Recommender System 고급기술

Heung-Nam Kim, Ph. D

Convergence Center, LG Electronics

2016. 12. 03

쉽게 배우는 인공지능 – 제1편 (딥러닝 및 추천기술)

인공지능 연구회 워크샵

Outline

- Context in Recommender Systems

  i.e., Context-aware Recommender Systems (CARS)
  - Context definition, Paradigms in CARS
  - Context-incorporated Modeling and Techniques: Algorithms

- Trust in Recommender Systems

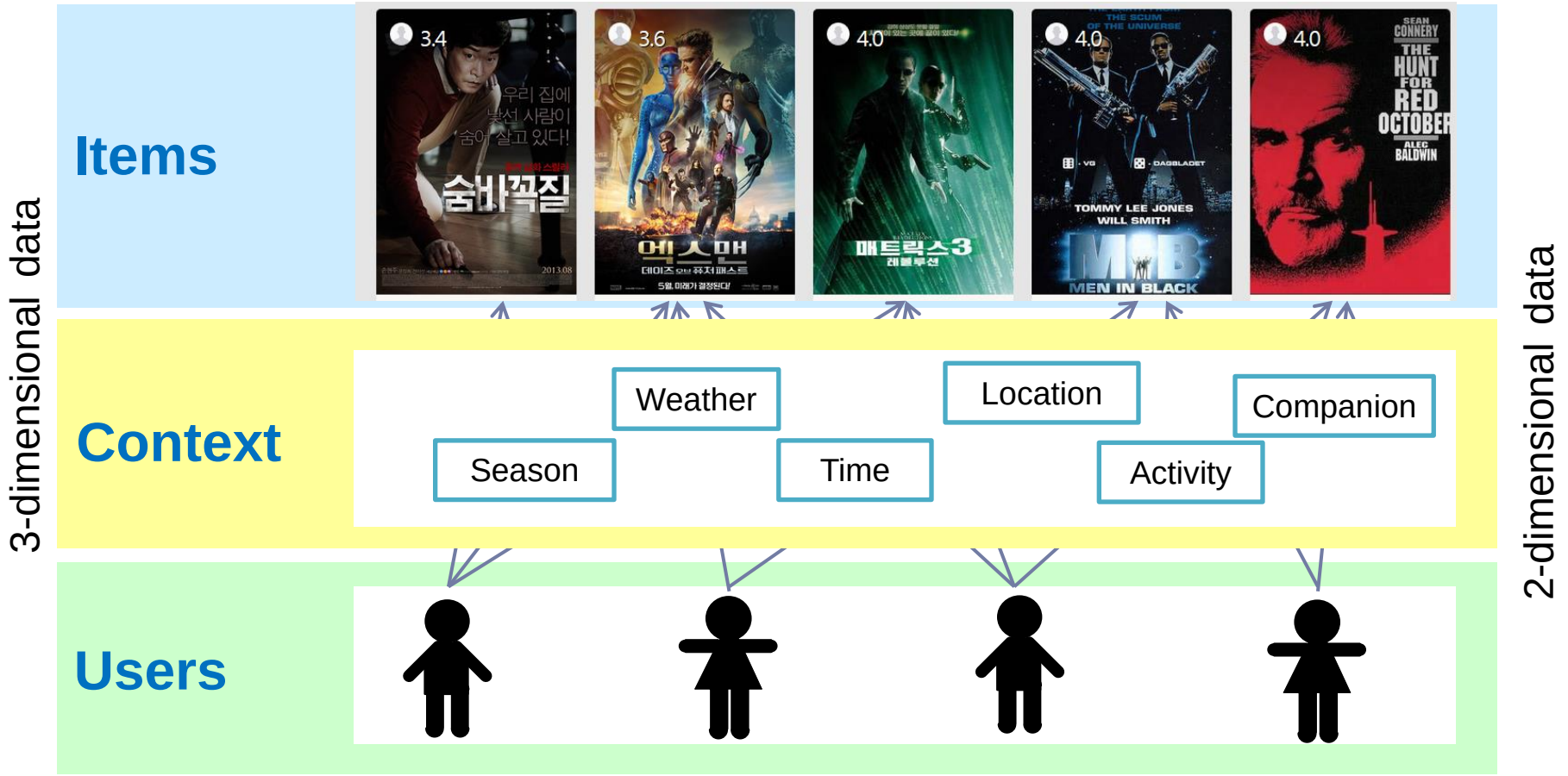
  i.e., Trust-aware Recommender Systems (TARS)
  - Trust definition, paradigms in TARS
  - Trust-incorporated Modeling and Techniques: Algorithms

- Online Resources
  - CARS Library
  - TARS Library
  - Data Sets for CARS/TARS

Section 1. CARS
Context-Aware Recommender Systems

Context-Aware Recommender Systems (CARS)

- Traditional RecSys: Users x Items → Ratings
- Context-Aware Recsys: Users x Items x Context → Ratings



What is Context?

"Context is any information that can be used to characterize the situation of an entity."

(*AnindK. Dey, 2001*)

- **Representative Context** vs. Interactive Context :
  - Fully Observable and Static vs. Non-fully observable and Dynamic

- Context for CARS: Observed Context

  Contexts are those variables which may change when a same activity is performed again.

  Contexts represents a set of explicit variables that model contextual factors in the underlying domain (e.g., time, location, weather, occasion, etc.)

  (Adomavicius, Mobasher, AI Magazion, 2011)

- Not all contextual information is relevant for recommendations
  - Determining relevance of context information: Manually vs. Automatically

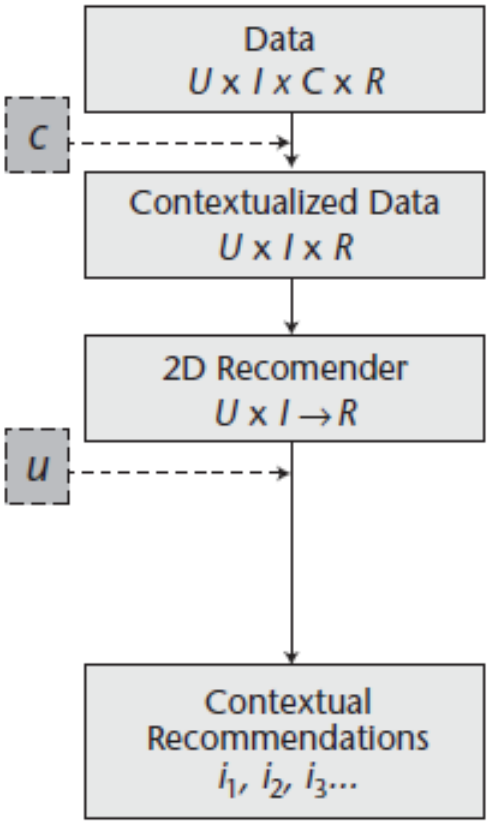
    Domain experts Feature selection, statistical tests

CARS Architectural Models (Adomavicius & Tuzhilin, RecSys, 2008)

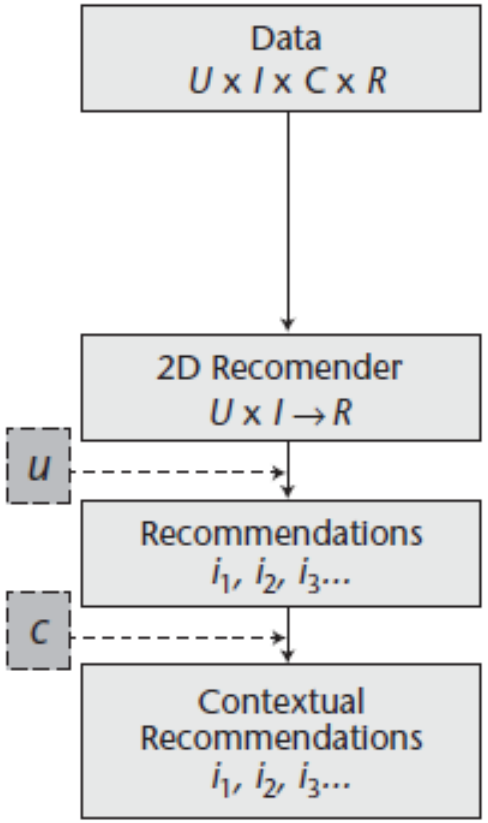
- Three types to build algorithms for CARS
  1. **Contextual Pre-filtering**
     contextual information to select or construct the most relevant 2D data (i.e., user-item) for recommendations
     e.g., *Splitting Approaches* (Zheng et al.,SAC 2014)
  2. **Contextual Post-filtering**
     Contextual information is used to filter/constrain/re-rank final set of recommendations
     e.g., *Weight and Filter* (Panniello et al.,RecSys 2009)
  3. **Contextual Modeling**
     contextual information is used directly in the modeling technique as part of rating estimation
     e.g., *Tensor Factorization* (Karatzoglou et al., RecSys 2010),
     Context-Aware Matrix Factorization (Baltrunas et al., RecSys 2011),
     Context-Aware Factorization Machines (Rendle et al. SIGIR 2011)

CARS Architectural Models

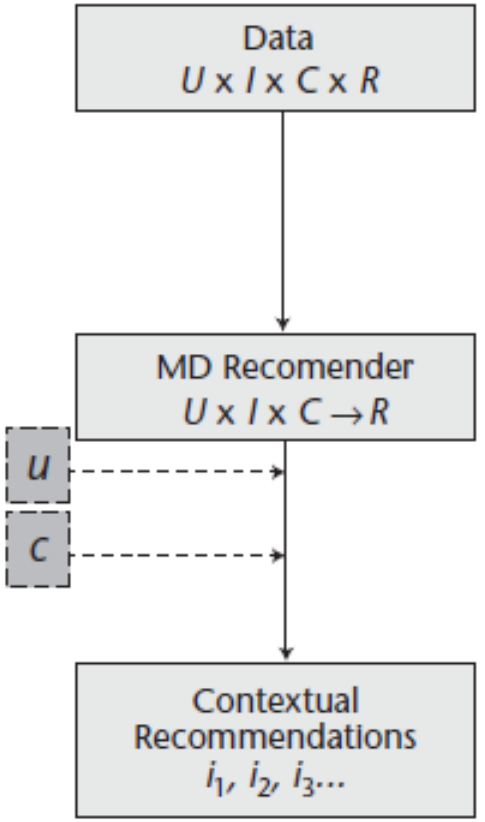
1. Contextual Pre-filtering



2. Contextual Post-filtering



3. Contextual Modeling



(from Adomavicius, Mobasher, AI Magazion, 2011)

Terminology in CARS

Contextual dimension

User	Item	Rating	Season	Location	Companion
U1	M1	5	Winter	Theater	Family
U2	M1	5	Winter	Theater	Family
U3	M2	4	Winter	DVD House	Friend
U1	M3	2	Fall	Home	Kids
U2	M2	3	Summer	Home	Friend
U1	M2	?	Summer	Home	Family

Contextual condition

Contextual situation

- **Contextual dimension** (factor): variables that model contextual information
  e.g., Season, Location, Companion
- **Contextual condition**: values of contextual factors
  e.g., Winter/Summer(Season), Theater/Home(Location), Family/Friend/Kids(Companion)
- **Contextual situation**: a set of context conditions
  e.g., {Winter, Theater, Family}, {Summer, Home, Friend}

Context-aware Splitting Approaches

- Item Splitting (Baltrunas et al.,RecSys 2009)
- User Splitting (Said et al., CARS@RecSys 2011)
- User and Item (UI) Splitting (Zheng et al., ACM SAC 2014)

Context-aware Splitting Approaches

- Goal: produce a 2D data set that incorporates context information associated with preference scores
  - *Pros*: can use a variety of well-known traditional recommendation algorithms in the modeling phase
  - *Cons*: determine splitting criteria
    lead to too much data sparsity

1. Item Splitting
   - contexts are *dependent on* ***items***
2. User Splitting
   - contexts are *dependent on* ***users***
3. User and Item Splitting
   - fuses dependent contexts to users and items simultaneously

Item Splitting (Baltrunas and Ricci, RecSys, 2009)

Depending on contextual conditions, consider *one item as two different items*.

User	Item	Season	Location	Companion	Rating
U1	M1	Winter	Theater	Family	5
U2	M1	Winter	Theater	Family	5
U3	M1	Winter	DVD House	Friend	4
U1	M1	Summer	Home	Friend	2
U2	M1	Summer	Home	Friend	3
U1	M1	Fall	DVD House	Alone	2

High rating

Low rating

Statistically significant difference ?
(e.g. t-test, z-test, chi-square test)

Item Splitting

Assuming **Season** (Winter vs. Non-Winter) is the best split condition

Item M1

M11:
M1 seen in winter

M12:
M1 seen not in winter

User-Item Ratings

User	Item	Rating
U1	M11	5
U2	M11	5
U3	M11	4
U1	M12	2
U2	M12	3
U1	M12	2

User Splitting (Said et al., CARS@RecSys, 2011)

Depending on contextual conditions, consider *one user as two different users*.

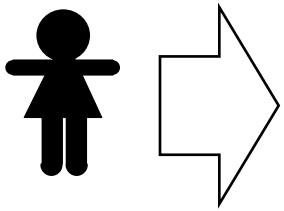
User	Item	Season	Location	Companion	Rating
U1	M1	Winter	Theater	Family	5
U2	M1	Winter	Theater	Family	5
U3	M1	Winter	DVD House	Friend	4
U1	M1	Summer	Home	Friend	2
U2	M1	Summer	Home	Friend	3
U1	M1	Fall	DVD House	Alone	2

High rating

Low rating

Statistically significant difference ?
(e.g. t-test, z-test, chi-square test)

User Splitting Assuming **Companion** (Family vs. Non-Family) is the best split condition



User U1

U11:
U1 saw the movie with family

U12:
U1 saw the movie alone or with a friend

User-Item Ratings

User	Item	Rating
U11	M1	5
U21	M1	5
U32	M1	4
U12	M1	2
U22	M1	3
U12	M1	2

User and Item (UI) Splitting (Zheng et al., ACM SAC, 2014)

Fusing contexts to both users and items (item splitting followed by user splitting on the resulting output

User	Item	Season	Location	Companion	Rating
U1	M1	Winter	Theater	Family	5
U2	M1	Winter	Theater	Family	5
U3	M1	Winter	DVD House	Friend	4
U1	M1	Summer	Home	Friend	2
U2	M1	Summer	Home	Friend	3
U1	M1	Fall	DVD House	Alone	2

(a) Item Splitting

User	Item	Rating
U1	M11	5
U2	M11	5
U3	M11	4
U1	M12	2
U2	M12	3
U1	M12	2

(b) User Splitting

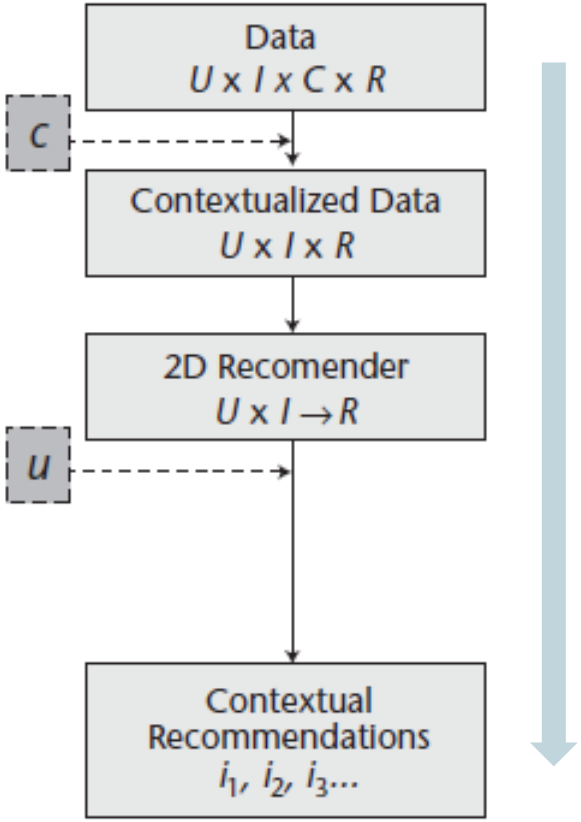
User	Item	Rating
U11	M1	5
U21	M1	5
U32	M1	4
U12	M1	2
U22	M1	3
U12	M1	2

(c) UI Splitting

User	Item	Rating
U11	M11	5
U21	M11	5
U32	M11	4
U12	M12	2
U22	M12	3
U12	M12	2

Recommendation Process

Contextual Pre-filtering



1. Find the best split to perform a splitting approach

2. Transforming 3D data to 2D rating data (user-item rating matrix)
   item splitting, user splitting, UI splitting

3. Apply any popular recommender algorithms

e.g., UserCF, ItemCF, PMF, FunkSVD, Biased MF, SVD++

Contextual Modeling

Context-Aware Matrix Factorization (CAMF)

(Baltrunas *et al. RecSys 2011*)

Biased Matrix Factorization (Koren, SIGKDD, 2008)

- Adding **biases** to Matrix Factorization (MF)
  - tendencies for some users to give higher ratings than others **user bias**
  - tendencies for some items to receive higher ratings than others **item bias**

Prediction using Biased MF

$$r'_{ui} = \mu + b_u + b_i + \mathbf{p}_u \mathbf{q}_i^{\mathrm{T}}$$

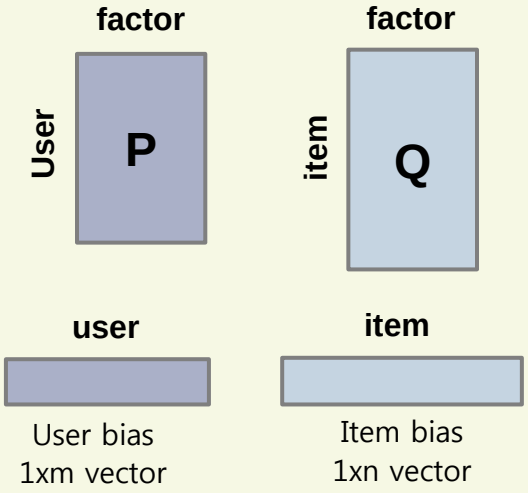
Global average | User bias | Item bias | User-item interaction

Minimizing the squared error function

$$\min_{b*,p*,q*} \sum_{r_{ui} \in \mathbf{R}} \underbrace{(r_{ui} - r'_{ui})^2}_{\text{error}} + \underbrace{\lambda\left(b_u^2 + b_i^2 + \|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2\right)}_{\text{Regularization}}$$

Learning Models

with Stochastic Gradient Descent



Context-Aware Matrix Factorization (CAMF)

- Extension of **Biased MF** for considering contextual information in generating recommendations
- Three models derived based on the interaction of context and user/items

1. **CAMF_C**: Assumes contextual effect is associated with each contextual condition only (*global influence* of each contextual condition on the ratings)

2. **CAMF_IC**: Assumes contextual effect is associated with item-context interactions (*each contextual condition and item pair*)

3. **CAMF_UC**: Assumes contextual effect is associated with user-context interactions (*each contextual condition and user pair*)

Contextual Rating Biases (Deviation)

- **Contextual Rating Biases** (Deviation)

  how users' rating is deviated from context c1 to c2?

