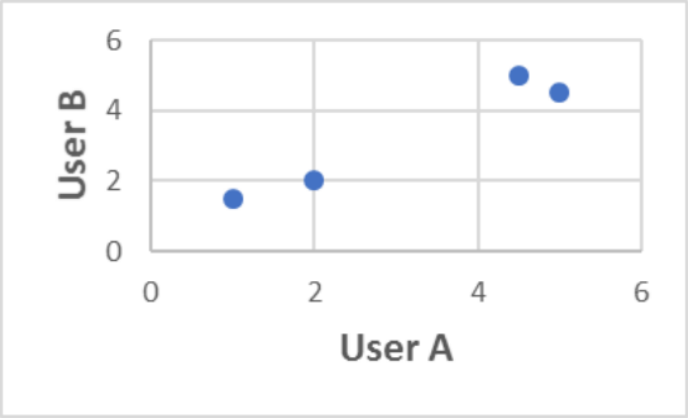
Items

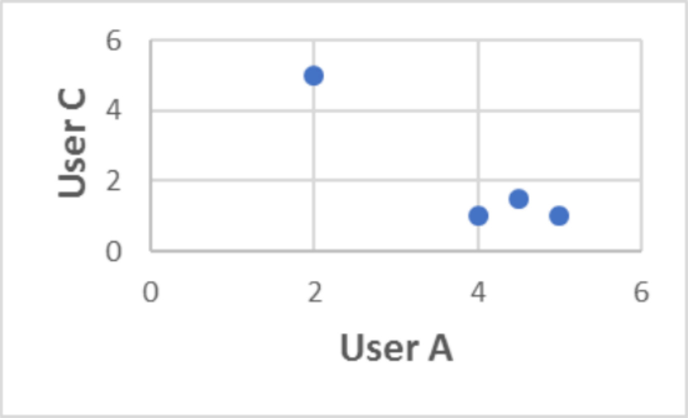
Users	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}	i_{11}	i_{12}
u_1		1		1		5	3			5		
u_2	1	3		4		4				2	4	
u_3												5
u_4		3		4		4		4	3			
u_5			4									
u_6	1			3	5	1		4			3	

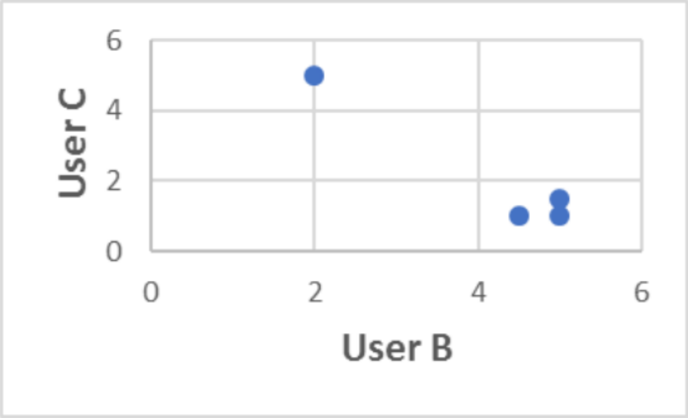
A rating on i_1 given by user u_6

Heuristic-Based Method in CF (cont.)

- **Similarity measure between users u and v (for neighbor identification)**
  - Example: three users' rating for six movies
    - User A = <4.0, 1.0, 4.5, 5.0, 2.0, __>
    - User B = <__, 1.5, 5.0, 4.5, 2.0, 5.0>
    - User C = <1.0, __, 1.5, 1.0, 5.0, 1.0>
  - *Pearson correlation coefficient (PCC)*
    - $w(u, v) = \frac{\sum_j (v_{u,j} - \overline{v_u})(v_{v,j} - \overline{v_v})}{\sqrt{\sum_j (v_{u,j} - \overline{v_u})^2} \sqrt{\sum_j (v_{v,j} - \overline{v_v})^2}}$

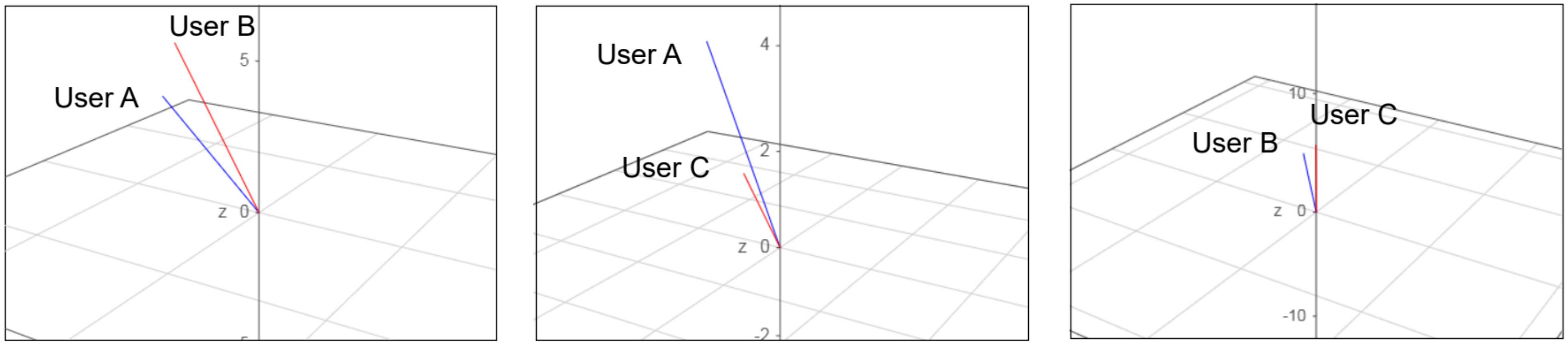






Heuristic-Based Method in CF (cont.)

- **Similarity measure between users $\boldsymbol{u}$ and $\boldsymbol{v}$ (for neighbor identification)**
  - Example: three users' rating for six movies
    - User A = <4.0, 1.0, 4.5, 5.0, 2.0, ___>
    - User B = <___, 1.5, 5.0, 4.5, 2.0, 5.0>
    - User C = <1.0, ___, 1.5, 1.0, 5.0, 1.0>
  - *Cosine similarity*
    - $w(u,v) = \sum_j \frac{v_{u,j} \cdot v_{v,j}}{\sqrt{\sum_{k \in I_u} v_{u,k}^2}\sqrt{\sum_{k \in I_v} v_{v,k}^2}}$



Heuristic-Based Method in CF (cont.)

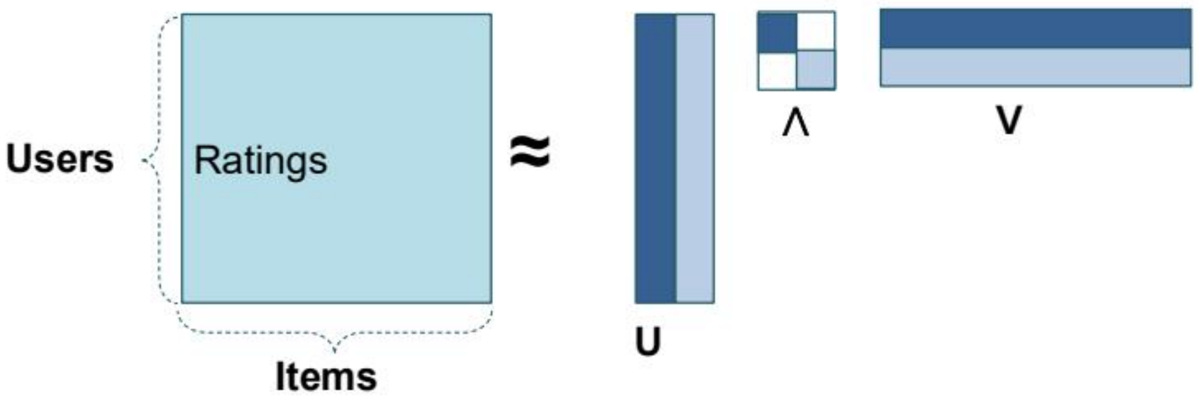
- **Aggregation of ratings on a target item given by the neighbors**