Context	D_1:Season	D_2:Location	D_3:Companion
C_1	Winter	Theater	Family
C_2	Summer	Home	Friend
$CBias(c_j)$	-0.2	0.5	-0.1

Users' rating in **Winter** is generally lower than users' rating in **Summer** by 0.2 **$CBias(D_1) = -0.2$**

Users' rating in **Theater** is generally higher than users' rating at **Home** by 0.5 **$CBias(D_2) = 0.5$**

Users' rating with **Family** is generally lower than users' rating with **Friend** by 0.1 **$CBias(D_3) = -0.1$**

(*Tutorial: Context In Recommender Systems, ACM SAC, 2016*)

Contextual Rating Biases (Deviation)

- Assume ∅ (NULL) context: without considering context

Context	D_1:Season	D_2:Location	D_3:Companion
∅	*unknown*	*unknown*	*unknown*
C_2	Summer	Home	Friend
CBias(c_j)	-0.2	0.5	-0.1

Assume a predicted rating of user *u* for item *i* in context ∅ is 4, *i.e.*,

$$r'_{ui\emptyset} = r'_{ui} = 4$$

Predict a rating of user *u* for item *i* in ***context*** C_2

$$r_{uiC_2} = 4 - 0.2 + 0.5 - 0.1 = 4.2$$

$$r'_{uiC} = r'_{ui} + \sum_{j=1}^{k} CBias(c_j) \qquad whre\ C = \{c_1, c_2, \dots, ck\}$$

CAMF Model with Context Biases

BiasedMF

$$r'_{ui} = \mu + b_u + b_i + \mathbf{p}_u \mathbf{q}_i^{\mathrm{T}}$$

Global average | User bias | Item bias | User-item interaction

Assumes contextual effect is associated with each contextual condition only (*global influence* of each contextual condition on the ratings)

CAMF_C Prediction

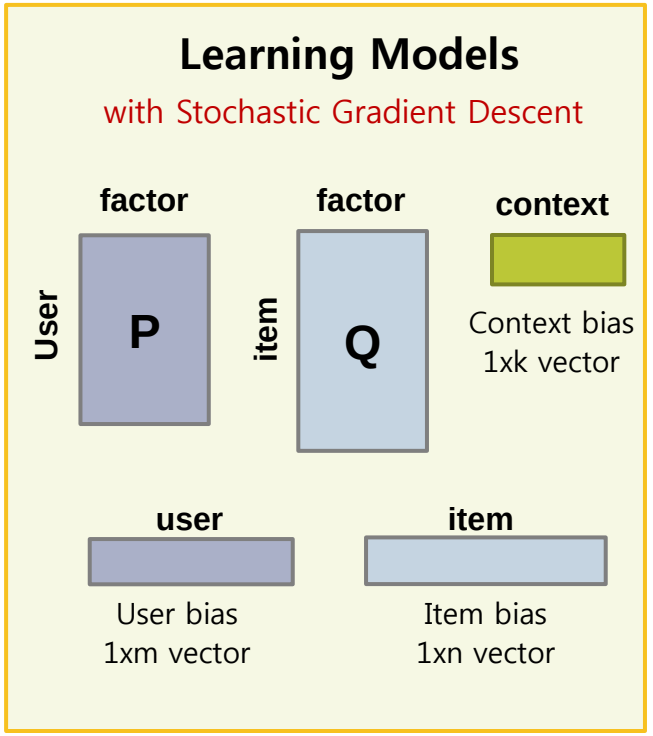
$$r'_{uic_1c_2\ldots ck} = \mu + b_u + b_i + \sum_{j=1}^{k} CBias(c_j) + \mathbf{p}_u \mathbf{q}_i^{\mathrm{T}}$$

Minimizing objective function

$$\sum_{r \in \mathbf{R}} (r_{uiC} - r'_{uiC})^2 + \lambda \left(b_u^2 + b_i^2 + \|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2 + \sum_{j=1}^{k} CBias(cj)^2 \right)$$

Regularization

$whre\ C = \{c_1, c_2, \ldots, ck\}$



CAMF Model with Item-Context Biases

BiasedMF

$$r'_{ui} = \mu + b_u + b_i + \mathbf{p}_u \mathbf{q}_i^{\mathrm{T}}$$

Global average | User bias | Item bias | User-item interaction

Assumes contextual effect is associated with item-context interactions (*each contextual condition and item pair*)

CAMF_IC Prediction

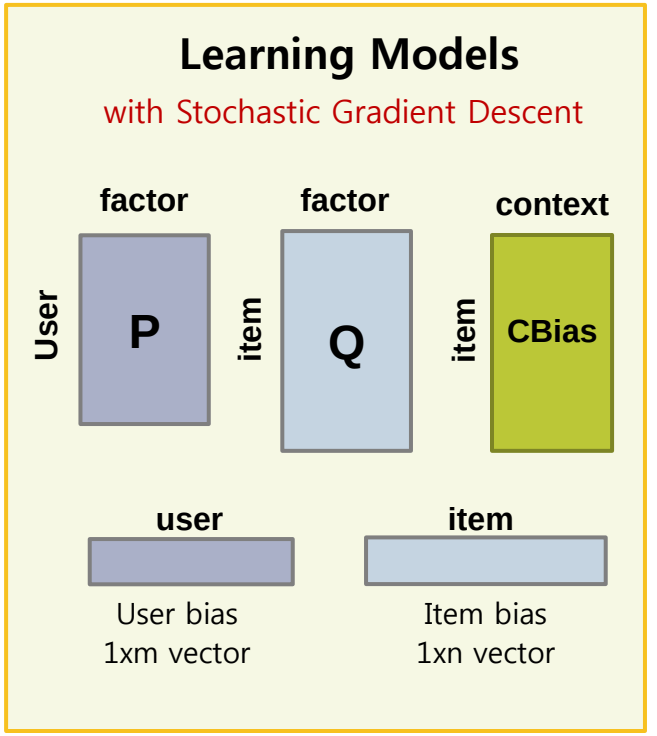
$$r'_{uic_1c_2\ldots ck} = \mu + b_u + \sum_{j=1}^{k} CBias(i, c_j) + \mathbf{p}_u \mathbf{q}_i^{\mathrm{T}}$$

Minimizing objective function

$$\sum_{r \in \mathbf{R}} (r_{uiC} - r'_{uiC})^2 + \lambda \left(b_u^2 + \|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2 + \sum_{j=1}^{k} CBias(i, cj)^2 \right)$$

Regularization

$whre\ C = \{c_1, c_2, \ldots, ck\}$



CAMF Model with User-Context Biases

BiasedMF

$$r'_{ui} = \mu + b_u + b_i + \mathbf{p}_u \mathbf{q}_i^{\mathrm{T}}$$

Global average | User bias | Item bias | User-item interaction

Assumes contextual effect is associated with user-context interactions (*each contextual condition and user pair*)

CAMF_UC Prediction

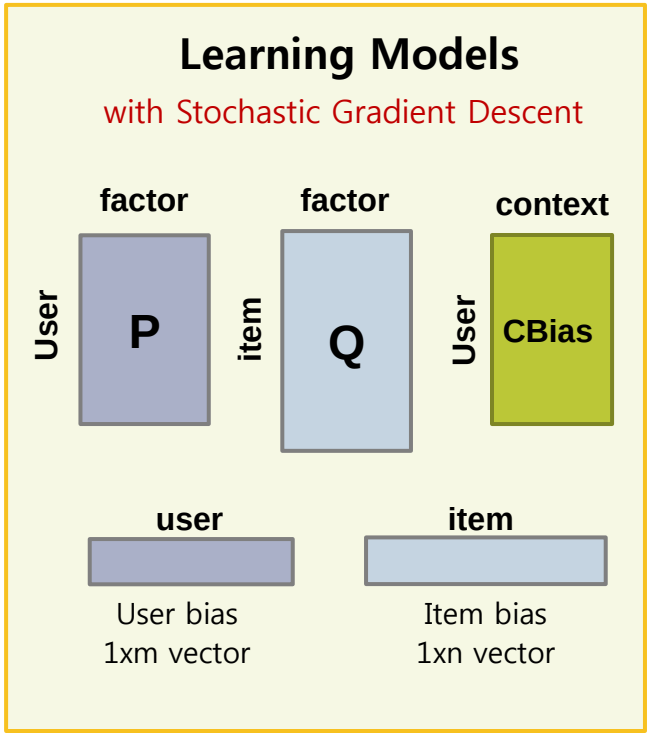
$$r'_{uic_1c_2\ldots ck} = \mu + \sum_{j=1}^{k} CBias(u, c_j) + b_i + \mathbf{p}_u \mathbf{q}_i^{\mathrm{T}}$$

Minimizing objective function

$$\sum_{r \in \mathbf{R}} (r_{uiC} - r'_{uiC})^2 + \lambda \left(b_i^2 + \|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2 + \sum_{j=1}^{k} CBias(u, cj)^2 \right)$$

Regularization

$whre\ C = \{c_1, c_2, \ldots, ck\}$



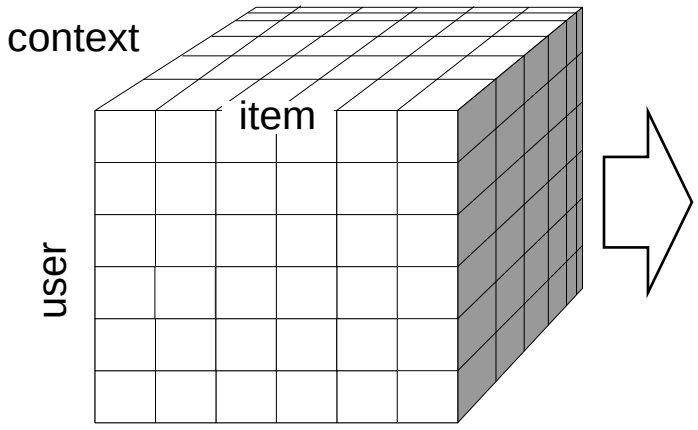
Contextual Modeling

Collaborative Latent Models

(***Heung-Nam Kim et al.***)

Tensor Factorization, But…

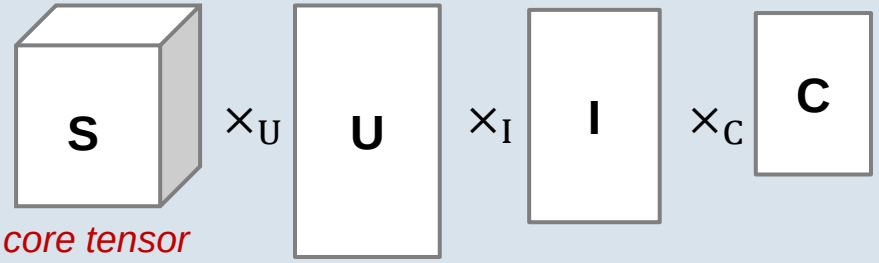
3-order Tensor



Users x Items x Contexts → Ratings

$\mathbf{Y} \in \mathbb{R}^{m \times n \times k}$

1) Tucker Decomposition (*Karatzoglou et al., RecSys 2010*)

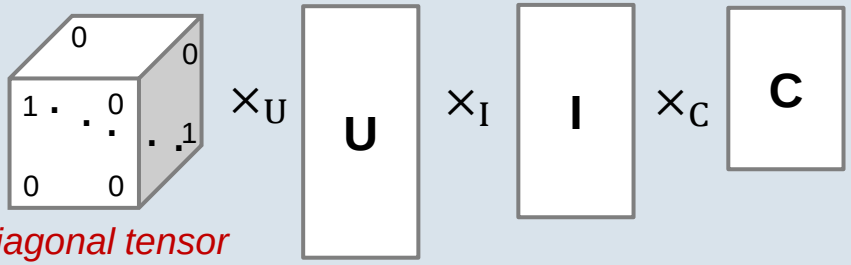


core tensor

$\mathbf{S} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ $\mathbf{U} \in \mathbb{R}^{m \times d_1}$ $\mathbf{I} \in \mathbb{R}^{n \times d_2}$ $\mathbf{C} \in \mathbb{R}^{k \times d_3}$

$$Y'_{uic} = \sum_v \sum_j \sum_b S_{vjb} \times U_{uv} \times I_{ij} \times C_{cb}$$

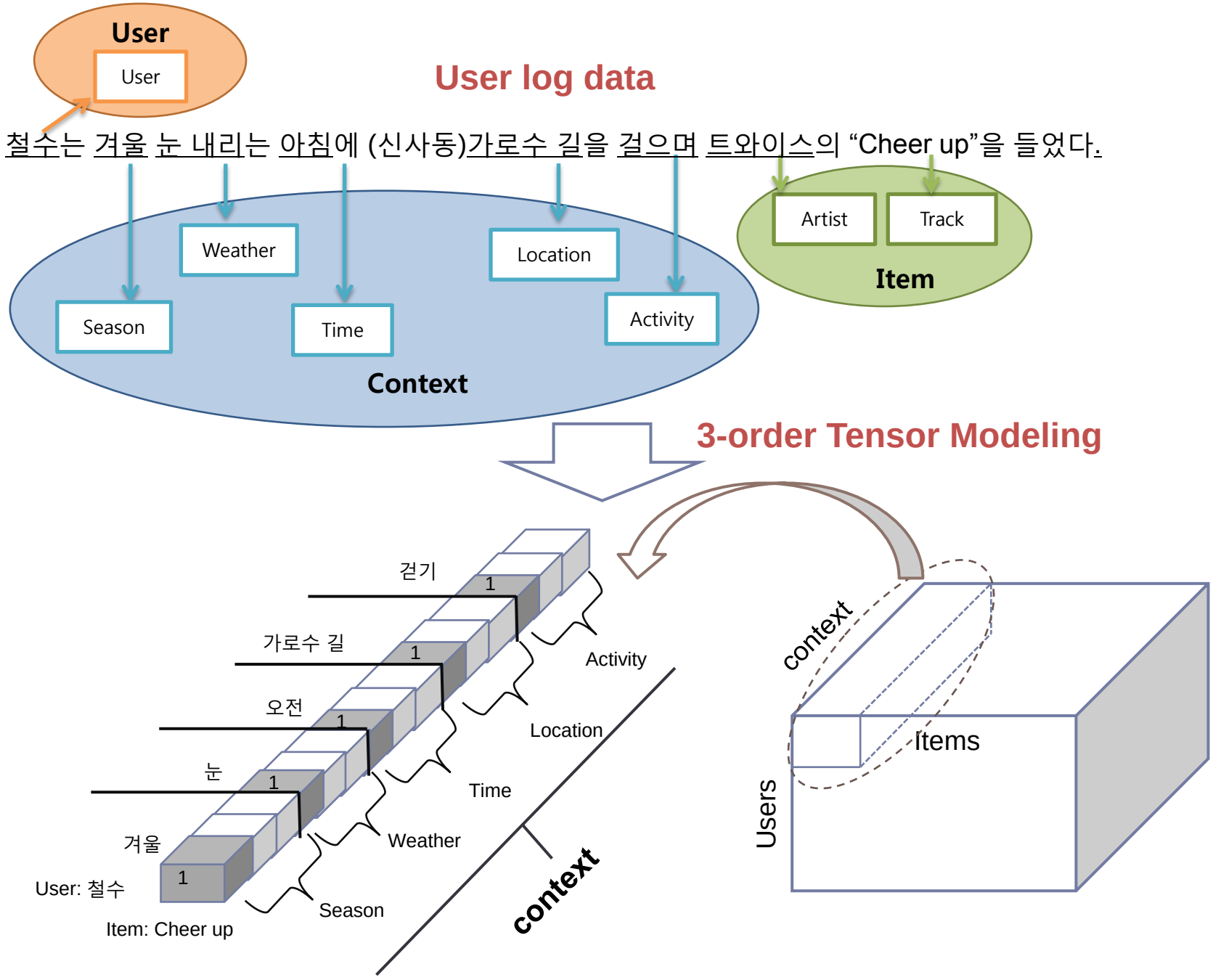
2) Canonical Decomposition (aka. *Parallel Factors Analysis*)



diagonal tensor

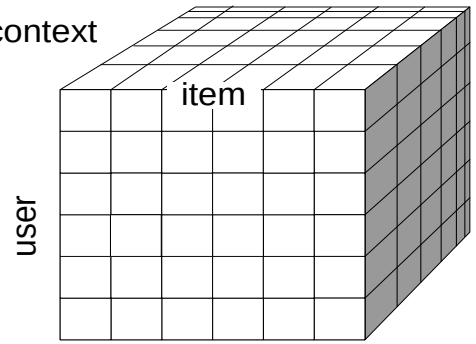
$\mathbf{S} \in \mathbb{R}^{d \times d \times d}$ $\mathbf{U} \in \mathbb{R}^{m \times d}$ $\mathbf{I} \in \mathbb{R}^{n \times d}$ $\mathbf{C} \in \mathbb{R}^{k \times d}$

$$Y'_{uic} = \sum_{j=1}^{d} U_{uj} \times I_{ij} \times C_{cj}$$


User
User
User log data
철수는 겨울 눈 내리는 아침에 (신사동)가로수 길을 걸으며 트와이스의 “Cheer up”을 들었다.
Weather
Location
Artist
Track
Season
Time
Activity
Item
Context
3-order Tensor Modeling
걷기
가로수 길
오전
눈
겨울
User: 철수
Item: Cheer up
1
Activity
Location
Time
Weather
Season
context
context
Users
Items

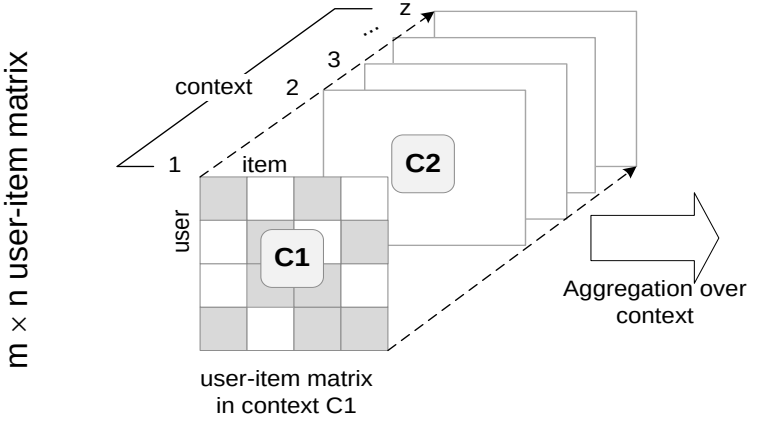
① user-item matrix **R**,
② user-context matrix **M**,
③ context-item matrix **N**

3-order tensor

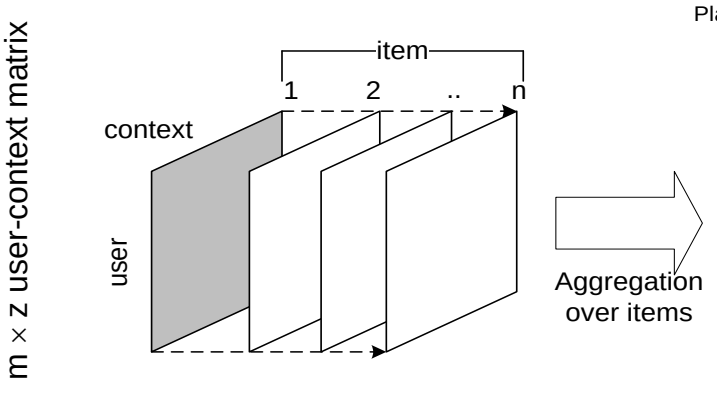


3개의 2-dimensional data

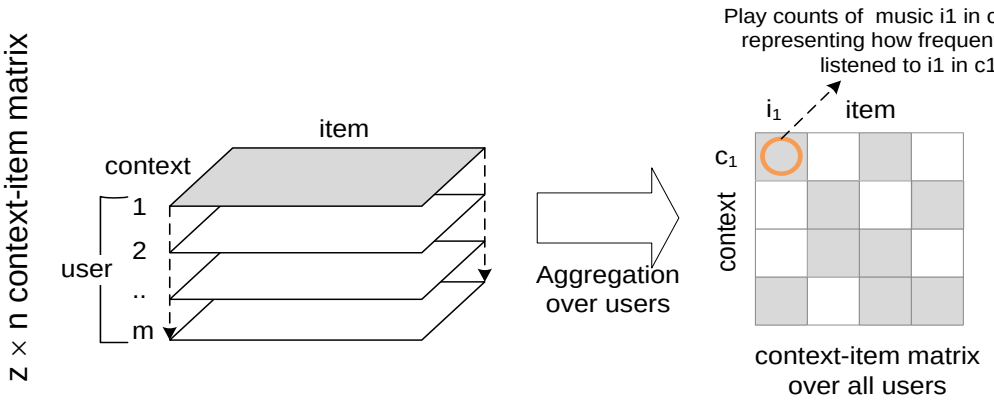
1



2



3



Collaborative Latent Models

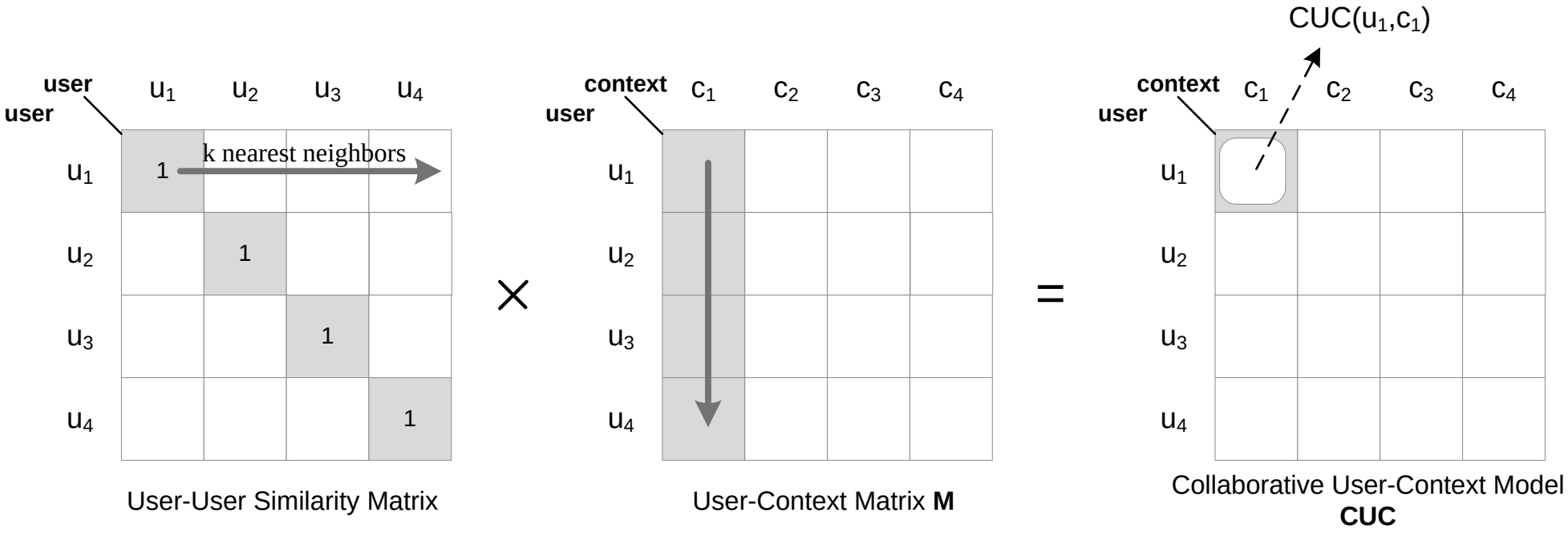
- 기존의 3개의 matrix 정보 기반으로 Collaborative Filtering 개념을 적용, 새로운 User-Context 관계와 Context-Item 관계를 유도함

1. **Collaborative User-Context Model (CUC):**
   User에 대한 Context의 잠재적 선호도를 Collaborative filtering 관점에서 수치화

2. **Collaborative Context-Item Model (CCI):**
   Context에 대한 Item의 잠재적 적합성을 Collaborative filtering 관점에서 수치화

Collaborative User-Context Model

- Collaborative User-Context 모델은 각 행에 KNN만을 포함하고 있는 user-user similarity matrix과 user-context matrix **M**의 두 행렬 곱으로 구축

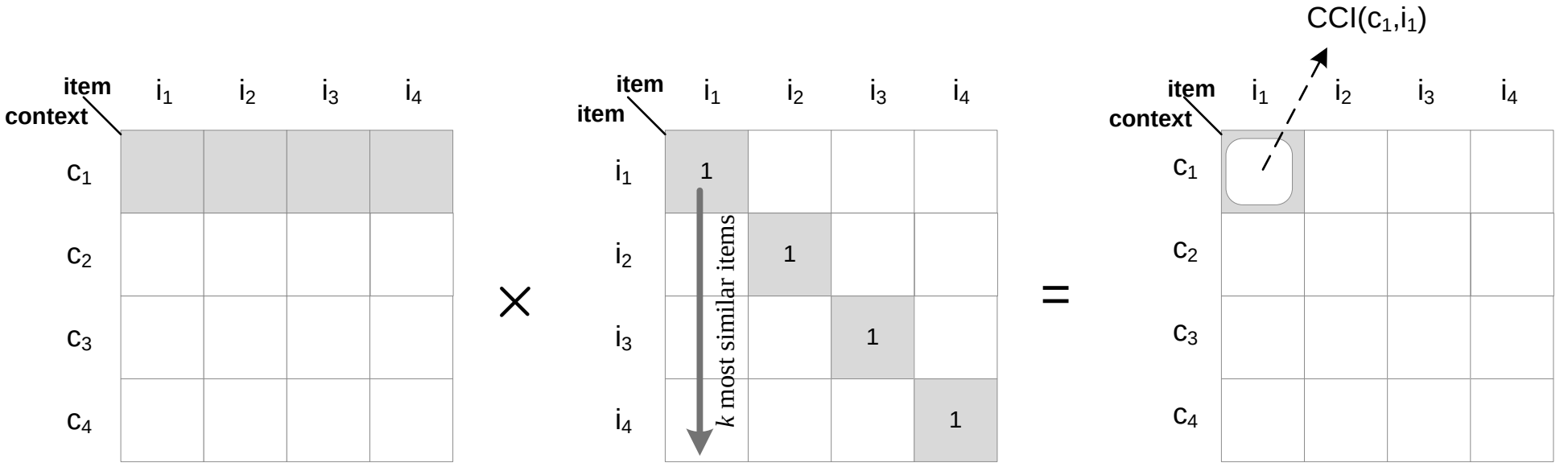


k nearest neighbors (KNN)

1. 음악적 취향의 유사성: 두 사용자가 얼마나 동일한 음악들을 많이 들었는지 (user-item matrix **R**)
2. 음악을 듣는 context의 유사성: 두 사용자가 얼마나 같은 상황에서 음악을 자주 들었는지 (user-context matrix **M**)

Collaborative Context-Item Model

- Collaborative Context-item 모델은 각 열에 KMI만을 포함하고 있는 item-item similarity matrix과 context-item matrix **N**의 두 행렬 곱으로 구축



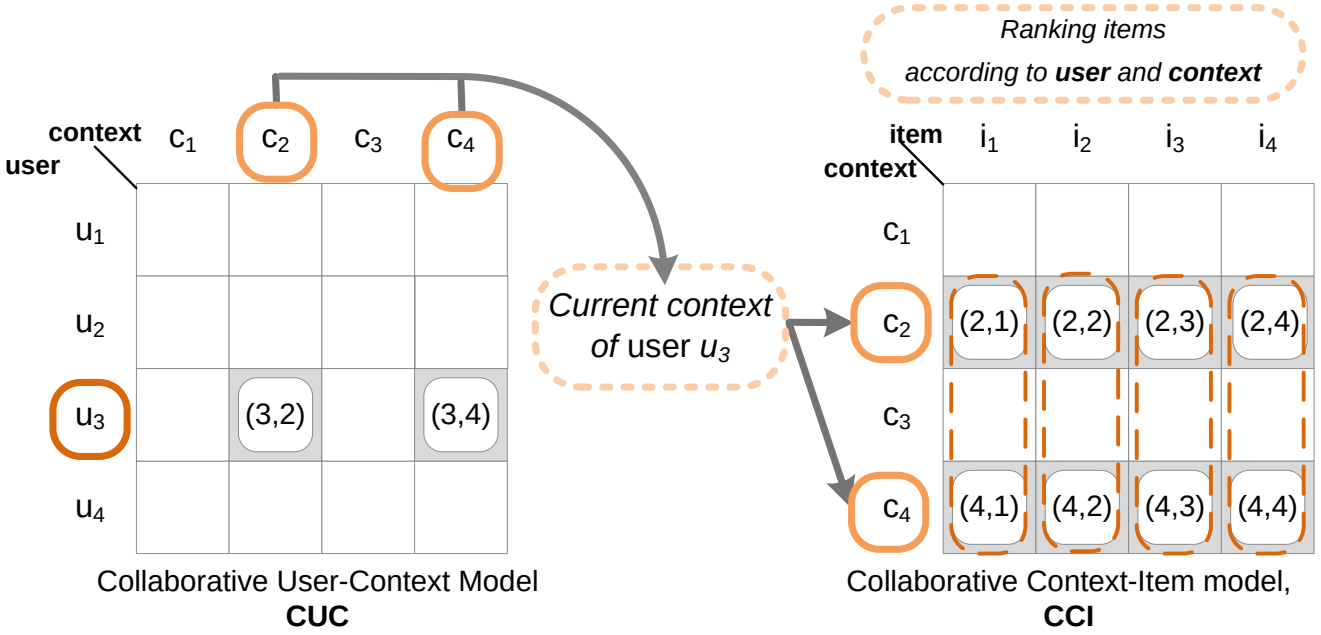
k most similar items (KMI)

1. 사용자 music play 패턴의 유사성: 두 아이템이 얼마나 동일한 사용자에게 같이 많이 play되었는지 (user-item matrix **R**)
2. 음악을 듣는 context의 유사성: 두 아이템이 얼마나 같은 상황에서 자주 play 되었는지 (context-item matrix **N**)

Context-aware Recommendations

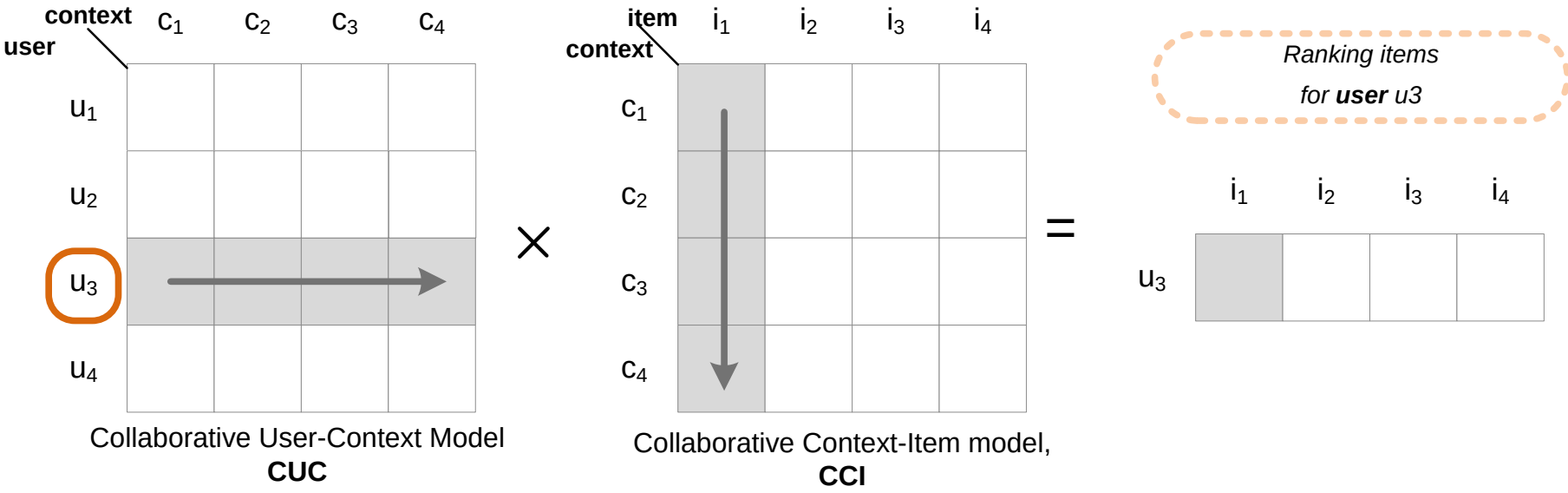
- 실시간으로 현재의 context elements들이 인지되었을 때 인지된 context에 적합한 Music Track을 Collaborative Filtering 관점에서 추천.
  - context situation $C = \{c_1, c_2, \ldots, c_p\}$에서 사용자 u에게 음악을 추천 (또는 playlist 자동재생)
  - 예) C = {겨울, 눈, 저녁, Walking}, C = {봄, 비, 오전}, C = {가을, 오전, Running}

- Collaborative User-Context (CUC) Model과 Collaborative Context-Item (CCI) Model를 함께 이용



Recommendation without Context

- Collaborative User-Context (CUC) Model과 Collaborative Context-Item (CCI) Model를 이용하여 Context를 고려하지 않은 개인화 추천에도 활용 가능



Collaborative User-Context Model
CUC

Collaborative Context-Item model,
CCI

Contextual Modeling

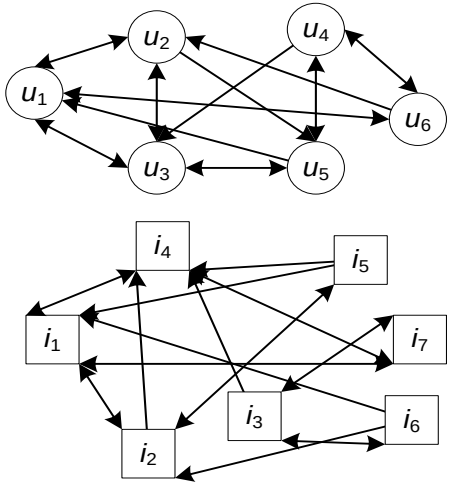
ContextRank: Graph Link-Structure Analysis for CARS

(***Heung-Nam Kim et al., RecSys 2011***)

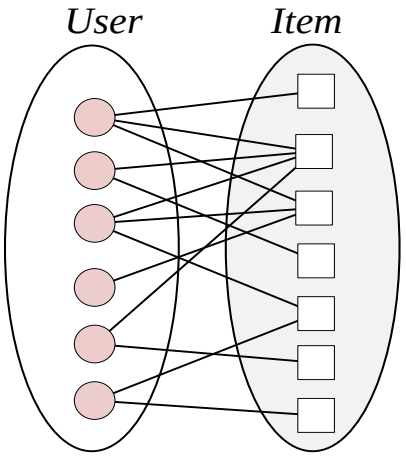
Graph Modeling

- Representation of a set of objects (called nodes or vertices) where some pairs of the objects are connected by links (called edges).
  - Node: 사용자, 아이템 (영화, 음악, 앱, 뉴스 등), 상황, …
  - Edge: relations, behaviors, log …

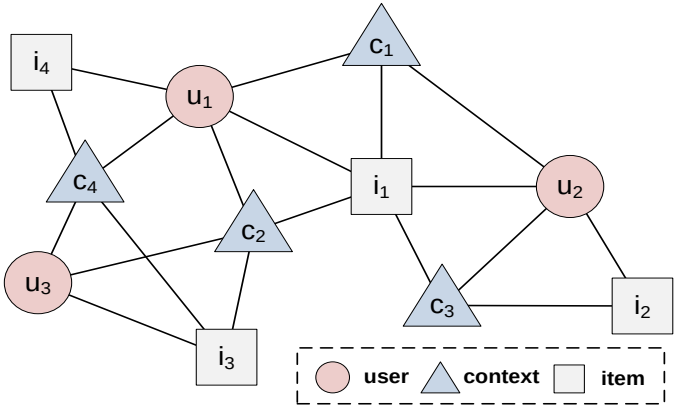
Many problems can be modeled as ***graph structures***.
Finding and extracting useful information from graph models



(a) Graph



(b) Bipartite Graph

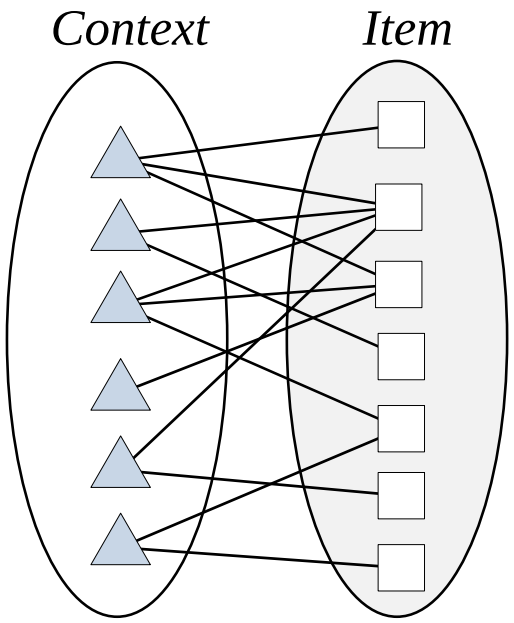


(c) Tripartite Graph

Individual Bipartite Graph

- Context에 따른 사용자 행태 로그 → Bipartite Graph Modeling

User A 행태 로그



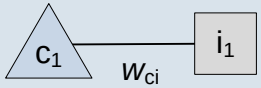
(a) Bipartite graph Model

Two types of nodes

Context condition, e.g., 집, 회사, 아침, 점심, etc.

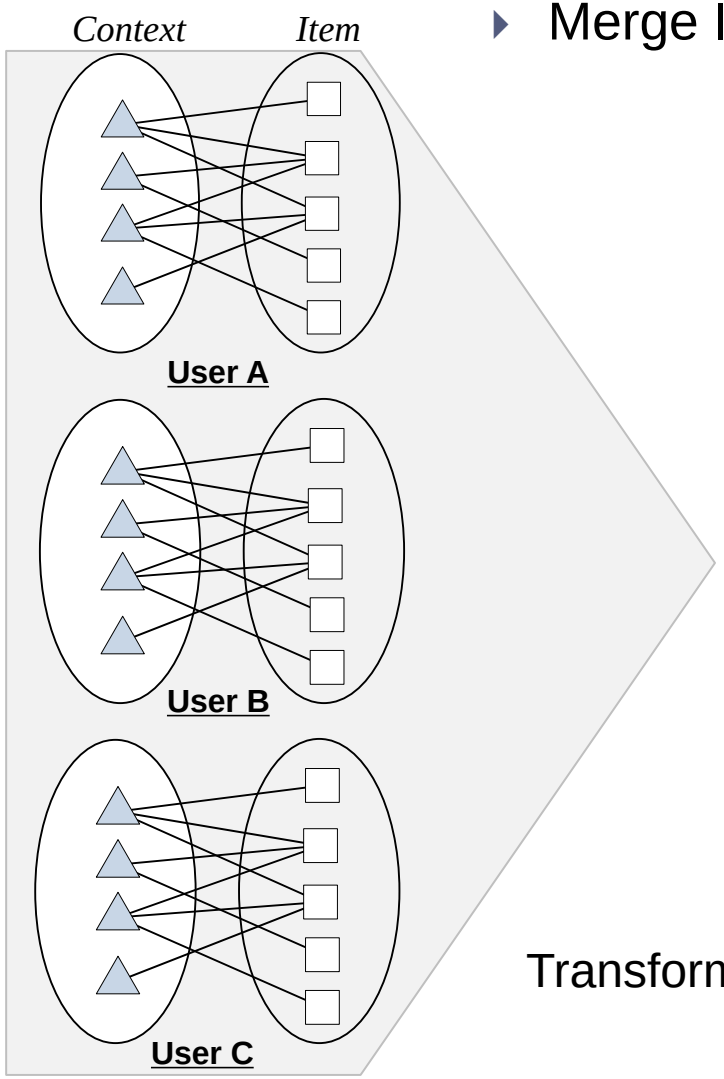
Action Item, e.g., music, films, Apps, etc.