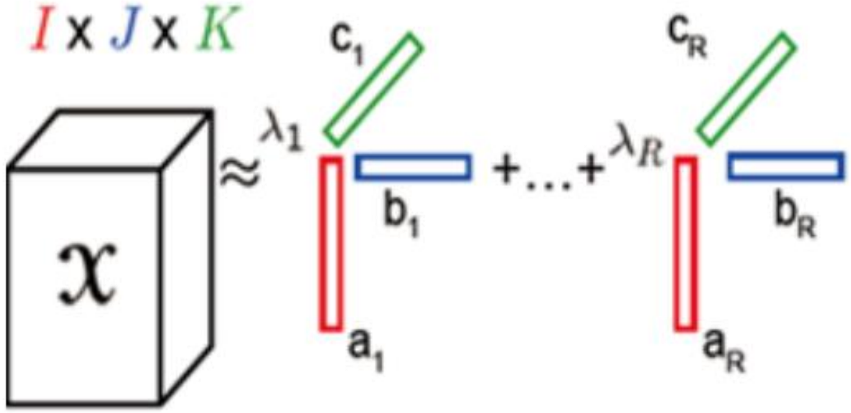
$$\boldsymbol{r_{c,s} = aggr_{c' \in \widehat{C}}\, r_{c',s}}$$

  - $r_{c,s}$: estimated rating on item *s* for user c
  - $\hat{C}$: set of neighbors for user *c*
  - Different methods for aggregation
    - $\boldsymbol{r_{c,s} = \frac{1}{N} \sum_{c' \in \widehat{C}} r_{c',s}}$
    - $\boldsymbol{r_{c,s} = k \sum_{c' \in \widehat{C}}} sim(c, c') \times \boldsymbol{r_{c',s}}$
    - $\boldsymbol{r_{c,s} = \overline{r_c} + k \sum_{c' \in \widehat{C}}} sim(c, c') \times (\boldsymbol{r_{c',s}} - \bar{r}_{\boldsymbol{c'}})$

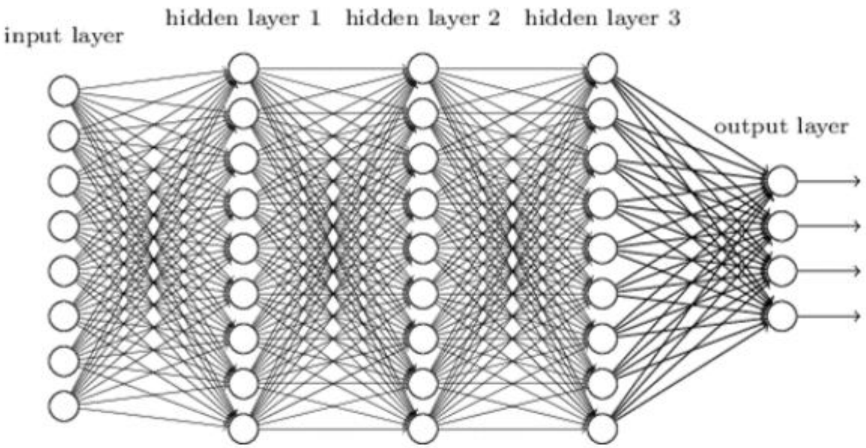
Machine-Learning-Based CF

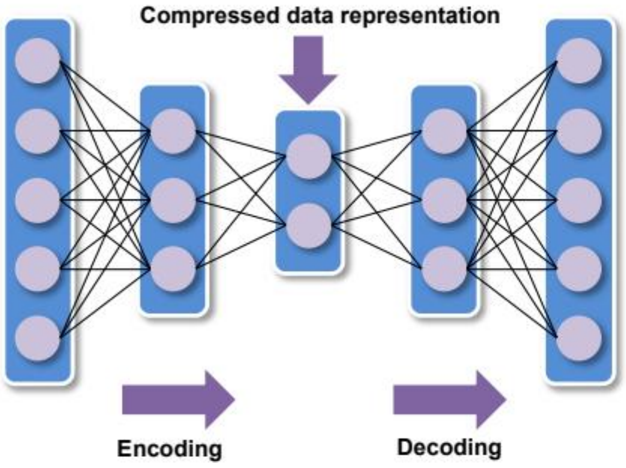
❑ Matrix/Tensor Factorization

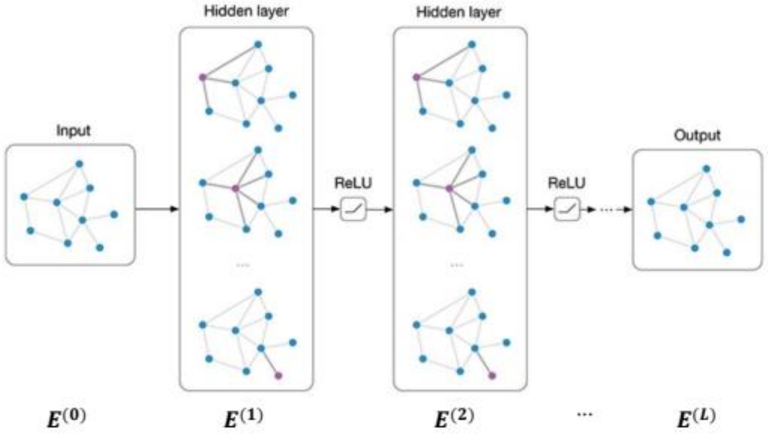




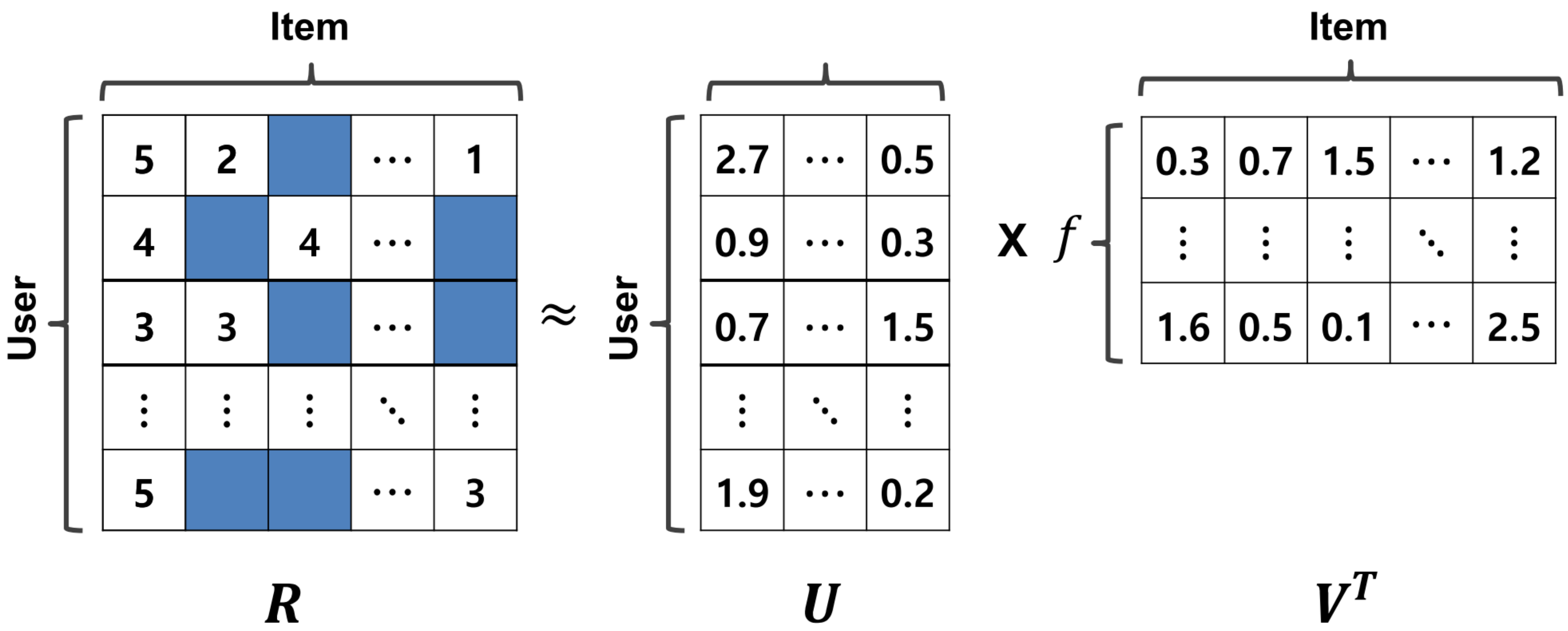
❑ Deep Learning







Matrix Factorization (MF)