Edges

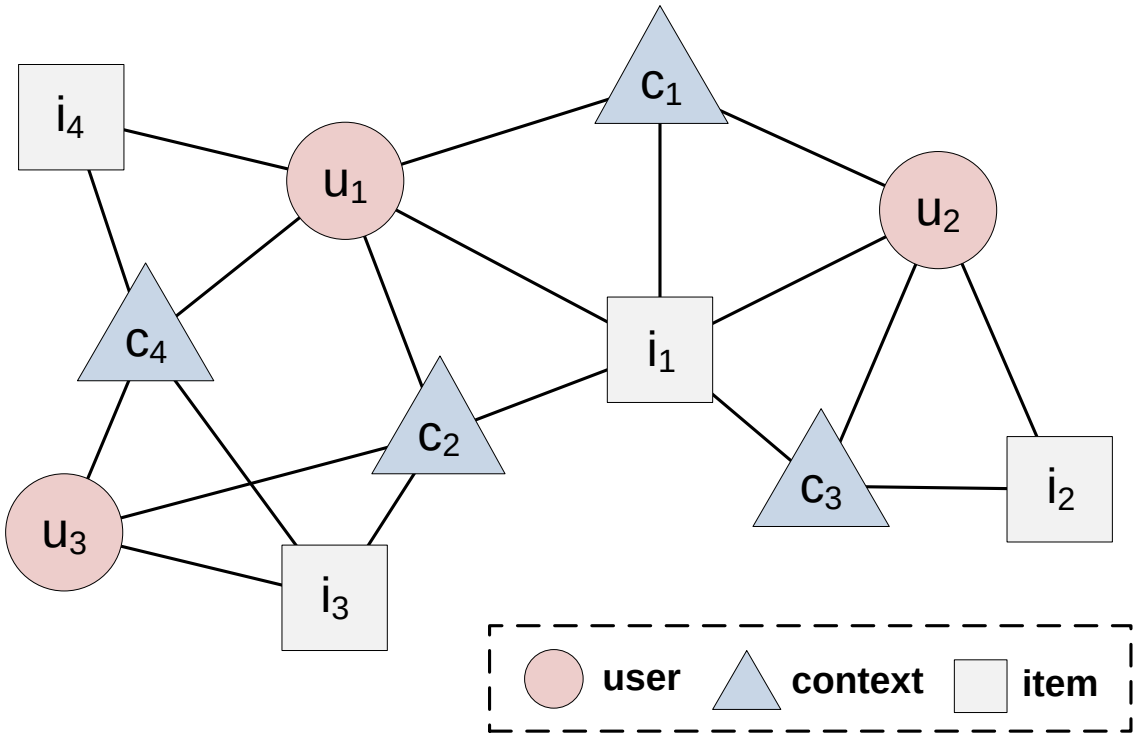


Representing how frequently a target user has used item i1 in context C1

Tripartite Graph for CARS



- Merge Individual Bipartite Graphs



Transforming data into ***an undirected, weighted tripartite*** **graph**

$G_{TRI} = (U \cup C \cup I, \mathrm{E})$

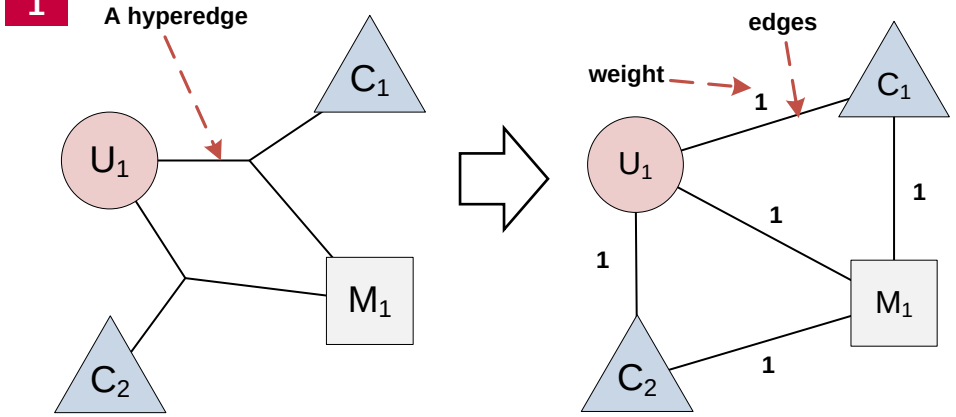
Tripartite Graph for CARS

- Assume each context dimension is *independent* of every other dimension

Example of Music Play History context

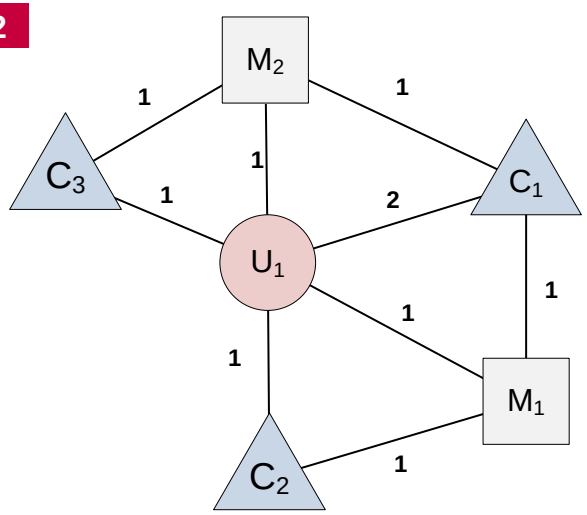
Log ID	User	Item	Time	Activity
1	Alice(U_1)	Cheer Up(M_1)	Morning(C_1)	Running(C_2)
2	Alice(U_1)	TT(M_2)	Morning(C_1)	In vehicle(C_3)
…	…	…	…	…

1



Hypergraph **Tripartite graph**

2

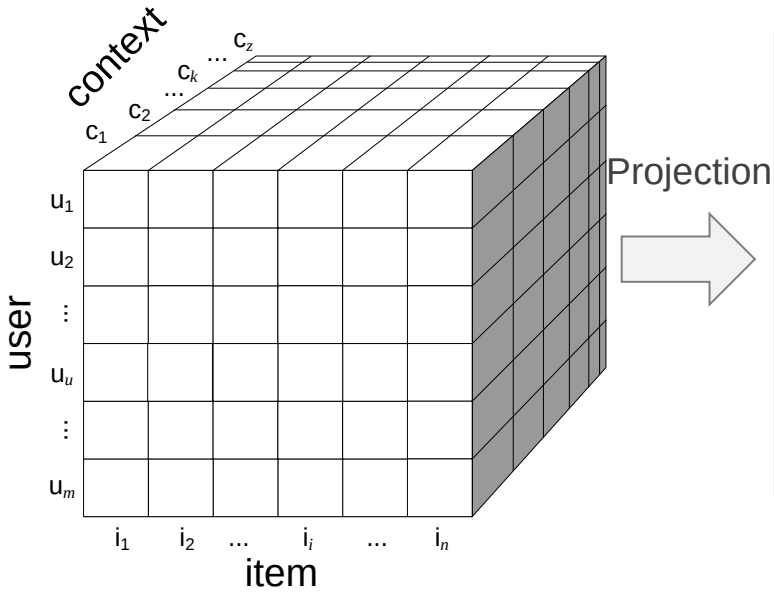


Adding Log ID2 into the graph

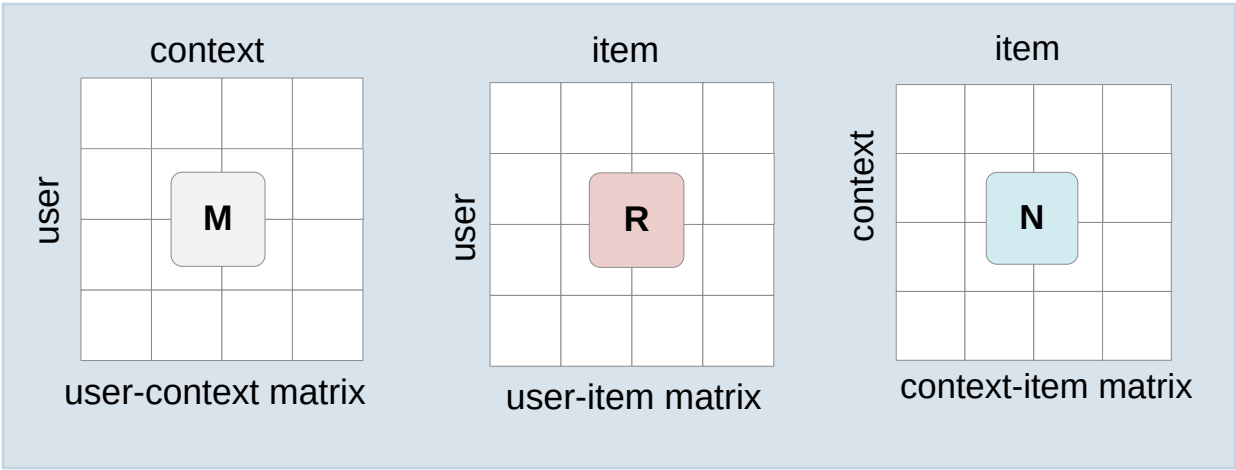
Log1 (U1, M1, C1) Alice는 아침에 “Cheer up” 음악을 들었다
(U1, M1, C2) Alice는 달리면서 “Cheer up” 음악을 들었다

Log2 (U1, M2, C1) Alice는 아침에 “TT” 음악을 들었다
(U1, M2, C3) Alice는 차 안에서 “TT” 음악을 들었다

Adjacency Matrix of CARS Tripartite Graph



Projection



Adjacency matrix corresponding to the tripartite graph $\boldsymbol{G_{TRI}}$

$$\mathbf{A} = \begin{array}{c} \\ \text{user} \\ \text{context} \\ \text{item} \end{array} \begin{array}{c} \begin{array}{ccc} \text{user} & \text{context} & \text{item} \end{array} \\ \begin{pmatrix} \mathbf{0} & \mathbf{M} & \mathbf{R} \\ \mathbf{M}^T & \mathbf{0} & \mathbf{N} \\ \mathbf{R}^T & \mathbf{N}^T & \mathbf{0} \end{pmatrix} \end{array}$$

인접행렬 A는 symmetric임

→ 그래프 특성상 dangling node 없음

Random Walk with Restart (RWR) Model

- 사용자가 특정 상황에서 사용 (행동)할 가능성이 가장 높은 아이템 발견을 위해 Random Walk with Restart (RWR) Model 이용
  - Target 사용자와 특정 context conditions에 시작하는 random walker
  - (1- *d*) 확률: 각 step에서 현재 머무르고 있는 node의 out-link를 따라 이동
  - *d* 확률: 특정 시작 context conditions으로 되돌아와 다시 시작

- Adapting ***PageRank*** into CARS *tripartite* graph

PageRank vector

$$\vec{\mathbf{r}} = (1-d)\bar{\mathbf{A}}\,\vec{\mathbf{r}} + d\vec{\mathbf{p}}$$

d: damping factor (*restart probability)*
$\bar{\mathbf{A}}$: column stochastic matrix of adjacency matrix $\mathbf{A}$
$\mathbf{p}$: preference vector (*restart vector*)

ContextRank

- PageRank vector with the ***non-uniform preference vector***
  - aka. Personalized PageRank Vectors (PPV)
  - For a given user-context conditions (u, c_1,c_2,…,c_k), set entries of preference vector **p**

current user

$$\mathbf{p}(u) = \frac{|U|}{|U| + |C|}$$

Current context

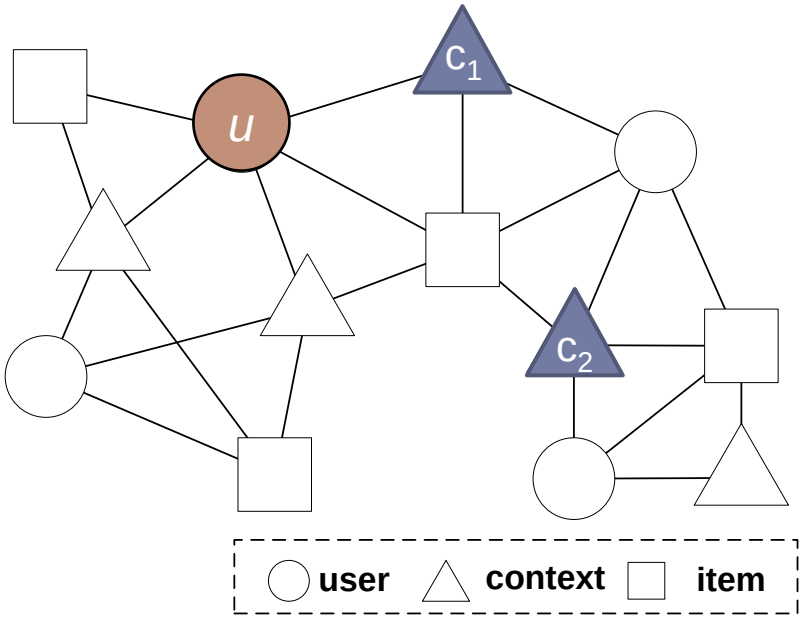
$$\mathbf{p}(c_j) = \frac{|C|}{k(|U| + |C|)}$$

all other entries

$$\mathbf{p}(v) = 0$$

such that $\| \mathbf{p} \|_1 = 1$

Given a user-context (u, c_1, c_2),
restarting from (or jumping back to)



ContextRank

- 현재 Context에서만 dominant한 아이템 발견 (Context에 무관하게 빈번히 발생되는 아이템 제외)
  - Undirected Graph 특성상 random walker가 항상 다음 단계에 이전 단계의 Node로 다시 return back 함.

- Differential ranking approach

$$\vec{\mathbf{r}} = (1-d)\overline{\mathbf{A}}\,\vec{\mathbf{r}} + d\vec{\mathbf{p}}$$

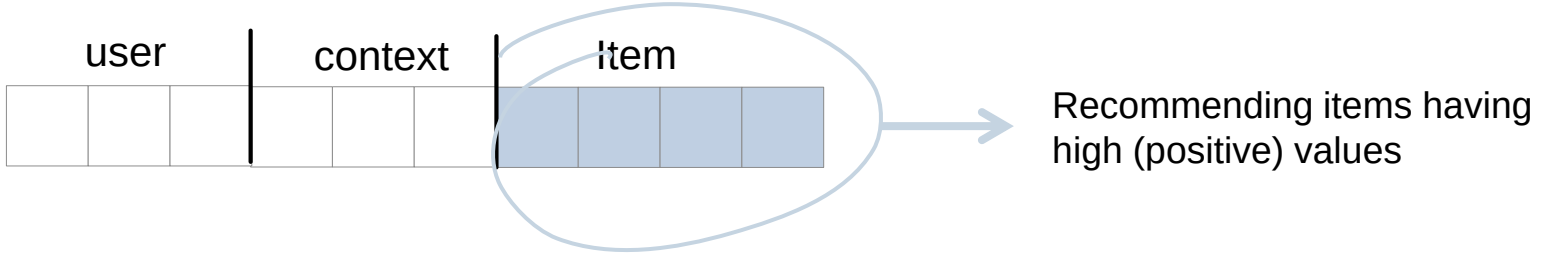
Personalized PageRank
(Random Walk with Restart)

$$\vec{\mathbf{g}} = \overline{\mathbf{A}}\vec{\mathbf{g}}$$

Markov Chain
(Random Walk)

$$\vec{\mathbf{e}} = \vec{\mathbf{r}} - \vec{\mathbf{g}}$$

ContextRank



Section2. TARS:
Trust-Aware Recommender Systems

Some Challenges in Collaborative Filtering

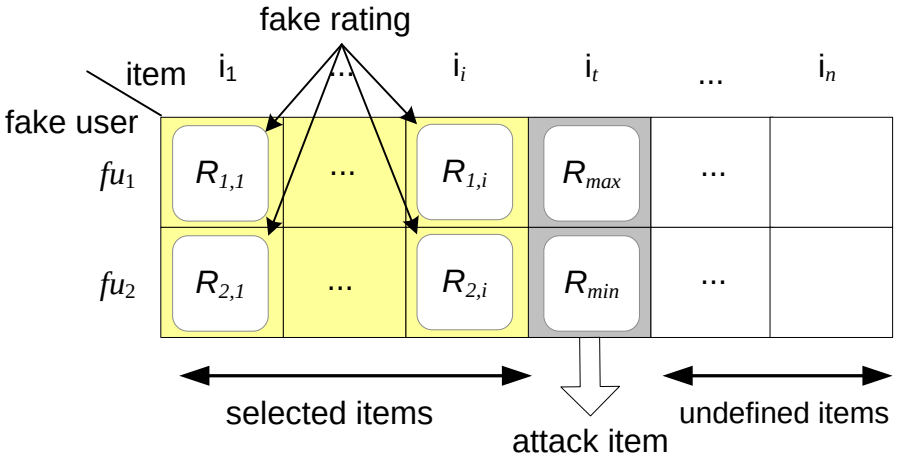
- **Cold Start Problem**:

  A new user joins the system and have few ratings, ***it is hard to***
  - identify similar users (i.e., neighborhood)
  - make the quality recommendation

- **Shilling Attacks** (profile injection attacks):

  Injecting manipulated profiles by *shills*
  - to recommend an item more highly (**push attack**)
  - to prevent an item from recommending (**nuke attack**)
  - to deteriorate the performance of a system

Shilling Attacks (O'Mahony et al., *ACM TOIT, 2004*)



Push attack

Nuke attack

Push Attack 예

attack item

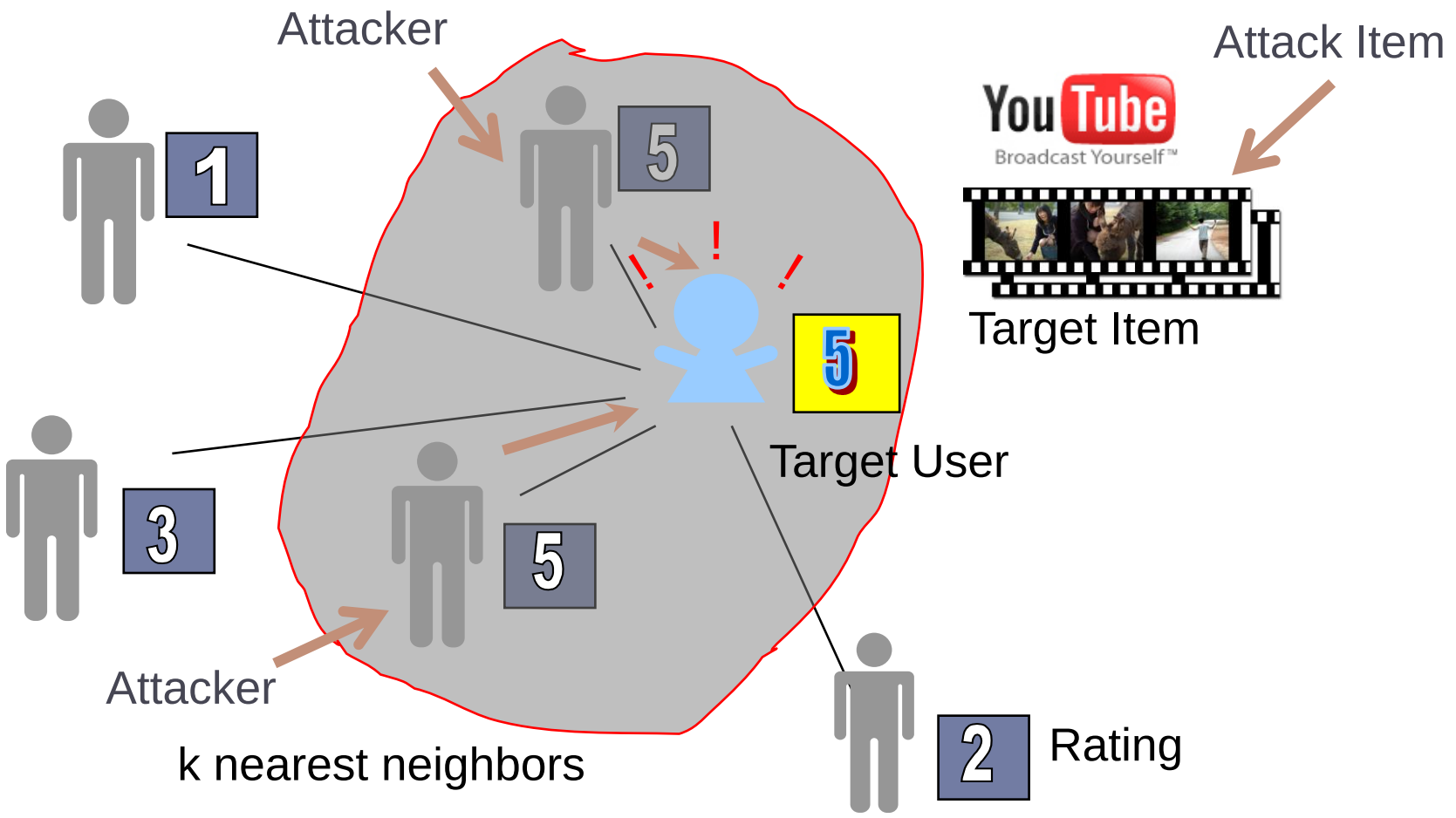
	명랑	변호인	암살	베테랑	괴물	아바타
Alice	3	5	-	1	?	5
Bob	4	4	5	4	3	-
John	2	5	4	-	?	4
Dannis	-	1	2	-	5	4
Faker 1	3	4		2	5	4
Faker 2	3		4		5	4

Alice의 추천이 fake users에 의해 영향을 받음

영화 "괴물"의 추천을 높이기 위한 fake users

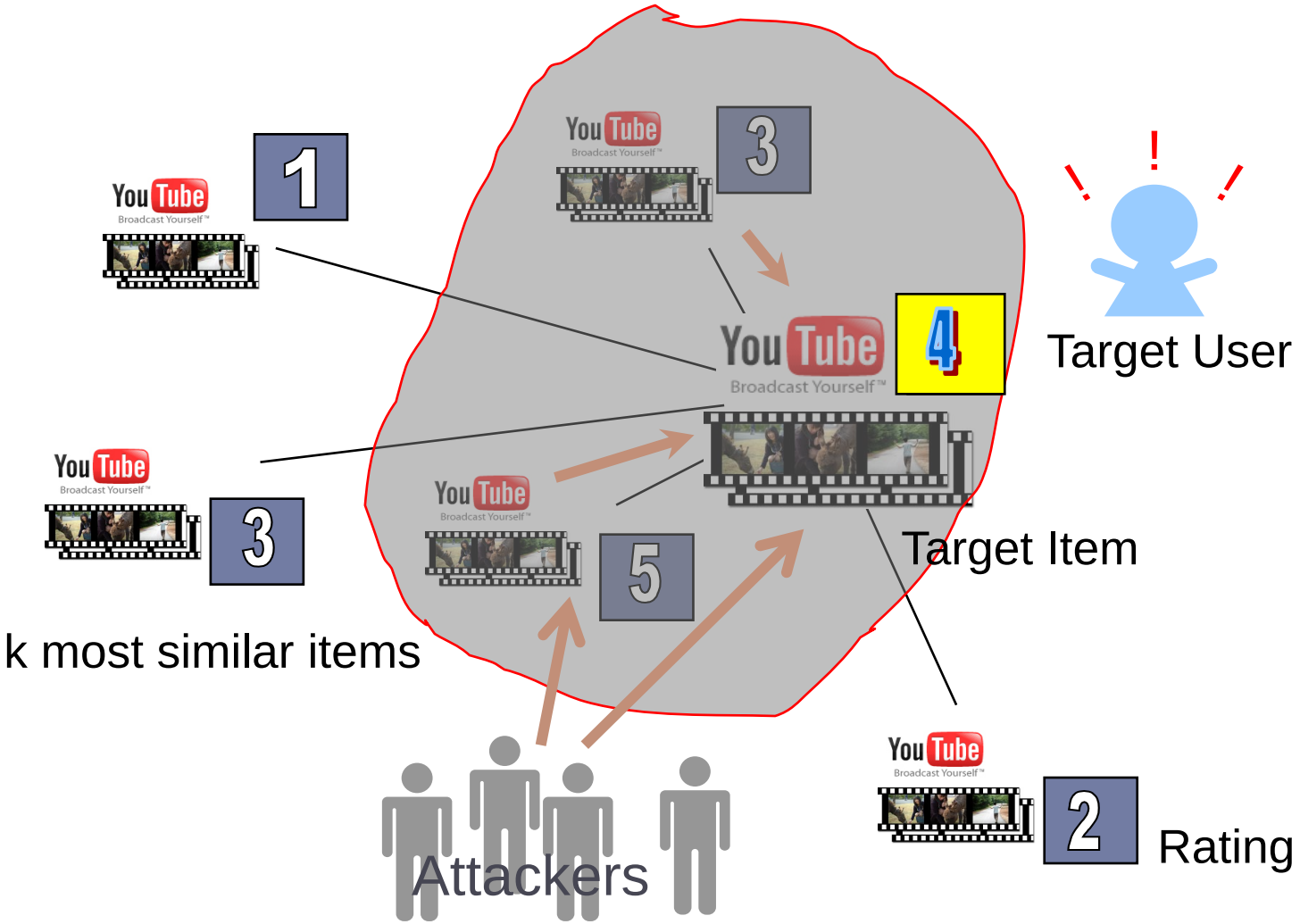
Manipulated rating by attackers

User-based Collaborative Filtering
If attacker are neighborhood of the target user….



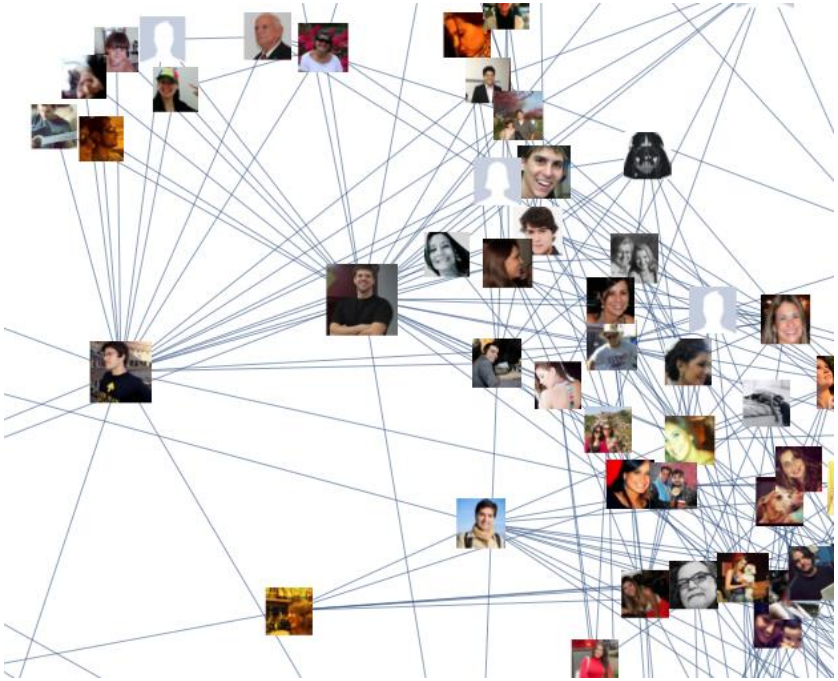
Item-based Collaborative Filtering

It is more robust because of looking into the set of items the target user has rated, but…

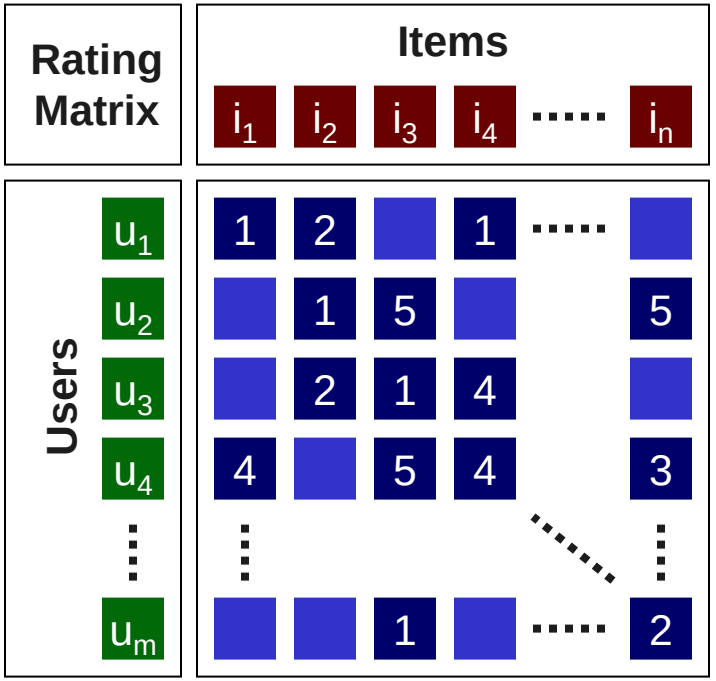


Trust-Aware Recommender Systems (TARS)

- Incorporating trust/social relationships into RecSys



Social (Trust) Network



User-Item Rating

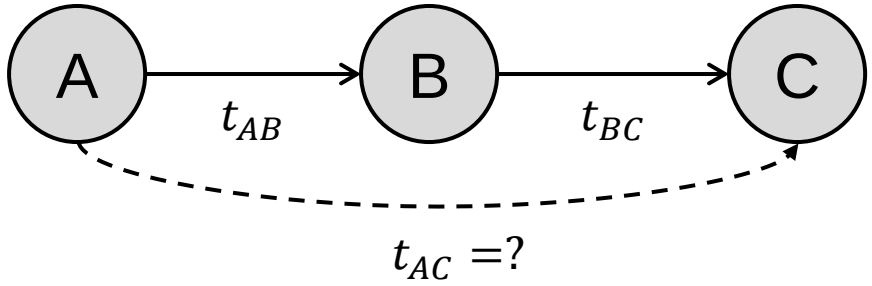
Users (truster) x **Users** (trustee) → Trusts

Users x Items → Ratings

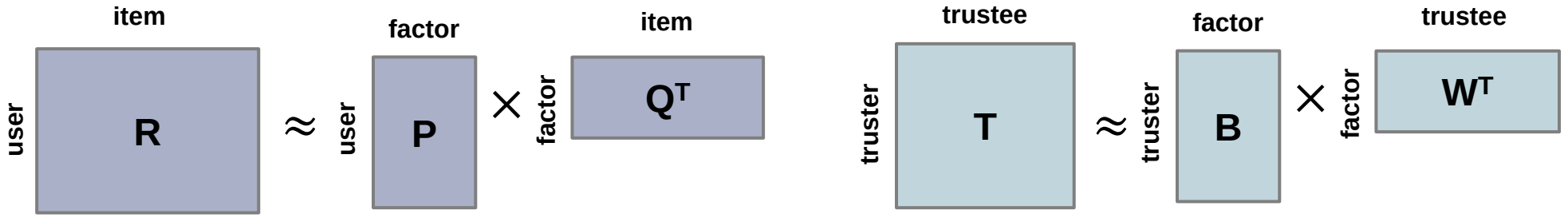
Trust-Aware Recommender Systems (TARS)

1. Trust Inference Approaches

- Use trust as a way to give more weight to some users or trust in place of (or combined with) user-user similarity



2. Matrix Factorization Approaches



Trust Inference Approaches in TARS

Major algorithms

MoleTrust Massa and Avesani, *Trust-aware Recommender Systems*, RecSys 2007

TidalTrust Golbeck, *Computing and applying trust in web-based social networks*, Doctoral Dissertation, 2005

TrustWalker Jamali and Ester, *TrustWalker: A Random Walk Model for Combining Trust-based and Item-based Recommendation*, SIGKDD 2009

Matrix Factorization Approaches in TARS

Major algorithms

SoRec Ma et al., *SoRec: Social Recommendation Using Probabilistic Matrix Factorization*, SIGIR 2008

RSET Ma et al., *Learning to Recommend with Social Trust Ensemble*, SIGIR 2009

SocialMF Jamali and Ester, *A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks*, RecSys 2010

SoReg Ma et al., *Recommender systems with social regularization*, WSDM,2011

TrustMF Yang et al., *Social Collaborative Filtering by Trust*, IJCAI 2013

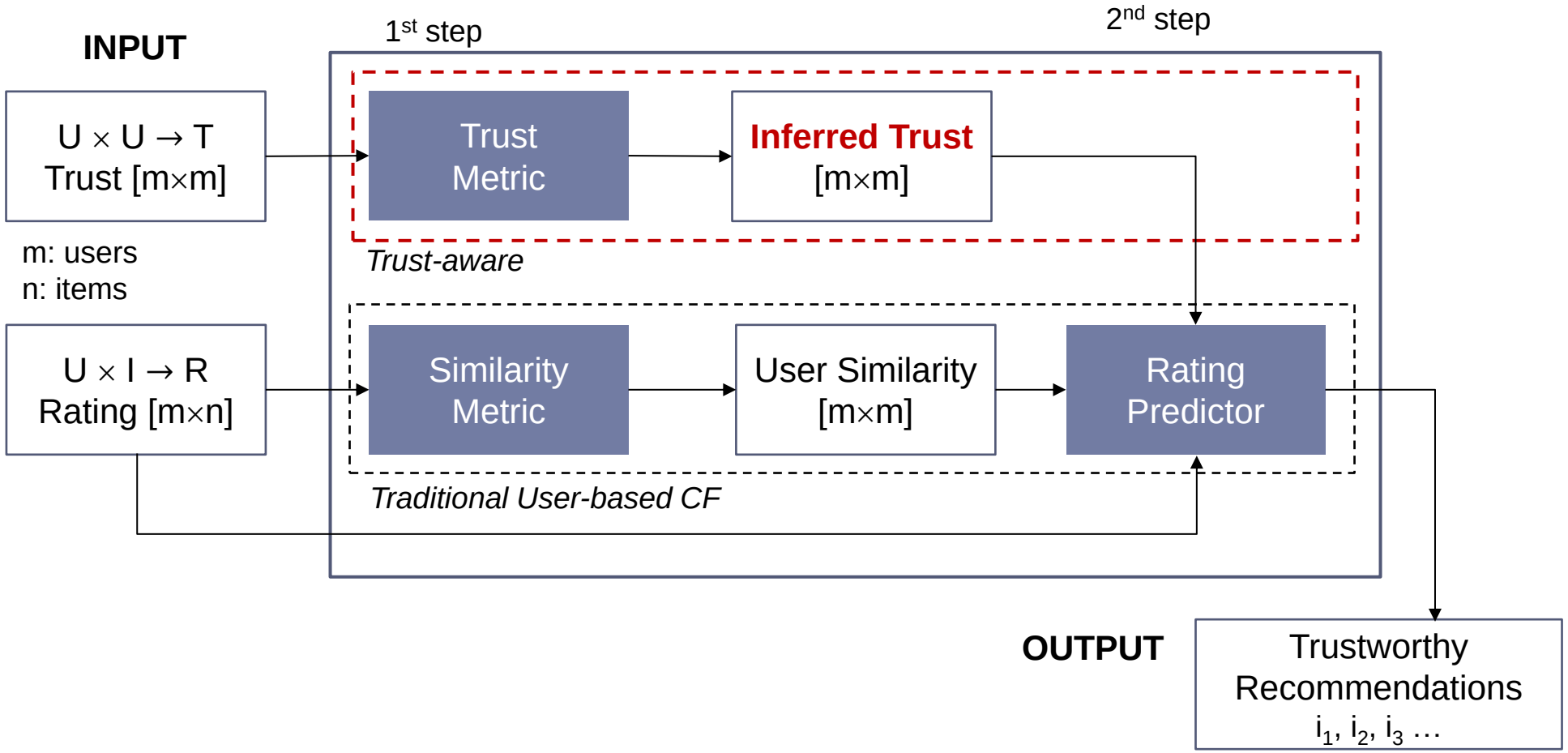
TrustSVD Guo et al., *TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings*, AAAI 2015

Trust Inference

Random Walk with Restart Model

Trust-aware Collaborative Filtering (CF)

- Use trust as neighbor selector and their weight for recommendations



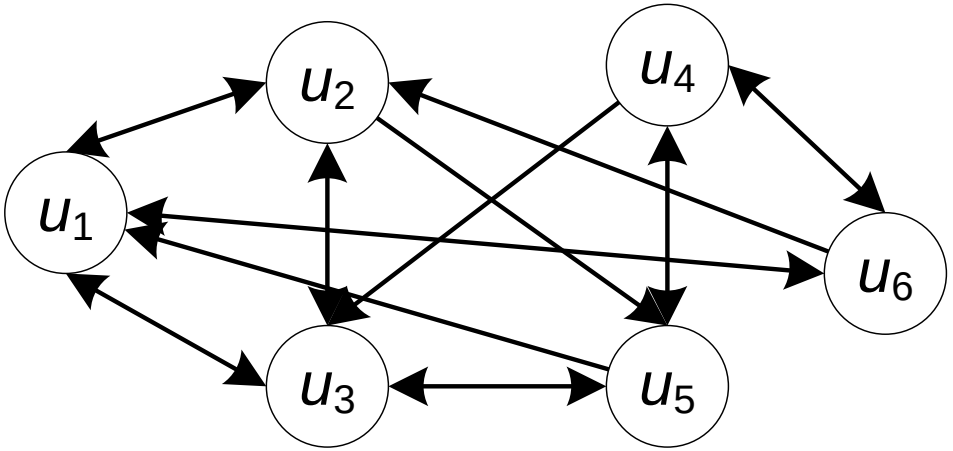
(Massa and Avesani, RecSys, 2007)

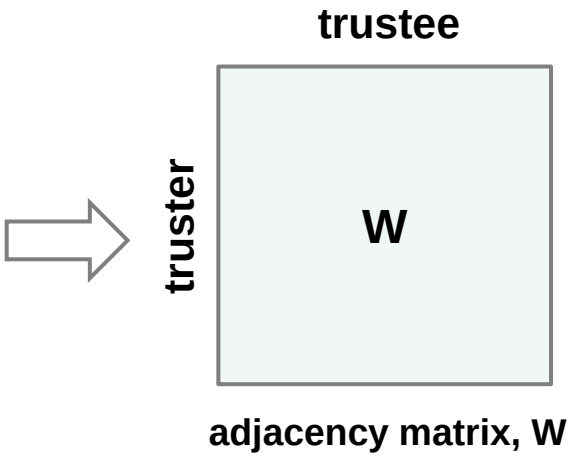
Trust (Social) Network

- Representation of a set of nodes where some pairs of the nodes are connected by links (called edges).
  - Node: users
  - Edge: user *a* follows (trusts) user *b*

$G = (U, E)$

a set U of nodes,
a set E of edges (pair of nodes)

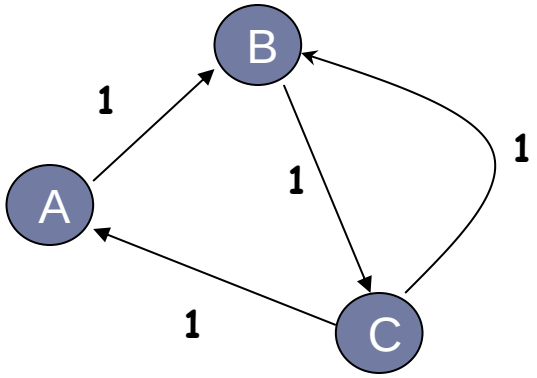


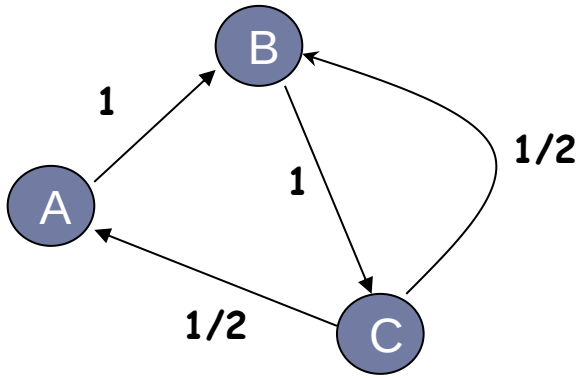


Graph Characteristics

- ✓ directed vs. undirected graph
- ✓ weighted vs. unweighted graph
- ✓ signed vs. unsigned graph

Random Walk Model?