- **Factorize a rating matrix *R* into two latent matrices *U* and *V***
  - *R*: represent the user-item rating matrix ($m \times n$ matrix)
  - *U*: represent the features of users as f latent factors ($m \times f$ matrix)
  - *V*: represent the features of items as f latent factors ($n \times f$ matrix)

Matrix Factorization (MF) (cont.)

$$\text{User}\left\{\overbrace{\begin{bmatrix} 2.7 & \cdots & 0.5 \\ 0.9 & \cdots & 0.3 \\ 0.7 & \cdots & 1.5 \\ \vdots & \ddots & \vdots \\ 1.9 & \cdots & 0.2 \end{bmatrix}}^{f}\right. \times f\left\{\overbrace{\begin{bmatrix} 0.3 & 0.7 & 1.5 & \cdots & 1.2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1.6 & 0.5 & 0.1 & \cdots & 2.5 \end{bmatrix}}^{\text{Item}}\right. = \text{User}\left\{\overbrace{\begin{bmatrix} 4.9 & 1.5 & 3.5 & \cdots & 1.2 \\ 3.5 & 2.4 & 4.0 & \cdots & 1.9 \\ 3.0 & 3.5 & 4.5 & \cdots & 0.5 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 4.8 & 1.3 & 3.4 & \cdots & 3.2 \end{bmatrix}}^{\text{Item}}\right.$$

$$\boldsymbol{U} \qquad\qquad \boldsymbol{V^T} \qquad\qquad \boldsymbol{\hat{R}}$$

- **Learn U and V iteratively by minimizing the squared error**

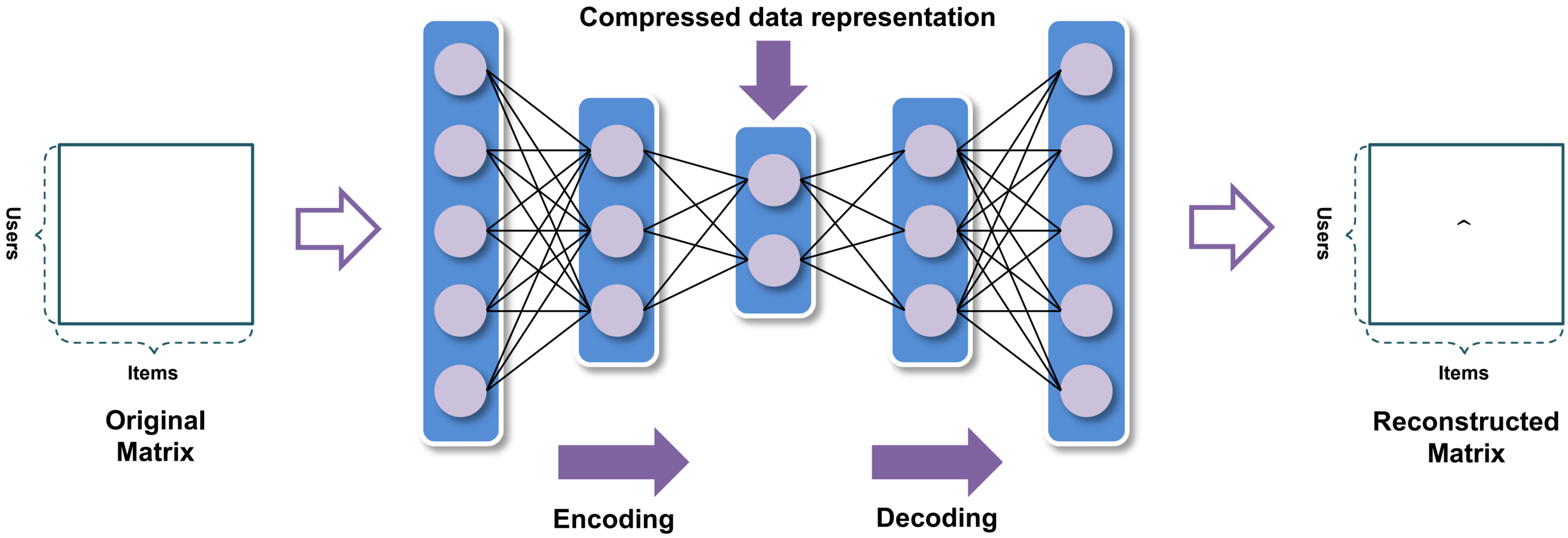
$$\underset{U,V}{\operatorname{argmin}} \sum_{i=1}^{m} \sum_{j=1}^{n} y_{ij}(r_{ui} - U_i V_j^T)^2 \qquad where\ \ y_{ij} \begin{cases} 1 & if\ r_{ui}\ exists \\ 0 & otherwise \end{cases}$$

  - *Stochastic gradient descent (SGD)* or alternating least square (ALS)

- **Estimate the missing rating of user *u* from item *v* as UV^T**

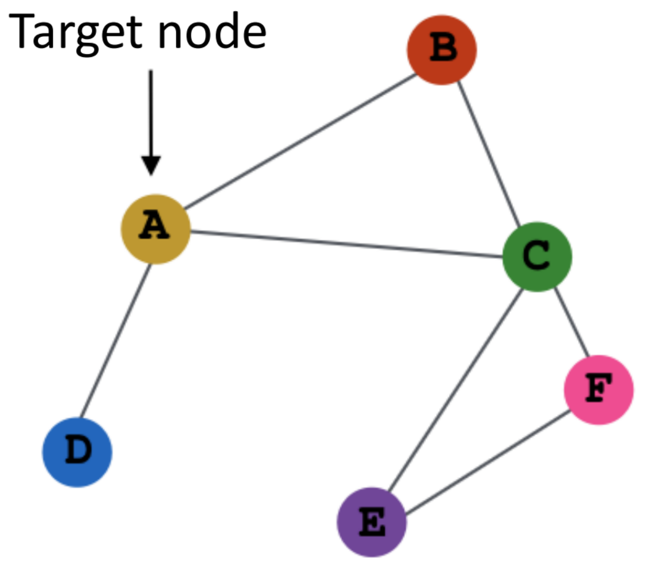
Autoencoder (AE)

- A feed-forward neural network trained to reconstruct its input at the output layer