Adjacency matrix

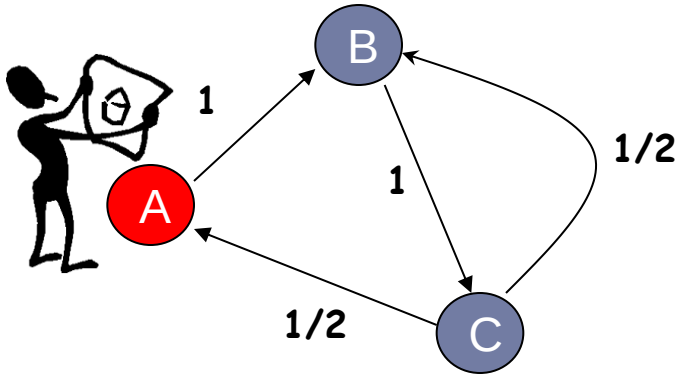
$$\begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \end{array}$$

Transition matrix

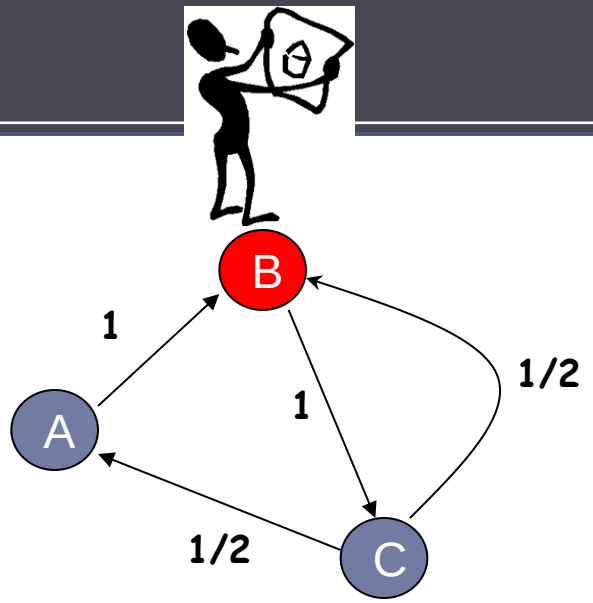
$$\begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1/2 & 1/2 & 0 \end{pmatrix} \end{array}$$

Random Walk Model?

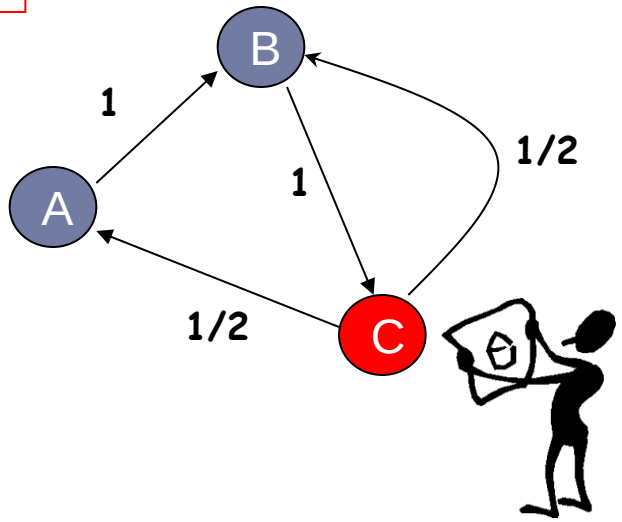
t=0



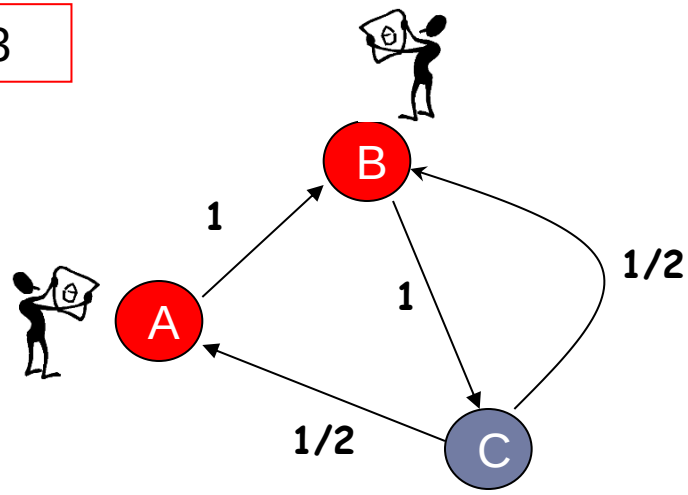
t=1



t=2



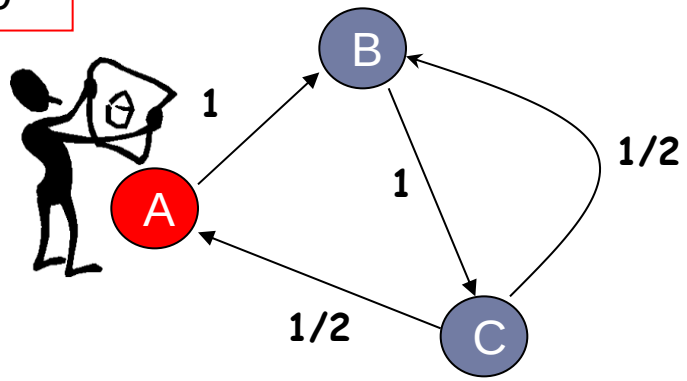
t=3



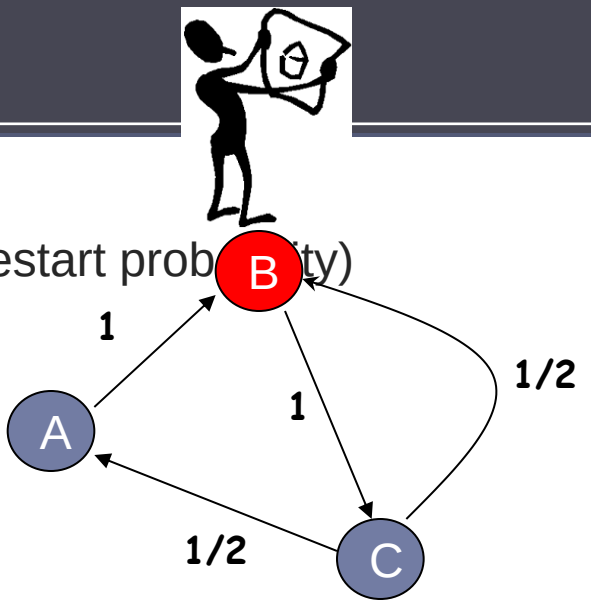
Random Walk with Restart Model?

- At any time-step, the random walker can
  - jump (**teleport**) to any other node with probability ***d*** (restart probability)
  - jump to its direct neighbors with probability ***1-d***.

t=0

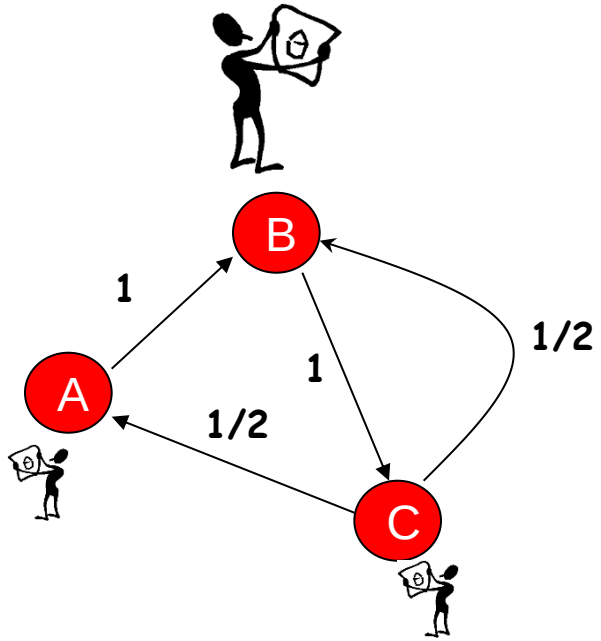


0.85 확률



t=1

0.15 확률



Random walker가 각 step에서 모든 node로 균등하게 restart 할 수 있다고 가정한다면 (t=1) 일 때 각 노드에 방문할 확률?

- ✓ P(A) = 0.15* 0.33333 = 0.05
- ✓ **P(B) = 0.85*1 + 0.15* 0.33333 = 0.9**
- ✓ P(C) = 0.15* 0.33333 = 0.05

Global PageRank (Page et al., TR Stanford, 1999)

- **PageRank**:
  Social Network Graph에서 random walker가 가장 도착할 확률이 높은 사용자

  → Social Network 서비스에서 영향력이 높은 (중요한) 사용자들 발견

- 사용자 u의 **Global PageRank**

$$PR(u) = \frac{d}{m} + (1-d) \sum_{v \in In(u)} \frac{PR(v)}{\sum_k w_{vk}}$$

  - m: 총 그래프 노드 수 (앱 수)
  - d: damping factor 또는 restart 확률
  - $In(u)$: 사용자 u로 이동하는 link가 있는 사용자들 집합 (u의 in-link 노드들)
  - w_{vk}: Adjacency matrix **W**의 v행 k열의 값
  - $PR(v)$: 사용자 v에 대한 Global PageRank

Global PageRank

- 모든 사용자에 대한 **Global PageRank** 계산을 위한 power iteration

$$\vec{\mathbf{r}}_{(t+1)} = (1-d)\bar{\mathbf{W}}^{T}\vec{\mathbf{r}}_{(t)} + d\vec{\mathbf{p}}$$

stationary distribution vector (GPageRank vector)

Transition matrix (Stochastic matrix of W)

Restart probability (d= 0.15)

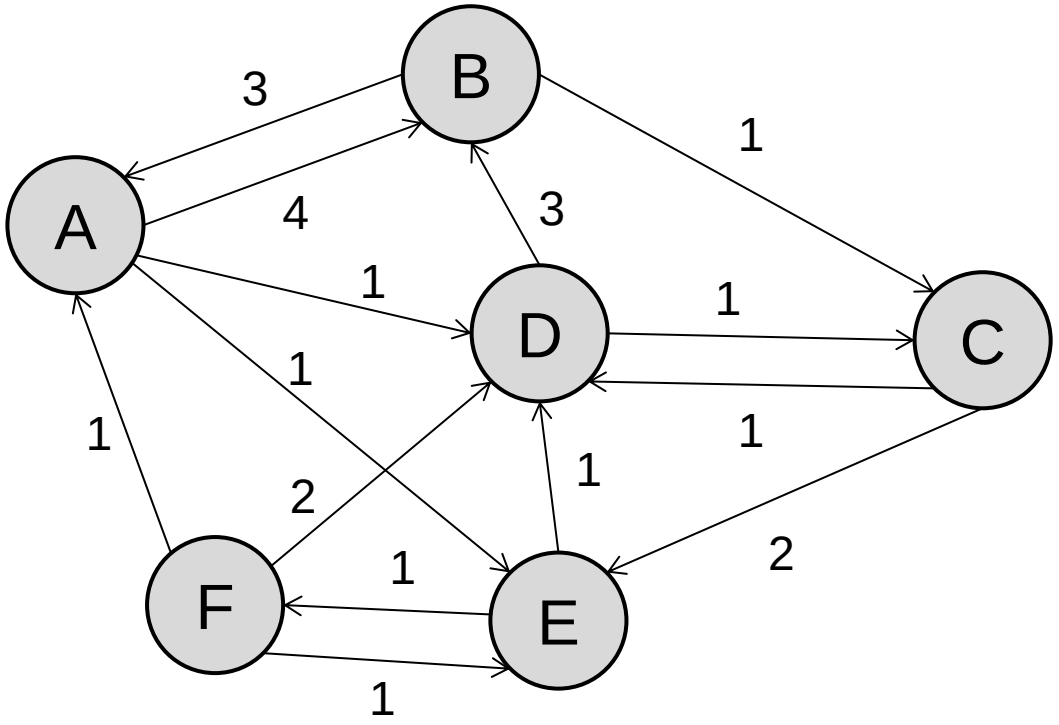
Uniform Restart vector

- $\bar{\mathbf{W}}$: (m × m) Adjacency matrix **W**의 row-stochastic matrix (각 행의 합이 1).
- $\vec{\mathbf{p}}$: (m × 1) global restart vector
- $\vec{\mathbf{r}}$: (m × 1) PageRank vector

* $\mathbf{p} = [1/m\ 1/m \ldots 1/m]^T$

* initial $\mathbf{r}_{(0)} = [1/m\ 1/m \ldots 1/m]^T$

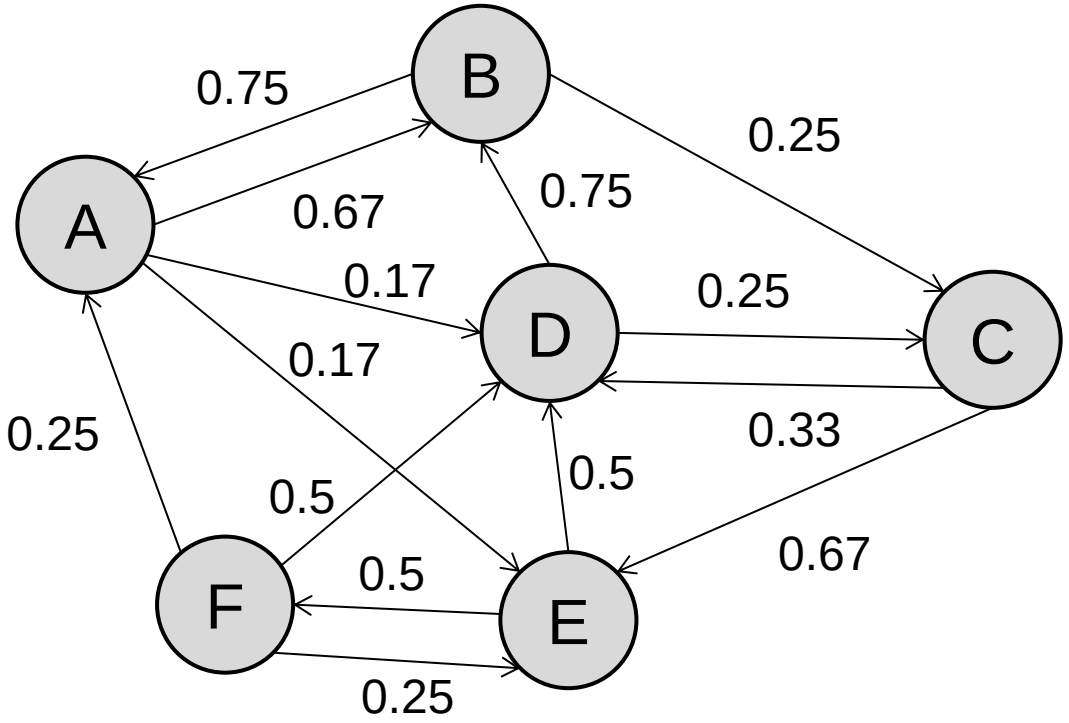
Example



	A	B	C	D	E	F
A	0	4	0	1	1	0
B	3	0	1	0	0	0
C	0	0	0	1	2	0
D	0	3	1	0	0	0
E	0	0	0	1	0	1
F	1	0	0	2	1	0

Adjacency matrix **W**

Example



	A	B	C	D	E	F
A	0	4	0	1	1	0
B	3	0	1	0	0	0
C	0	0	0	1	2	0
D	0	3	1	0	0	0
E	0	0	0	1	0	1
F	1	0	0	2	1	0

Adjacency matrix **W**

Row-stochastic(**W**)

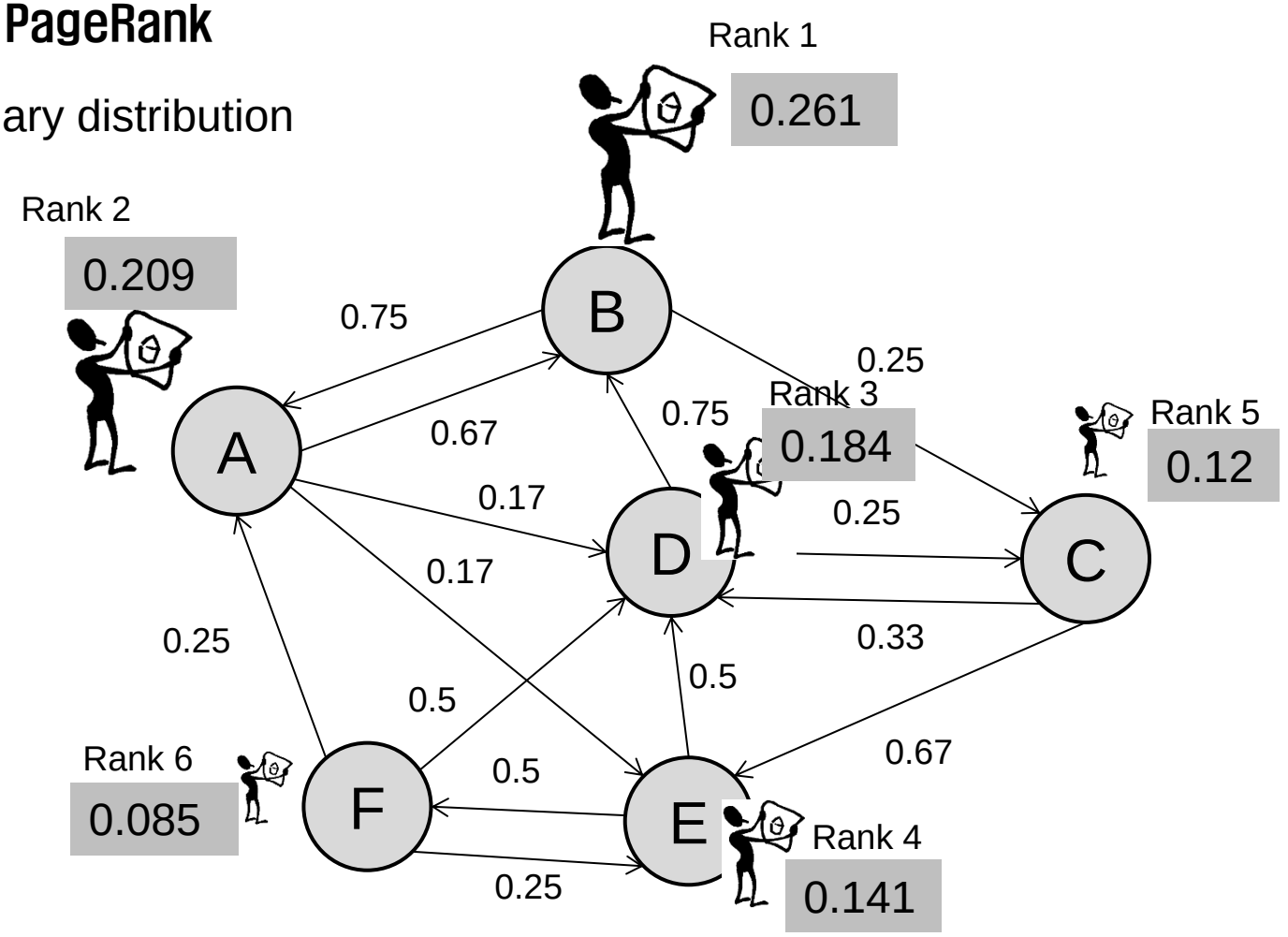
	A	B	C	D	E	F
A	0	0.67	0	0.17	0.17	0
B	0.75	0	0.25	0	0	0
C	0	0	0	0.33	0.67	0
D	0	0.75	0.25	0	0	0
E	0	0	0	0.5	0	0.5
F	0.25	0	0	0.5	0.25	0