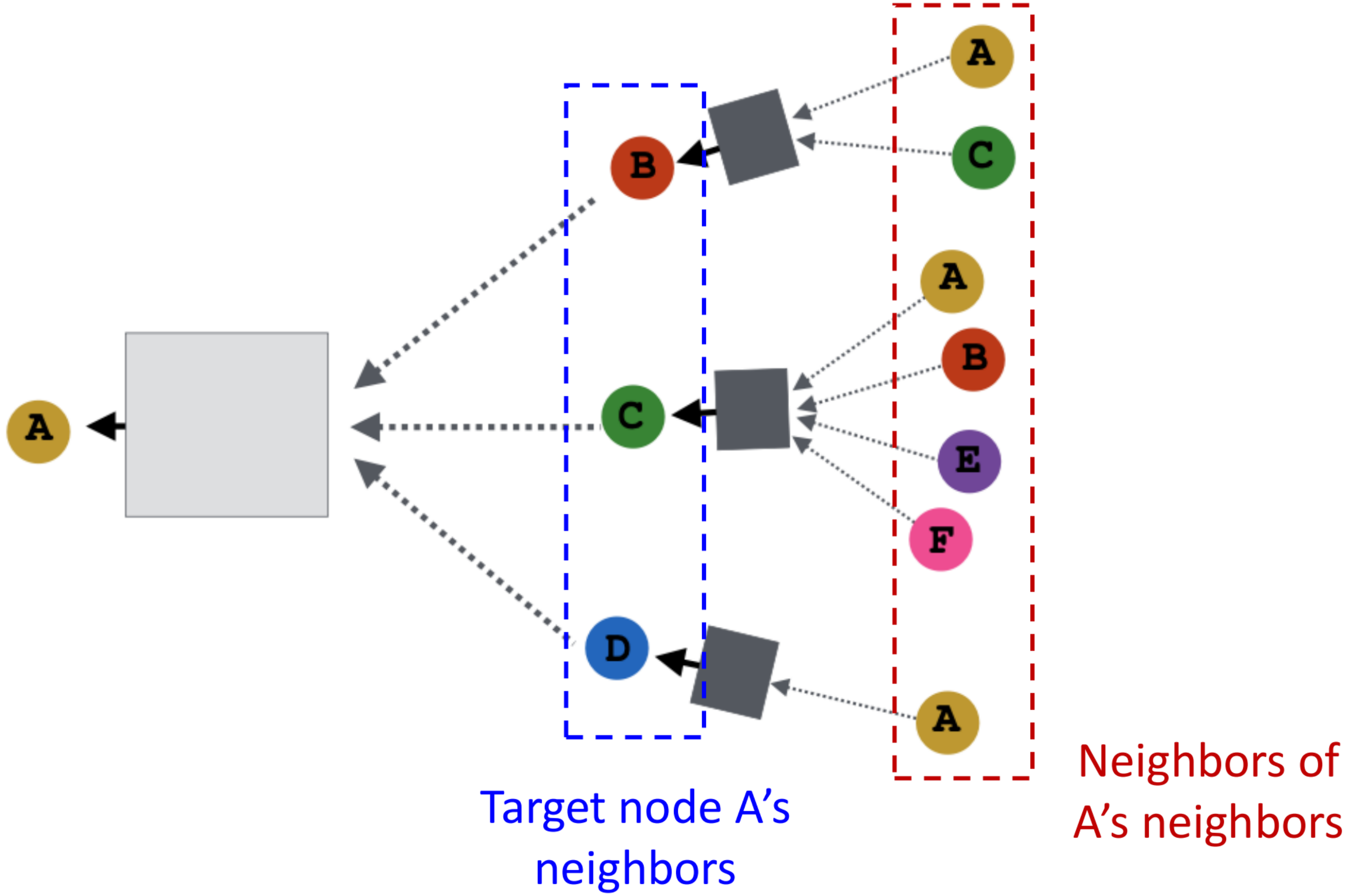


Graph Neural Network (GNN)

- **Key idea:** **aggregating information from neighbors**
  - GNNs generate a node's embedding based on its local network neighborhoods



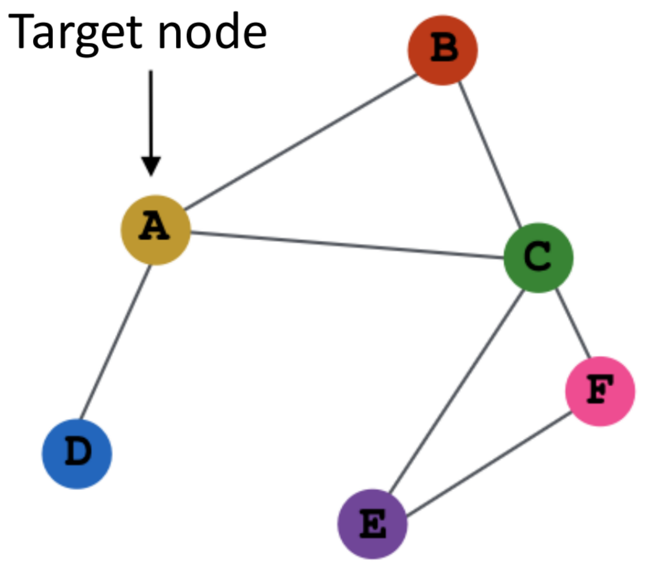
Graph G



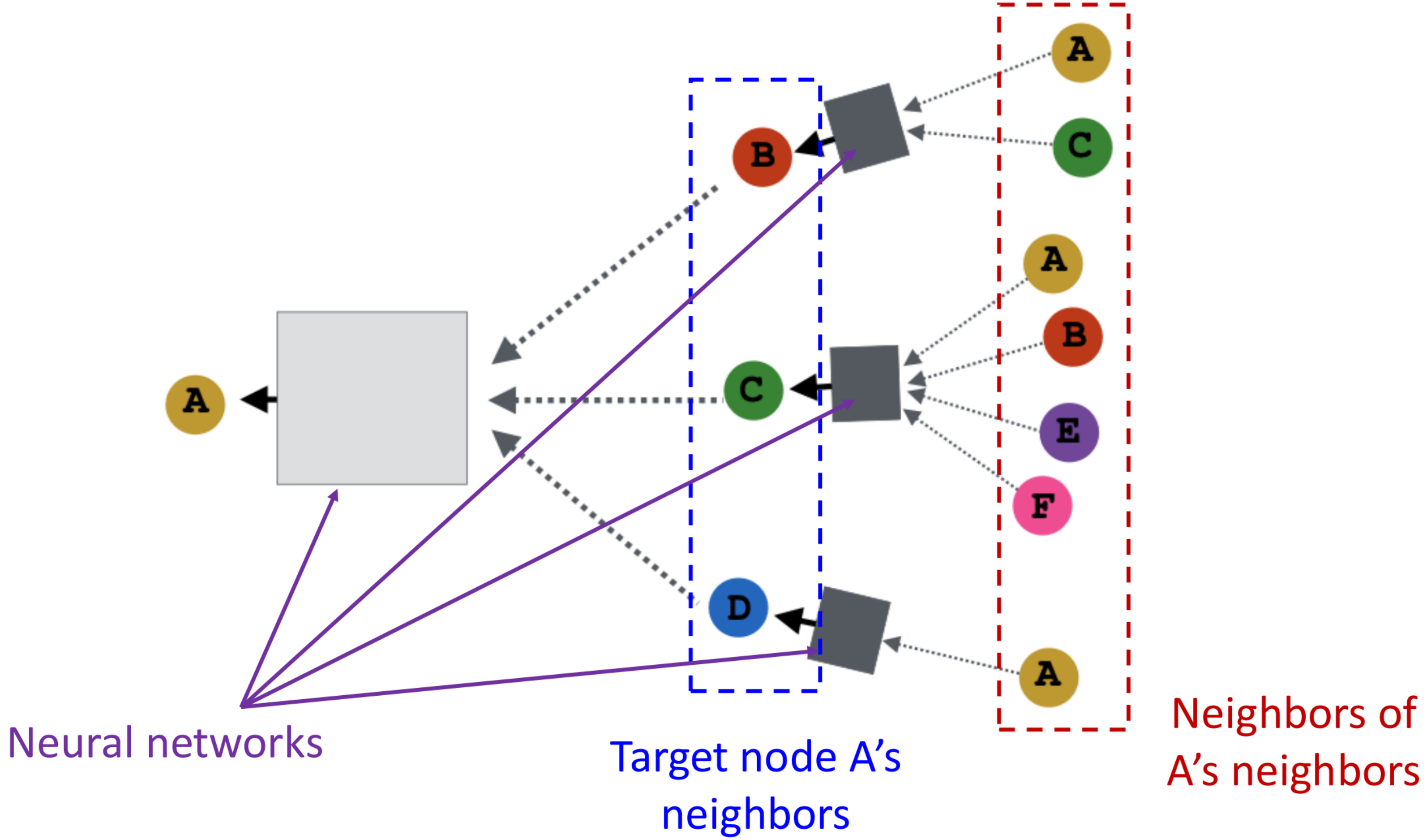
Graph Neural Network (GNN) (cont.)