Transition matrix $\bar{\mathbf{W}}$

Example - Global PageRank

Global PageRank

stationary distribution



Rooted PageRank

- **Rooted PageRank** (aka. Personalized PageRank)
  random walker가 특정한 **하나**의 노드에서 출발하여 주기적으로 출발 노드로 이동하여 다시 출발할 때 가장 도착할 확률이 높은 사용자들 계산

  → 현재 focus중인 사용자에 대하여 ranking vector 계산

- 사용자 u에 대한 **Rooted PageRank**

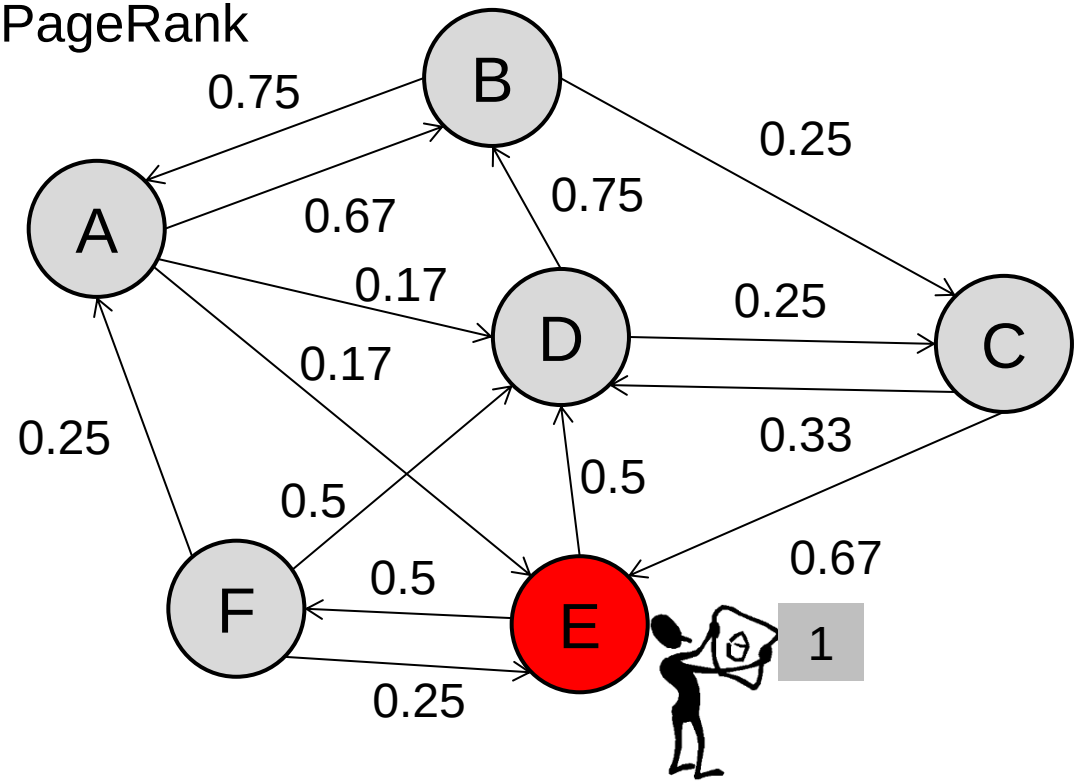
$$\vec{\mathbf{r}}_u = (1-d)\bar{\mathbf{W}}^T\vec{\mathbf{r}}_u + d\vec{\mathbf{p}}_u$$

Non-uniform Restart vector

  - $\overrightarrow{p_u}$: (m × 1) single restart vector
    (현재 target 사용자 노드의 값만 **p**(u) = 1 이고 다른 노드의 값은 0)
  - Initial $\overrightarrow{r_0} = \overrightarrow{p_i}$
    (현재 target 사용자 노드의 값만 1이고 다른 노드의 값은 0)
  - $\vec{\mathbf{r}}_u$: (m × 1) Rooted PageRank vector for user *u*

Example - Rooted PageRank

사용자 E에 대한 PageRank



$t = 0$

	A	B	C	D	E	F
$\vec{r}_{(0)} =$	0	0	0	0	1	0
$\vec{p} =$	0	0	0	0	1	0

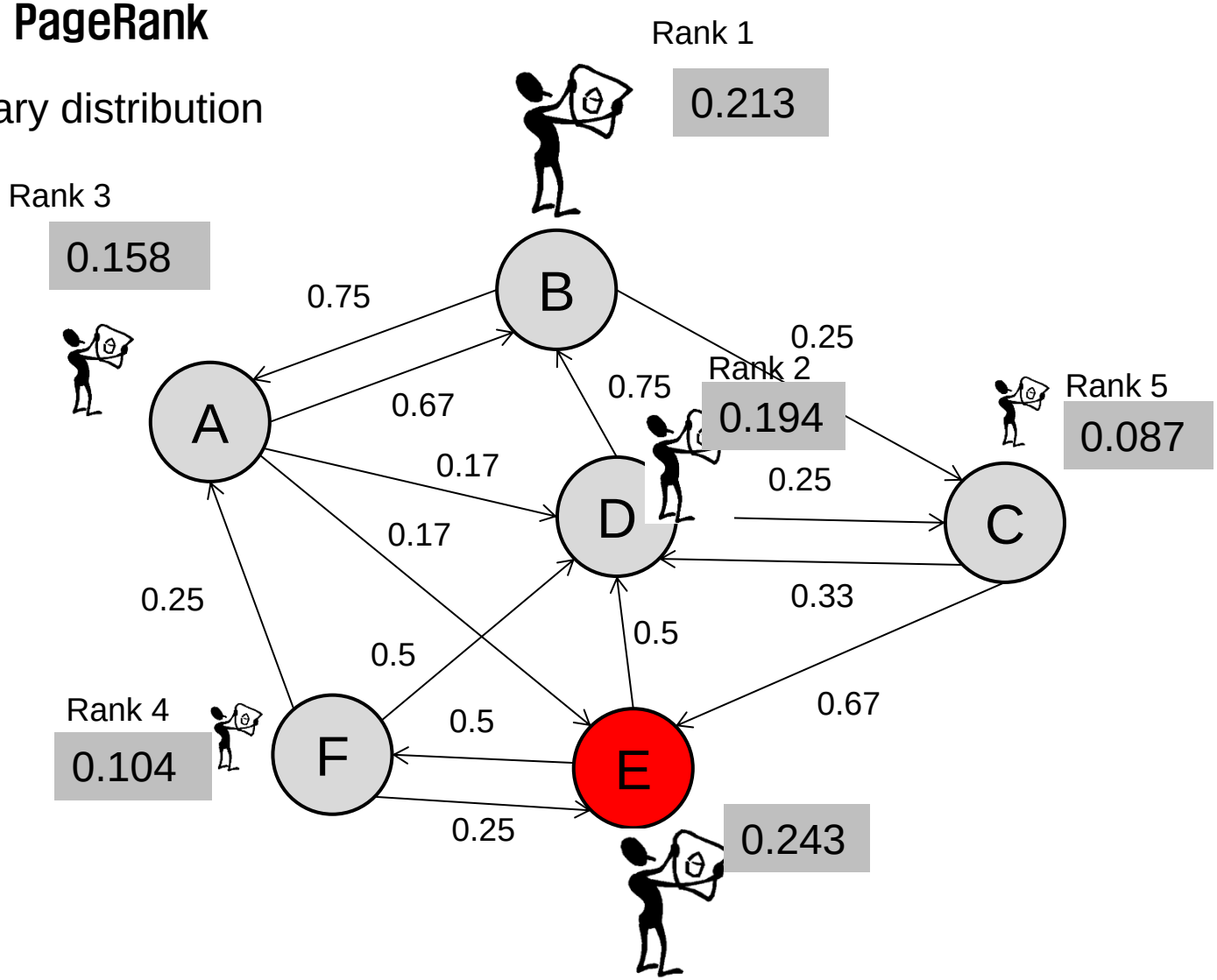
출발 노드

Restart 노드

Example - Rooted PageRank

Rooted PageRank

stationary distribution



Trust-based Prediction

- Rating Prediction

: *k* trusted neighbors들의 rating pattern을 이용하여 특정 아이템에 대한 target 사용자의 rating을 예측

1. Weighted Sum

$$p_{ui} = \frac{\sum_{v \in T_i(u)} r_{vi} \times t_{uv}}{\sum_{v \in T_i(u)} t_{uv}}$$

2. Mean-centering

$$p_{ui} = \bar{r_u} + \frac{\sum_{v \in T_i(u)} (r_{vi} - \bar{r_v}) \times t_{uv}}{\sum_{v \in T_i(u)} t_{uv}}$$

→ Normalized trust value with respect to user u (i.e., Rooted PageRank)

$$t_{uv} = \frac{\mathbf{r}_u(v) - \mathbf{r}_{u\min}}{\mathbf{r}_{u\max} - \mathbf{r}_{u\min}}$$

$$0 \le t_{uv} \le 1$$

아이템 i를 rating한 사용자들 중
최대 k명의 높은 신뢰도를 가진 사용자 집합

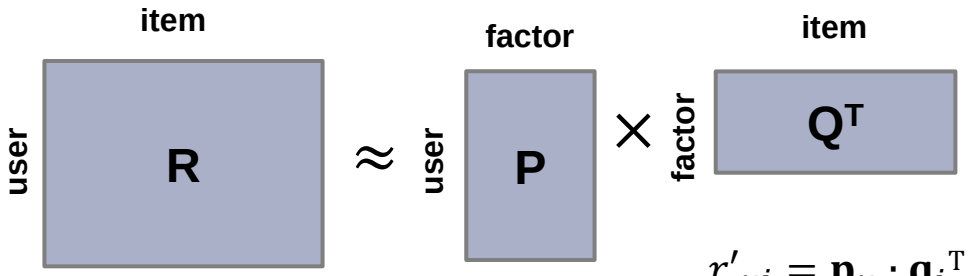
Matrix Factorization

TrustMF

(*Bo Yang et al., IJCAI 2013*)

Matrix Factorization

- User-Item Rating Matrix



$$r'_{ui} = \mathbf{p}_u \cdot \mathbf{q}_i{}^{\mathrm{T}}$$

Minimizing objective function

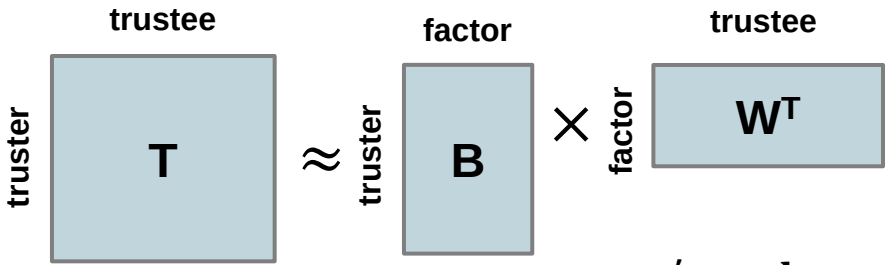
$$\min_{p,q} \sum_{r_{ui} \in \mathbf{R}} (r_{ui} - \mathbf{p}_u \mathbf{q}_i{}^{\mathrm{T}})^2 + \lambda(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2)$$

예측 값

Over-fitting을 방지하기 위한 Regularization

λ : regularization level 조절

- User-User Trust Matrix

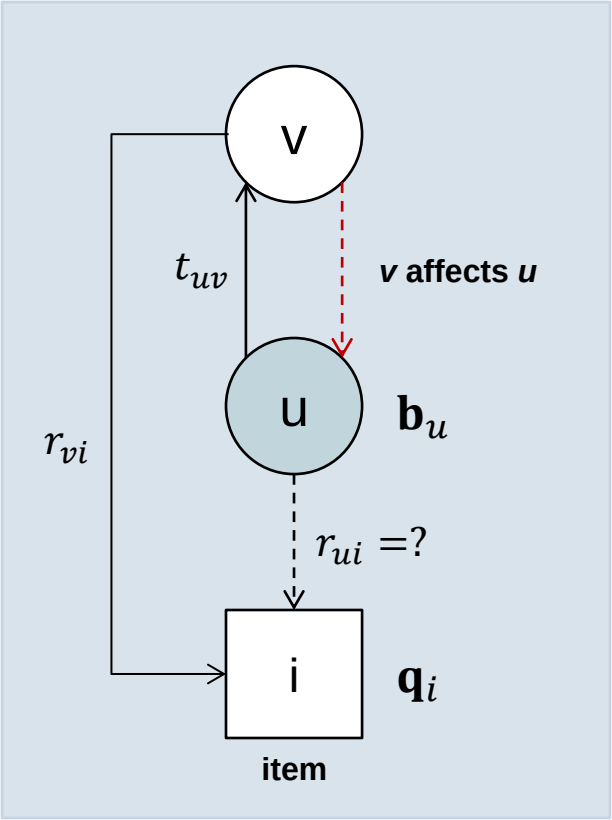


$$t'_{uv} = \mathbf{b}_u \cdot \mathbf{w}_v{}^{\mathrm{T}}$$

Minimizing objective function

$$\min_{b,w} \sum_{t_{uv} \in \mathbf{T}} (t_{uv} - \mathbf{b}_u \mathbf{w}_v{}^{\mathrm{T}})^2 + \lambda(\|\mathbf{b}_u\|^2 + \|\mathbf{w}_v\|^2)$$

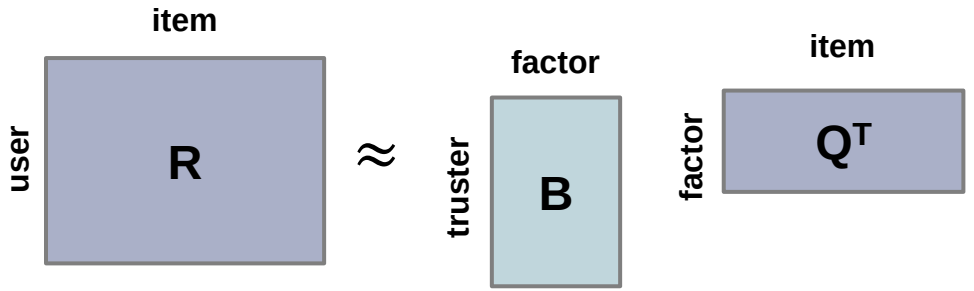
TrusterMF – Truster Model



Prediction

$$r'_{ui} = g(\mathbf{b}_u \cdot \mathbf{q}_i{}^{\mathrm{T}}) \times r_{max}$$

- Truster Model:
  How others affect a given user's rating on an item



Minimizing objective function Optimizing with gradient descents

$$L = \sum_{r_{ui} \in \mathbf{R}} (f(r_{ui}) - g(\mathbf{b}_u \mathbf{q}_i{}^{\mathrm{T}}))^2 + \lambda_T \sum_{t_{uv} \in \mathbf{T}} (t_{uv} - g(\mathbf{b}_u \mathbf{w}_v{}^{\mathrm{T}}))^2$$

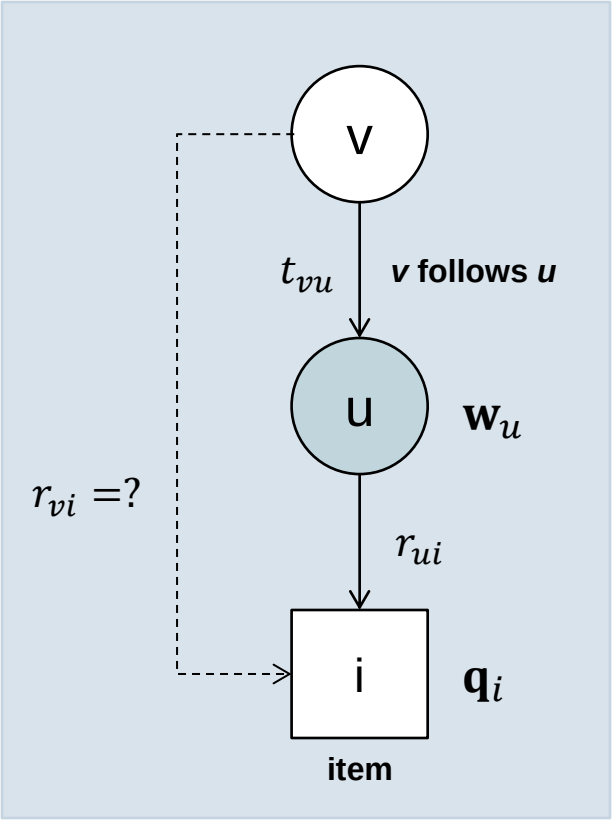
$$+\lambda(\|\mathbf{b}\|^2 + \|\mathbf{q}\|^2 + \|\mathbf{w}\|^2)$$

regularization

where $f(x) = \frac{x}{r_{max}}$ $g(x) = \frac{1}{(1 + \exp(-x))}$

convert values between 0 and 1

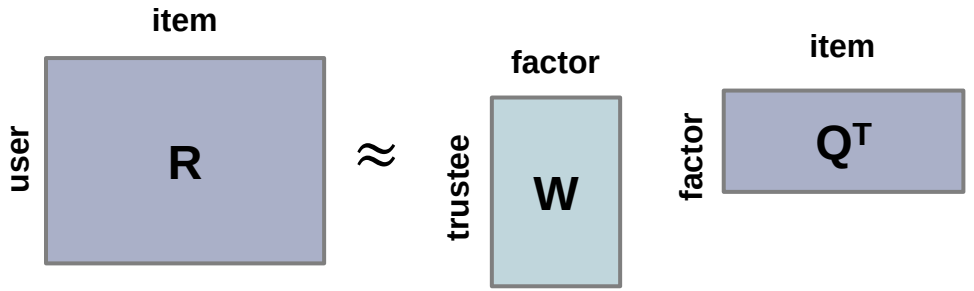
TrusteeMF – Trustee Model



Prediction

$$r'_{ui} = g(\mathbf{w}_u \cdot \mathbf{q}_i{}^{\mathrm{T}}) \times r_{max}$$

- Trustee Model:
  How a given user affect others who trust her.