❑ Intuition

- Each node *aggregates information from their neighbors* using neural networks



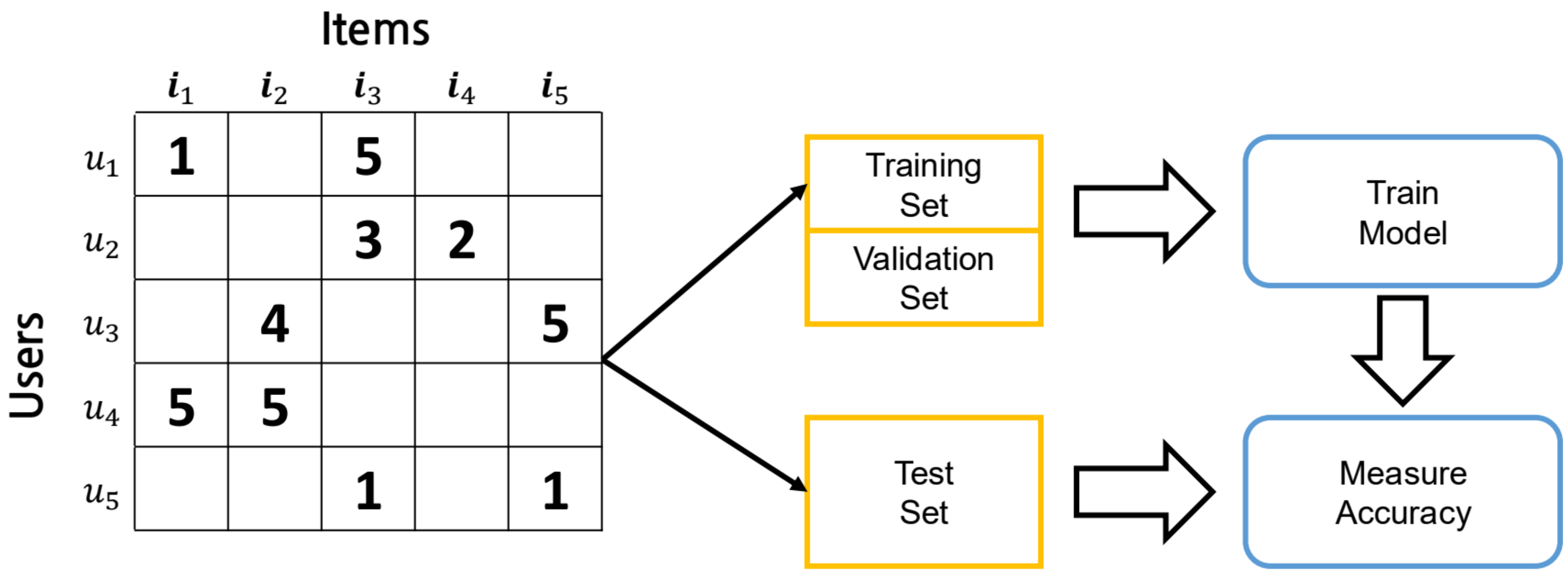
Graph *G*



Evaluation of Recommender Systems

❑ Overall procedure

- *Partition* a given dataset (e.g., a rating matrix) into *three complementary subsets*, i.e., training, validation, and test sets
- *Train* a recommendation model based on *the training set*
- *Optimize* a set of hyperparameters based on *the validation set*
- *Measure* the recommendation accuracy based on *the test set*

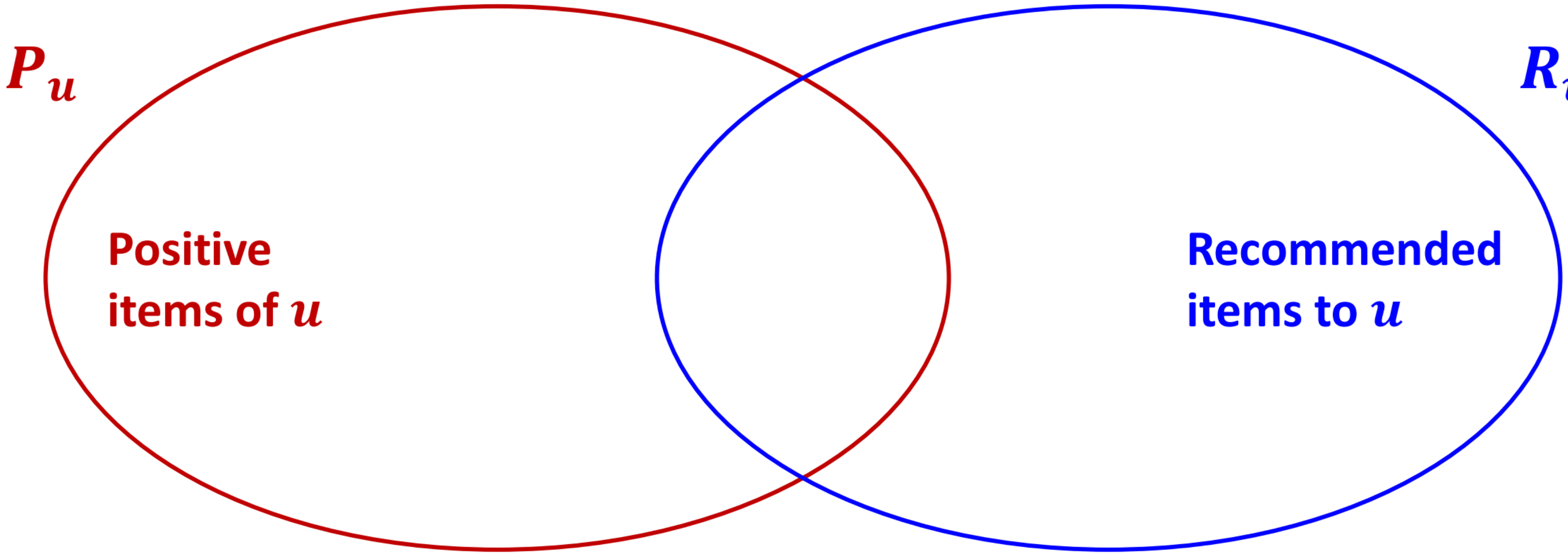


Top-*K* Recommendation

- **For each user, recommend *K* items**
  - For recommendation to be effective, *K needs to be much smaller than the total number of items*
  - K is typically in the order of 10 ~ 100
- **The goal is to include as many positive items as possible in the top-K recommended items**
  - Positive items indicates those that *the user will interact with in the future*
- **Popular evaluation metrics (defined next)**
  - *Precision, recall, and F1 score*
  - *Normalized discounted cumulative gain (NDCG)*
  - *Mean reciprocal rank (MRR)*

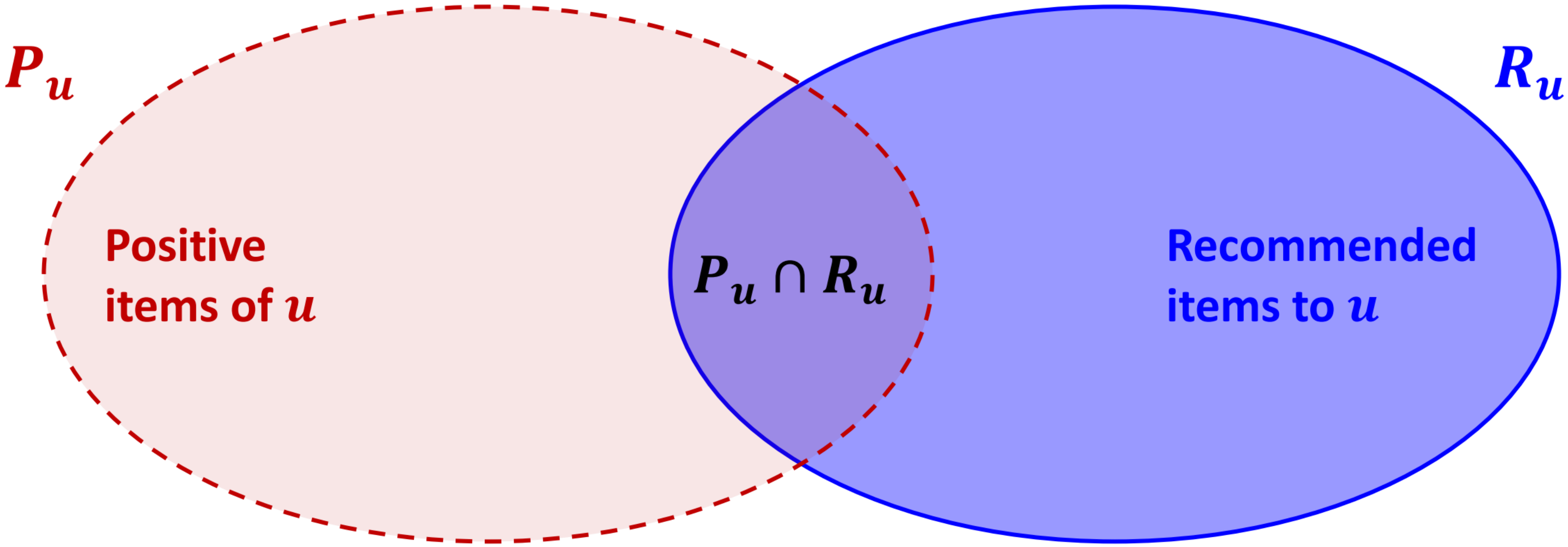
Evaluation Metrics: Precision, Recall, and F1

- **For each user *u*,**
  - Let P_u be a set of positive items the user will interact in the future
- **Let R_u be a set of items recommended by the model**
  - In top-K recommendation, $|R_u| = K$
  - Items that the user has already interacted are excluded



Evaluation Metrics: Precision@K

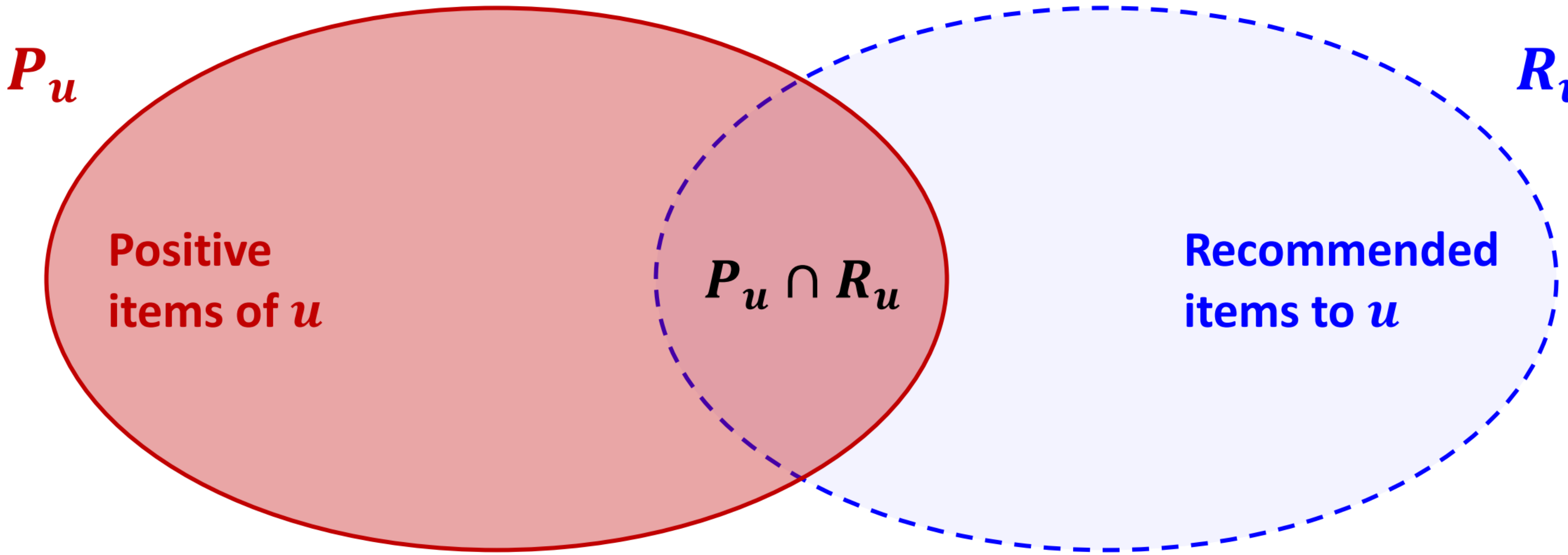
- **Precision@K for user u is $|P_u \cap R_u| / R_u$ (= K)**
  - Higher value indicates more positive items are recommended in top-K for user u



- **Final Precision@K is computed by averaging precision values across all users**

Evaluation Metrics: Recall@K

- **Recall@K for user *u* is $|P_u \cap R_u| / P_u$**
  - Higher value indicates more positive items are recommended in top-*K* for user *u*



- **Final Recall@*K* is computed by averaging recall values across all users**

Evaluation Metrics: F1@K

- **F1 Score**

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

- **Final F1@K is computed by averaging the F1 score values across all users**

Evaluation Metrics: NDCG@K

❑ Normalized discounted cumulative gain (NDCG)

- Reflect the *importance of ranked positions* of items in a set of top-K items

$$NDCG@K = \frac{1}{|U|}\sum_{u \in U} \frac{DCG_u@K}{IDCG_u@K}, \qquad DCG_u@K = \sum_{k=1}^{K} \frac{2^{y_k} - 1}{log_2(k+1)}$$

  - ❑ $y_k \in \{0,1\}$: a binary variable for the *k-th item i_k in a set R_u* of top-K items
    - $y_k = 1$ if $i_k \in P_u$ (i.e., a set of items considered relevant to the user u)
    - Otherwise, $y_k = 0$
  - ❑ $IDCG_u@K$: an *ideal* DCG at K
    - For *every item* i_k in R_u, y_k is set to 1

Evaluation Metrics: MRR@K

❑ Mean reciprocal rank (MRR)

- Reflect the average inversed rankings of *first relevant items* in top-K items

$$MRR@K = \frac{1}{|U|}\sum_{u \in U}\frac{1}{rank_u}$$

  - ❑ $Rank_u$: the *rank position of the first relevant item* in a set of top-K items R_u

Outline

- ❑ **Preliminary of Recommendation**
- ❑ **Recommender Systems Overview**
- ❑ **Data Imputation for Improving the Accuracy**
- ❑ **Summary**

Motivation

- CF approaches focus on **only the ratings given by users**
  - *Data sparsity problem*: most users have evaluated only a few items
  - There are only a few ratings in a rating matrix (< 4%)
- CF approaches suffer from low accuracy and coverage

Items

Users	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}	i_{11}	i_{12}
u_1		1		1		5	3			5		
u_2	1	3		4		4				2	4	
u_3												5
u_4		3		4		4		4	3			
u_5			4									
u_6	1			3	5	1		4			3	

A rating on i_1 given by user u_6

Data Imputation: Using Unrated Items

- **Exploit uncharted unrated items**
  - A fraction of rated items in a rating matrix R is extremely small (< 4%)
  - *Exploiting the vast number of "unrated" items* in R can lead to a *significant improvement* in CF approaches

Items

Users	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}	i_{11}	i_{12}
u_1		1		1		5	3			5		
u_2	1	3		4		4				2	4	
u_3												5
u_4		3		4		4		4	3			
u_5			4									
u_6	1			3	5	1		4			3	

unrated items

Data Imputation: Using Unrated Items

- **Unrated items**
  - Users were not aware of their existence
    - Candidates for recommendation
  - *Users knew but did not like and thus did not purchase*
    - *Uninteresting items*
- **Uninteresting items (of a user)**
  - Items on which the user has *"negative"* preferences

Users' Preferences: Two New Notions

- **Pre-use preferences**
  - User's impression on items **before** *purchasing and using them*
  - Determined *via meta data of items (known before purchasing)*
- **Post-use preferences**
  - User's impression on items **after** *purchasing and using them*
  - Determined *via real content of items (unknown before purchasing)*