Minimizing objective function Optimizing with gradient descents

$$L = \sum_{r_{ui} \in \mathbf{R}} (f(r_{ui}) - g(\mathbf{w}_u \mathbf{q}_i{}^{\mathrm{T}}))^2 + \lambda_T \sum_{t_{uv} \in \mathbf{T}} (t_{uv} - g(\mathbf{b}_u \mathbf{w}_v{}^{\mathrm{T}}))^2$$

$$+\lambda(\|\mathbf{b}\|^2 + \|\mathbf{q}\|^2 + \|\mathbf{w}\|^2)$$

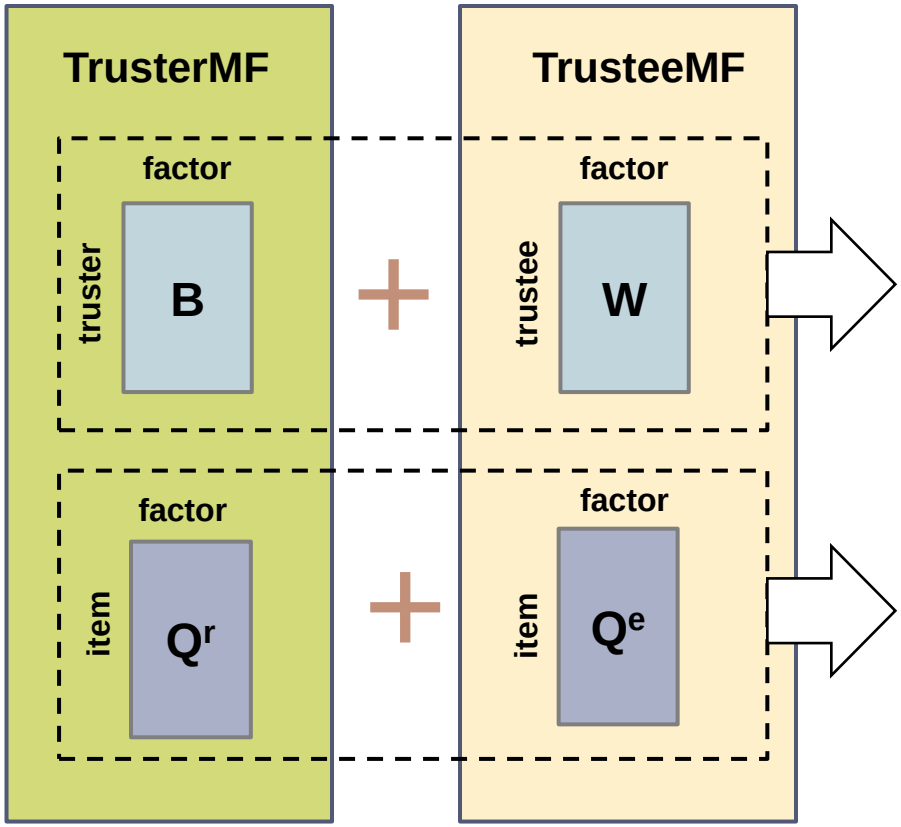
regularization

where $f(x) = \frac{x}{r_{max}}$ $g(x) = \frac{1}{(1 + \exp(-x))}$

convert values between 0 and 1

TrustMF – TrusterMF + TrusteeMF

- A user's decision would be affected by not only his trustees, but also trusters



$$\mathbf{uf}_u = \frac{\mathbf{b}_u + \mathbf{w}_u}{2}$$ user factor

$$\mathbf{if}_i = \frac{\mathbf{q}^{\mathbf{r}}{}_i + \mathbf{q}^{\mathbf{e}}{}_u}{2}$$ item factor

Prediction

$$r'_{ui} = g(\mathbf{uf}_u \cdot \mathbf{if}_i{}^{\mathrm{T}}) \times r_{max}$$

Matrix Factorization

TrustSVD

(*Guibing Guo et al., AAAI 2015*)

SVD++ (Koren, SIGKDD, 2008)

- considering also implicit feedback into *biased MF*, which provides an additional indication of user preferences

  특히 사용자의 explicit rating이 많지 않을 때 향상된 성능 제공할 수 있음

아이템 i에 대한 사용자 u의 **biased MF** 예측 rating

$$r'_{ui} = \mu + b_u + b_i + \mathbf{p}_u \mathbf{q}_i^{\mathrm{T}}$$

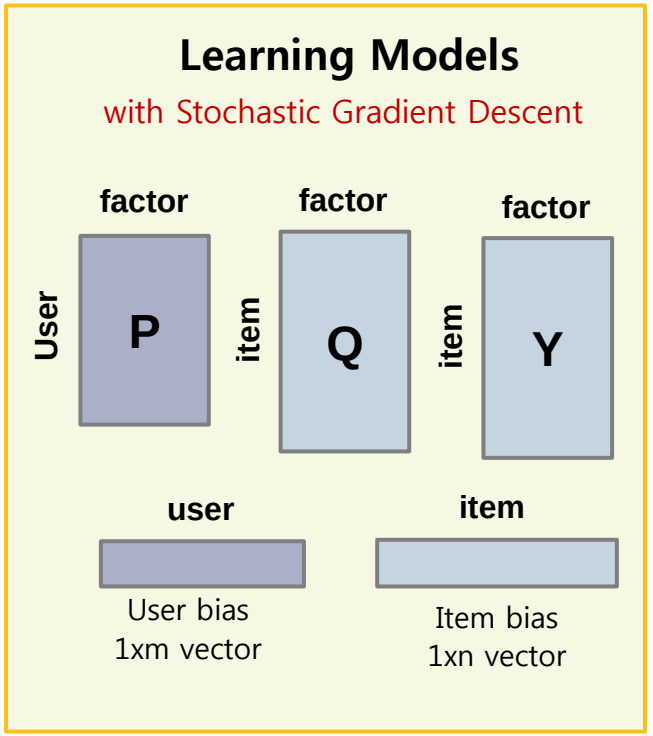
Global average | User bias | Item bias | User-item interaction

Implicit feedback이 반영된 biased MF (**SVD++**)

$$r'_{ui} = \mu + b_u + b_i + \mathbf{q}_i^{\mathrm{T}} \left(\mathbf{p}_u + \frac{1}{\sqrt{|I(u)|}} \sum_{j \in I(u)} \mathbf{y}_j \right)$$

사용자 u의 모델: explicit + implicit

$I(u)$: 사용자 u가 rate한 아이템 집합



TrustSVD

- Incorporating *trust influence* into SVD++ model
  - Consider simultaneously implicit effect of trusted users on item ratings

$$r'_{ui} = \mu + b_u + b_i + \mathbf{q}_i^{\mathrm{T}} \left(\mathbf{p}_u + \frac{1}{\sqrt{|I(u)|}} \sum_{j \in I(u)} \mathbf{y}_j + \frac{1}{\sqrt{|T(u)|}} \sum_{v \in T(u)} \mathbf{w}_v \right)$$

SVD++

Adding trusted users' effect

$T(u)$: a set of trusted users of user u

$$\mathbf{p}_u + \frac{1}{\sqrt{|I(u)|}} \sum_{j \in I(u)} \mathbf{y}_j + \frac{1}{\sqrt{|T(u)|}} \sum_{v \in T(u)} \mathbf{w}_v$$

User u is modeled by

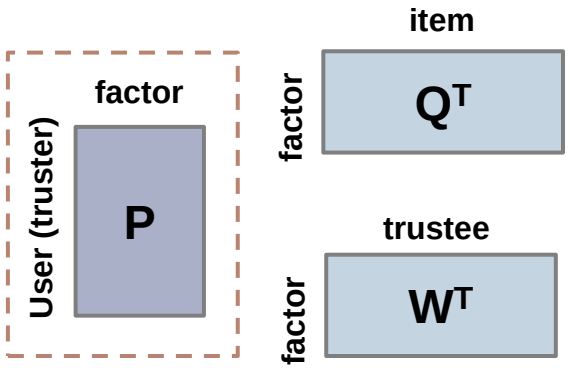
1. explicit rating $\mathbf{p}_u$
2. implicit feedbacks
   - → items that she previously rated. $\mathbf{y}_j$
   - → users (i.e., trustees) trusted by her. $\mathbf{w}_v$

TrustSVD

- Original objective function to minimize (like SVD++)

$$\sum_{r_{ui} \in \mathbf{R}} (r_{ui} - r'_{ui})^2 + \lambda \underbrace{\left(b_u^2 + b_i^2 + \|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2 + \|\mathbf{y}_j\|^2 + \|\mathbf{w}_v\|^2 \right)}_{\text{Regularization}}$$

- Bridging *rating matrix* and *trust matrix*



Sharing user features

New objective function to minimize

Optimization using gradient descents

$$\sum_{r_{ui} \in \mathbf{R}} (r_{ui} - r'_{ui})^2 + \lambda_T \sum_{t_{uv} \in \mathbf{T}} (t_{uv} - \mathbf{p}_u \mathbf{w}_v^{\mathrm{T}})^2$$

with (*inverse*) **weighted-λ-regularization**

- less penalties (regularization) for popular users and items
- more penalties (regularization) for cold start users and items

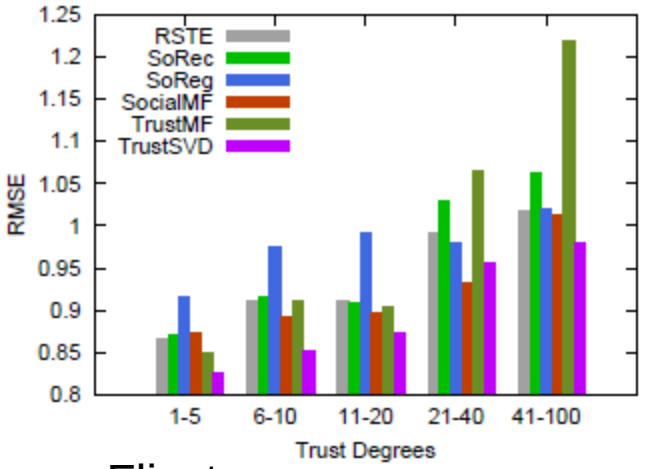
Comparison with Matrix Factorization Methods

Statistics of data sets

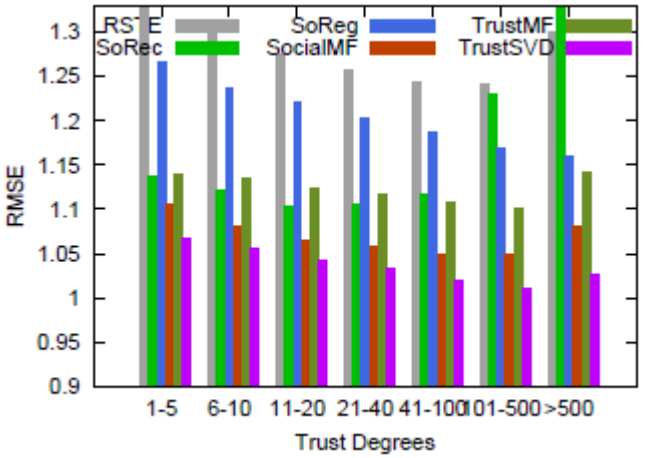
Feature	Epinions	FilmTrust
users	40,163	1,508
items	139,738	2,071
ratings	664,824	35,497
density	0.051%	1.14%
trusters	33,960	609
trustees	49,288	732
trusts	487,183	1,853
density	0.029%	0.42%

Feature	Flixster	Ciao
users	53,213	7,375
items	18,197	99,746
ratings	409,803	280,391
density	0.04%	0.03%
trusters	47,029	6,792
trustees	47,029	7,297
trusts	655,054	111,781
density	0.03%	0.23%

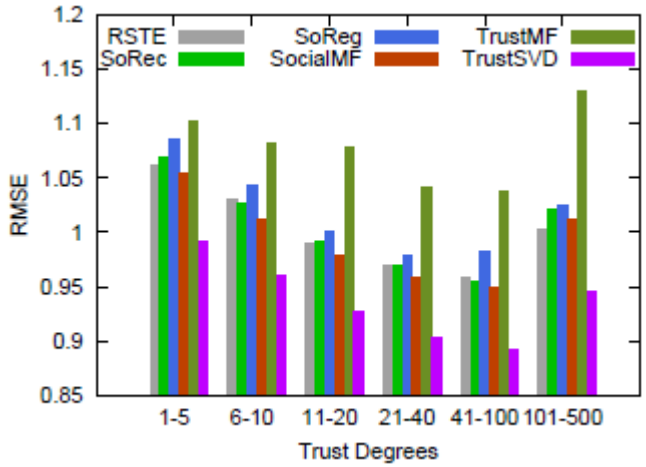
FilmTrust



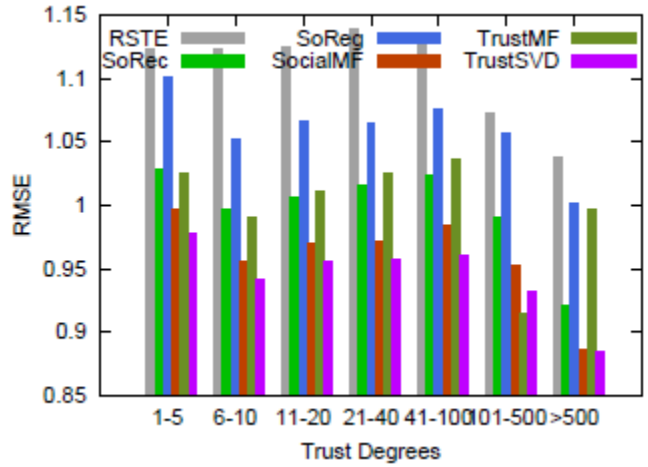
Epinions



Flixster



Ciao



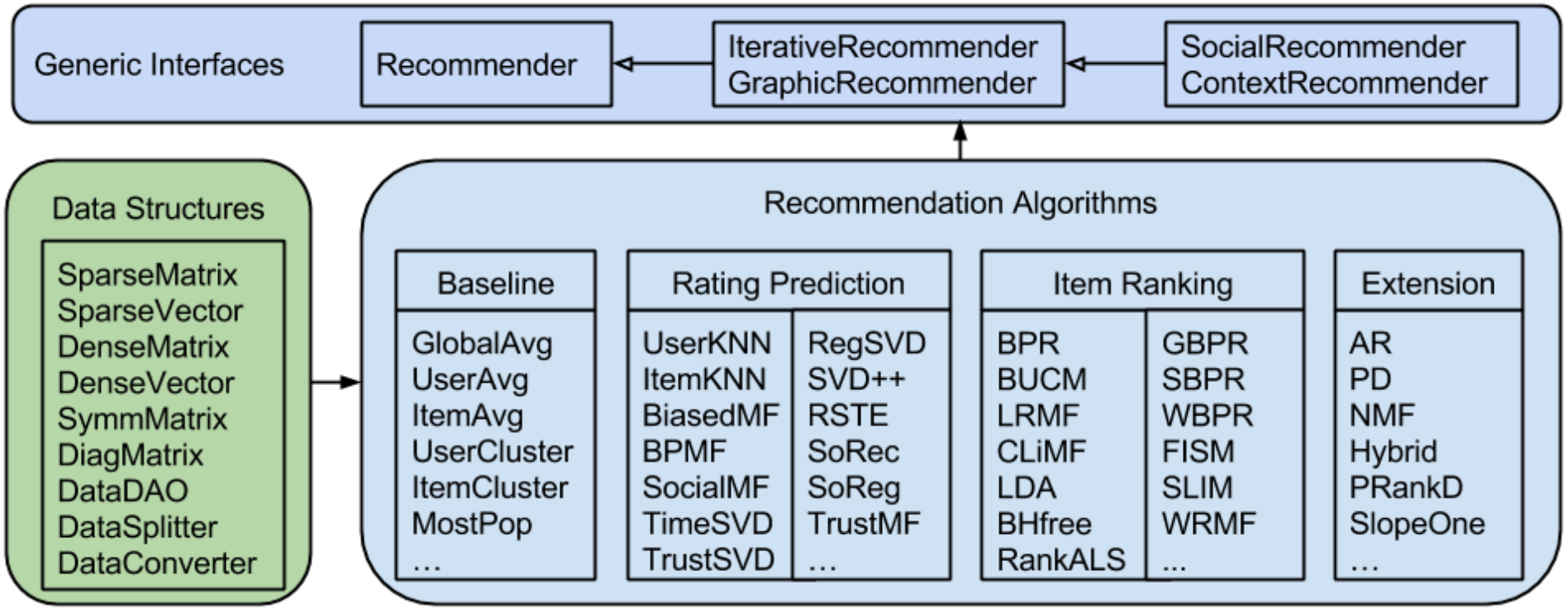
(from Guo et al., AAAI, 2015)

Appendix.
Online Resources

TARS Library

Librec: A Java Library for Recommender Systems

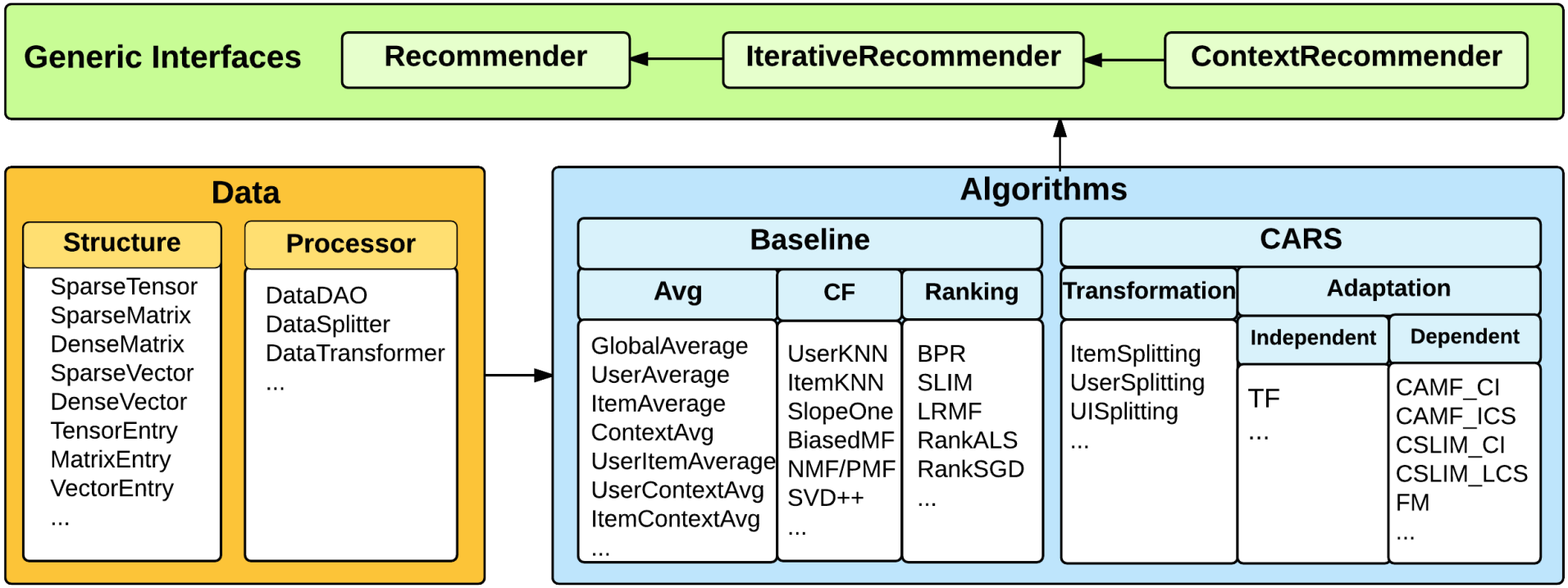
http://librec.net/



CARS Library

CARSKit: A Java-based Open-source Context-aware Recommendation Library

https://github.com/irecsys/CARSKit



TARS Data Sets

Data Set	Site Link	Ratings				Trust/Social networks		
		Users	Items	Ratings (Scale)		Users	Links (Type)	
CiaoDVD	Link	7,375	99,746	278,483	[1, 5]	7,375	111,781	Trust
Douban	Link	129,490	58,541	16,830,839	[1, 5]	129,490	1,692,952	Friendship
Epinions (665K)	Link	40,163	139,738	664,824	[1, 5]	49,289	487,183	Trust
Epinions (510K)	Link	71,002	104,356	508,960	[1, 5]			Trust
Epinions (Extended)	Link	120,492	755,760	13,668,320	[1, 5]			Trust/Distrust
Flixster[1)]	Link	147,612	48,794	8,196,077	[0.5, 5.0]	787,213	11794648	Friendship
FilmTrust	Link	1,508	2,071	35,497	[0.5, 4.0]	1,642	1,853	Trust

1) site is not working

http://www.librec.net/datasets.html#list

CARS Data Sets

Most of data sets are very small size (except for Frappe)

- Food: AIST Japan Food, Mexico Tijuana Restaurant Data
- Movies: AdomMovie, DePaulMovie, LDOS-CoMoDaData
- Music: InCarMusic
- Travel: TripAdvisor, South Tyrol Suggests
- Mobile: Frappe

available at https://github.com/irecsys/CARSKit/tree/master/context-aware_data_sets

Conferences related to RecSys/CARS/TARS

- **ACM RecSys (Recommender Systems)**
- **ACM UMAP (User Modeling, Adaptation and Personalization)**
- ACM IUI (Intelligent User Interfaces)

주제 비중이 높음

- WWW
- AAAI
- SIGIR
- WI (Web Intelligence)
- ICDM
- Others (AI/data mining/machine learning conferences)

비중이 매우 높은 편은 아니나 관련 논문이 꾸준히 발표됨

Thank you!

Heung-Nam Kim
LG Electronics
nami4596@gmail.com