Pre-Use/Post-Use Preferences and Ratings

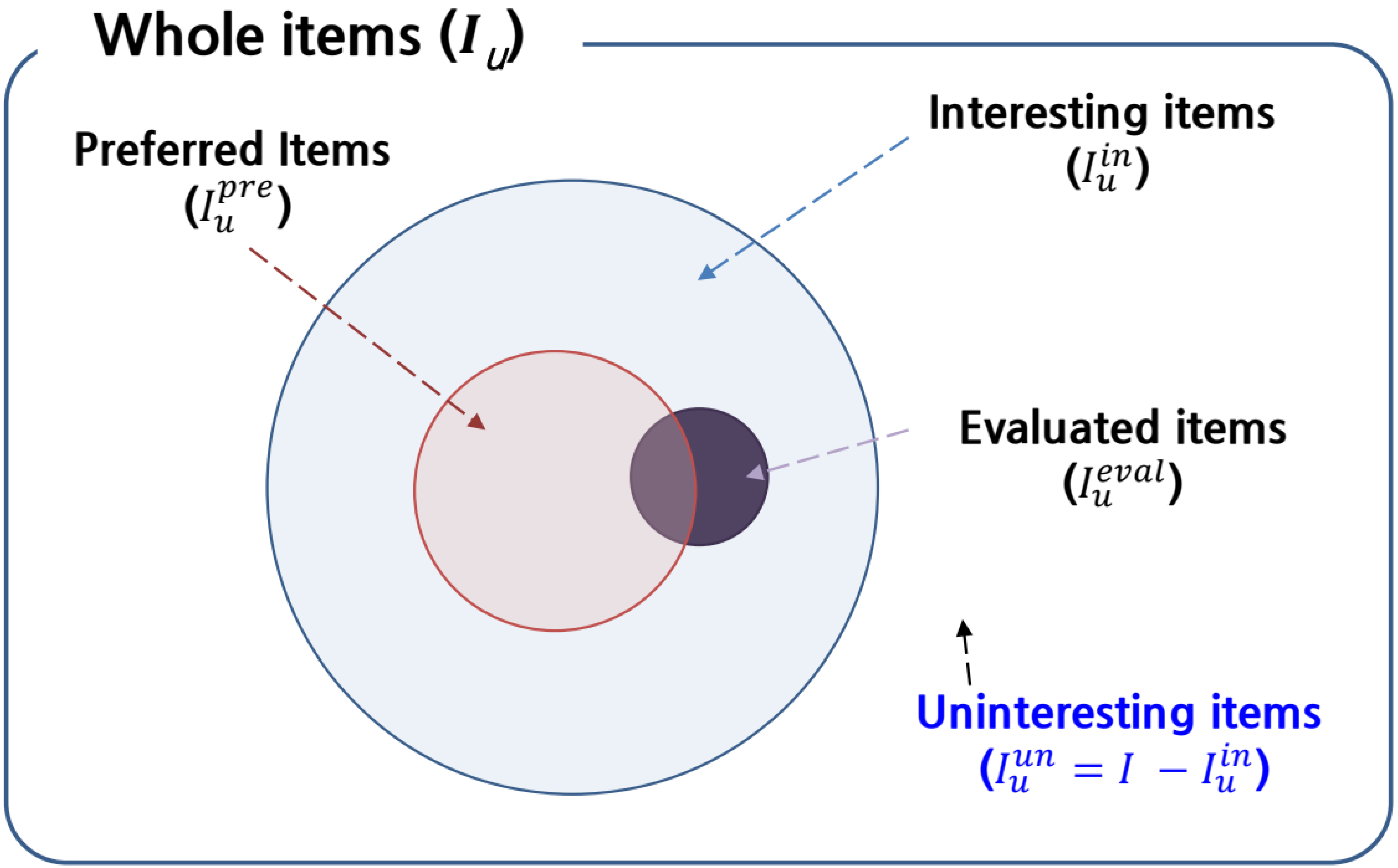
- **A user has high pre-use preference for Movie #1 and Movie #2**
  - The user likes Movie #1 but is disappointed at Movie #2
- **A user has low pre-use preference for Movie #3**

Movie	Pre-use preference	Post-use preference	Rating
Movie#1	High	High	5
Movie#2	High	Low	1
Movie#3	Not high	Unknown	Unrated

- **In this case**
  - *Uninteresting items*: Movie #3
  - Interesting items: Movie #1 (*preferred*) and Movie #2 (not preferred)

Venn Diagram for Preferences of Items

- Entire set of items that an active user thinks



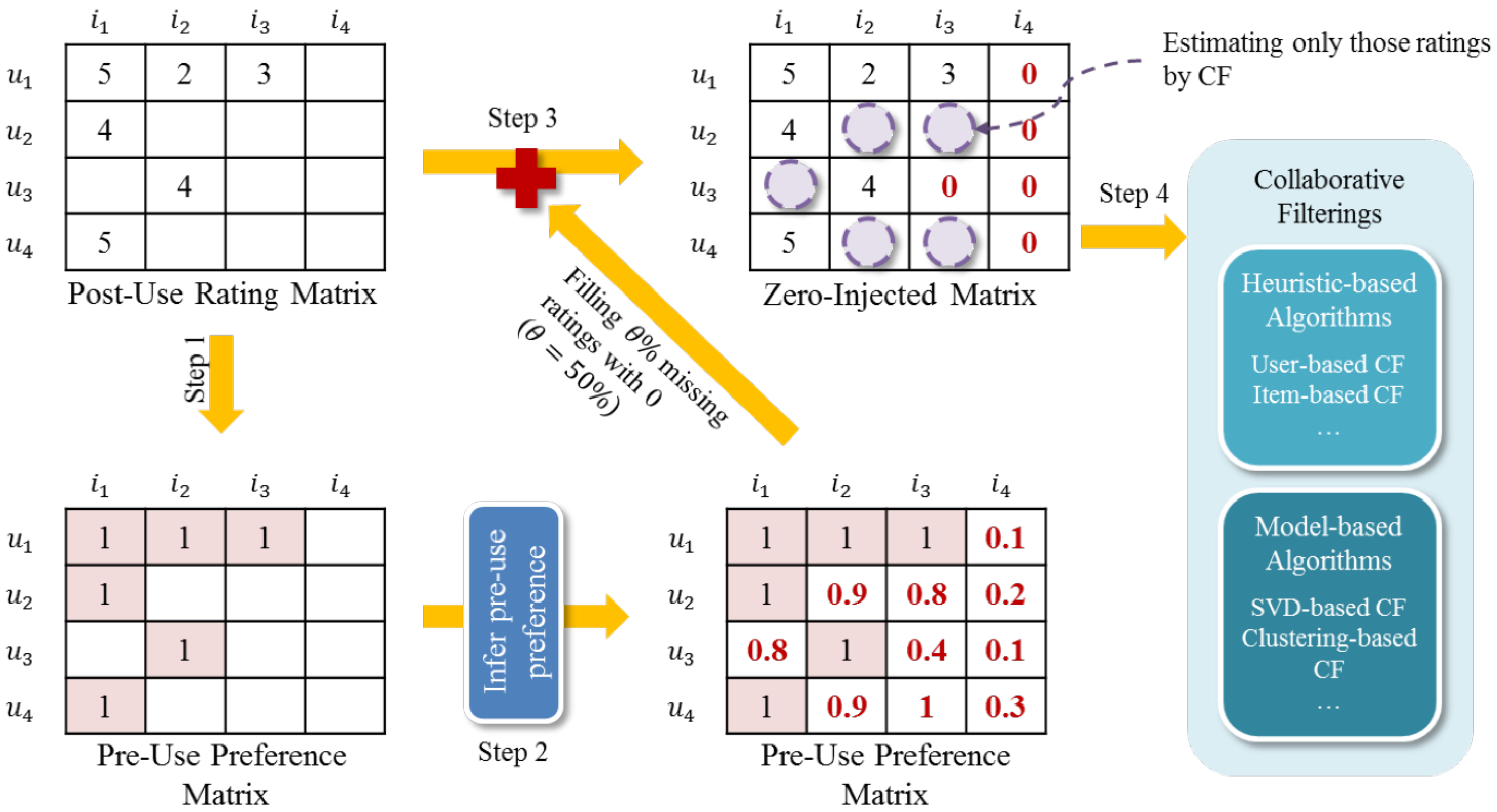
Pre-Use Preference and Uninteresting Items

- **Challenge: to identify uninteresting items among unrated items**
- **A user's uninteresting items**
  - Her/His pre-use preferences on them are *relatively low*
- **How to know a user's pre-use preferences**
  - *For all "rated" items: highest pre-use preferences*
    - Otherwise, users would not have bought them in the first place
  - *For unrated items: pre-use preferences need to be inferred*
    - Based on pre-use preferences on rated items

Exploiting Uninteresting items

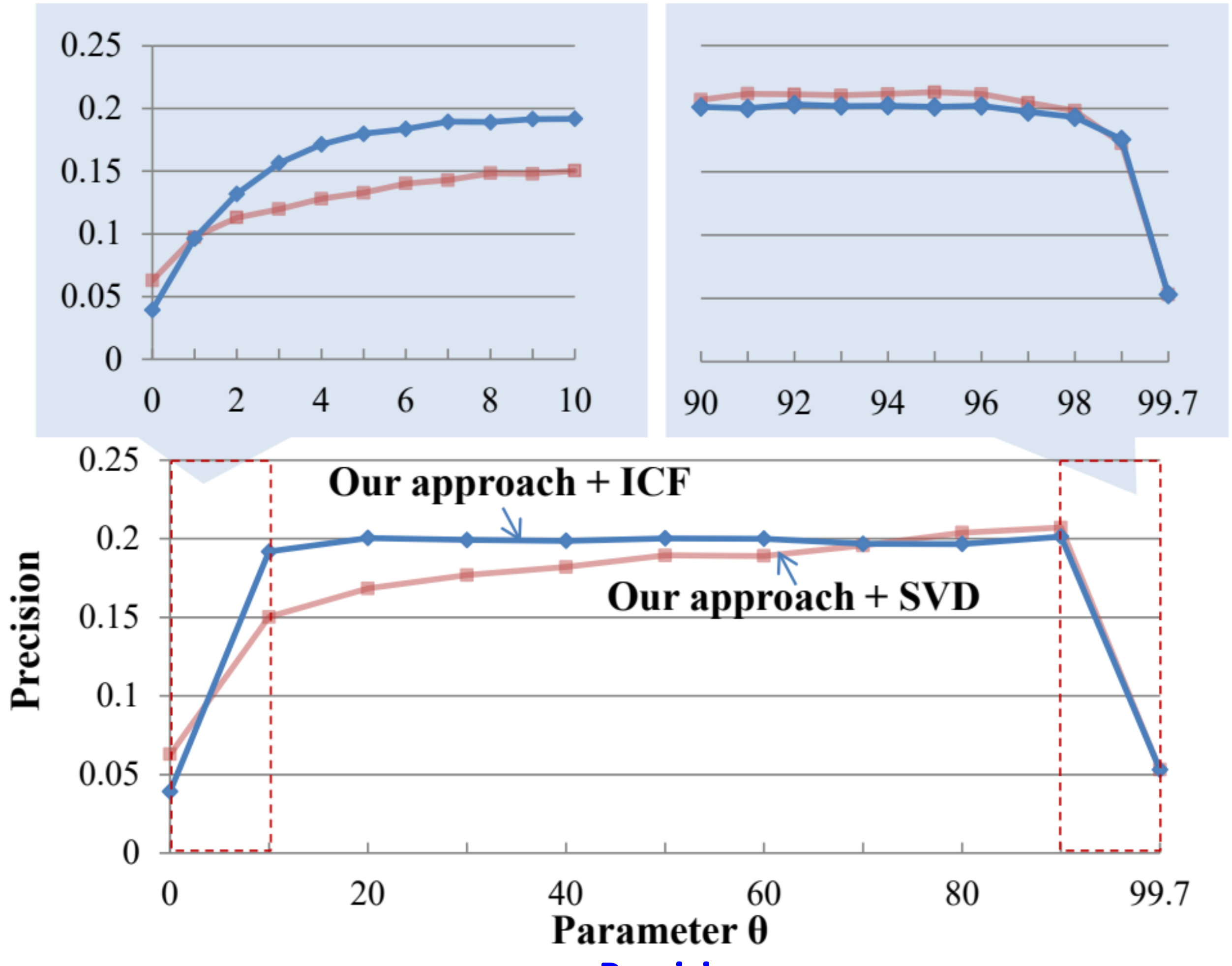
- **Final goal**
  - Identify *top-K (preferred) items*
    - They are interesting items (i.e., with high pre-use preferences)
    - Their post-use preferences are higher than others
- **Two strategies with uninteresting items**
  - *Strategy 1: to exclude* uninteresting items from the final recommendation list
  - *Strategy 2: to exploit* both uninteresting items and ratings in CF

Zero-Injection

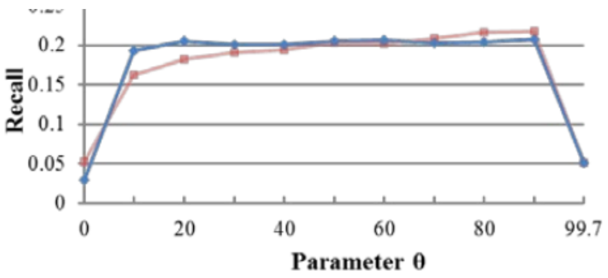


Effectiveness

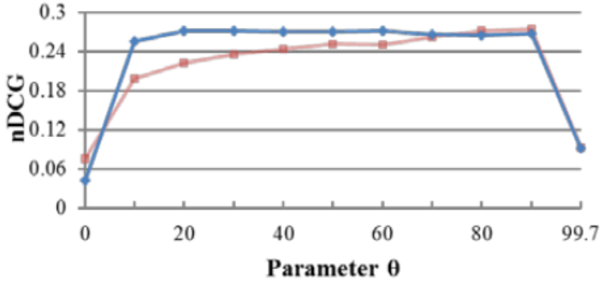
- Accuracy of ICF and SVD equipped with our zero-injection under varying parameter θ



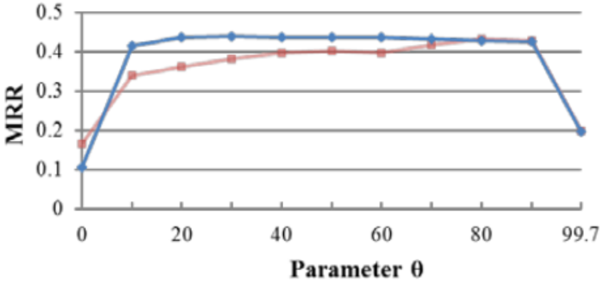
<Precision>



<Recall>



<nDCG>



<MRR>

Effectiveness

- Accuracy of four CF methods equipped with zero-injection ($\theta\theta$ = 90%)

Metric		ICF			SVD			SVD++			PureSVD		
		Orginal	Ours	Gain	Orginal	Ours	Gain	Orginal	Ours	Gain	Orginal	Ours	Gain
Precision	@5	**0.039**	**0.201**	**413.8%**	**0.063**	**0.207**	**229.7%**	**0.076**	**0.193**	**153.3%**	**0.100**	**0.106**	**16.7%**
	@10	0.041	**0.161**	**292.6%**	0.056	**0.166**	**196.9%**	0.069	**0.154**	**123.9%**	0.082	**0.089**	**19.1%**
	@15	0.040	**0.137**	**243.7%**	0.053	**0.142**	**169.9%**	0.063	**0.134**	**112.0%**	0.071	**0.078**	**11.3%**
	@20	0.039	**0.121**	**211.7%**	0.048	**0.125**	**159.1%**	0.058	**0.118**	**102.3%**	0.063	**0.071**	**13.7%**
Recall	@5	0.030	**0.207**	**600.3%**	0.052	**0.218**	**316.0%**	0.063	**0.194**	**209.6%**	0.112	**0.120**	**16.9%**
	@10	0.059	**0.305**	**412.7%**	0.089	**0.325**	**265.9%**	0.109	**0.288**	**163.1%**	0.175	**0.191**	**19.3%**
	@15	0.085	**0.375**	**341.4%**	0.121	**0.394**	**226.3%**	0.150	**0.361**	**141.2%**	0.220	**0.245**	**11.4%**
	@20	0.111	**0.428**	**285.4%**	0.144	**0.450**	**213.5%**	0.184	**0.415**	**125.3%**	0.254	**0.293**	**15.4%**
nDCG	@5	0.043	**0.268**	**527.9%**	0.076	**0.274**	**260.7%**	0.087	**0.256**	**196.0%**	0.135	**0.143**	**16.0%**
	@10	0.053	**0.285**	**436.0%**	0.084	**0.297**	**252.0%**	0.099	**0.272**	**175.6%**	0.151	**0.162**	**17.6%**
	@15	0.062	**0.303**	**390.7%**	0.094	**0.315**	**234.8%**	0.110	**0.291**	**163.7%**	0.164	**0.178**	**18.9%**
	@20	0.071	**0.319**	**351.7%**	0.101	**0.332**	**227.3%**	0.121	**0.306**	**153.5%**	0.174	**0.193**	**10.9%**
MRR		0.106	**0.426**	**303.0%**	0.165	**0.428**	**159.2%**	0.181	**0.416**	**129.3%**	0.262	**0.274**	**14.7%**

Outline

- ❑ Preliminary of Recommendation
- ❑ Recommender Systems Overview
- ❑ Data Imputation for Improving the Accuracy
- ❑ Summary

Summary

- Recommender systems (RecSys.)
  - **Content-based:** recommending those items that have *similar contents* to those of the active user's favorite items
  - **Collaborative filtering (CF):** recommending items *related high by neighbors* who have preferences similar to that of the active user
  - **Hybrid:** recommending items by *integrating the approaches above*
- CF techniques
  - **Heuristics-based:** measure similarity between a target user and users, and then predict a rating by aggregating similar users' rating
  - **Machine-learning-based:** build a ML/DL model and then predict a rating through the model

Summary

- **Evaluation of RecSys.**
  - **Overall procedure:** partition, train, optimize, and measure
  - **Accuracy metrics**: Precision, Recall, F1, NDCG, and MRR
- **Improving the Accuracy of RecSys.**
  - **Data imputation:** estimating and filling in missing values in the rating matrix
  - **Pre-use preference:** user's impression on items *before purchasing and using them*
  - **Post-use preference:** user's impression on items *after purchasing and using them*

Q&A:
Recommender Systems