Recommender Systems

Jee-Hyong Lee

Information & Intelligence System Lab.
Department of Computer Science & Engineering
Sungkyunkwan University

Unique Origin Unique Future

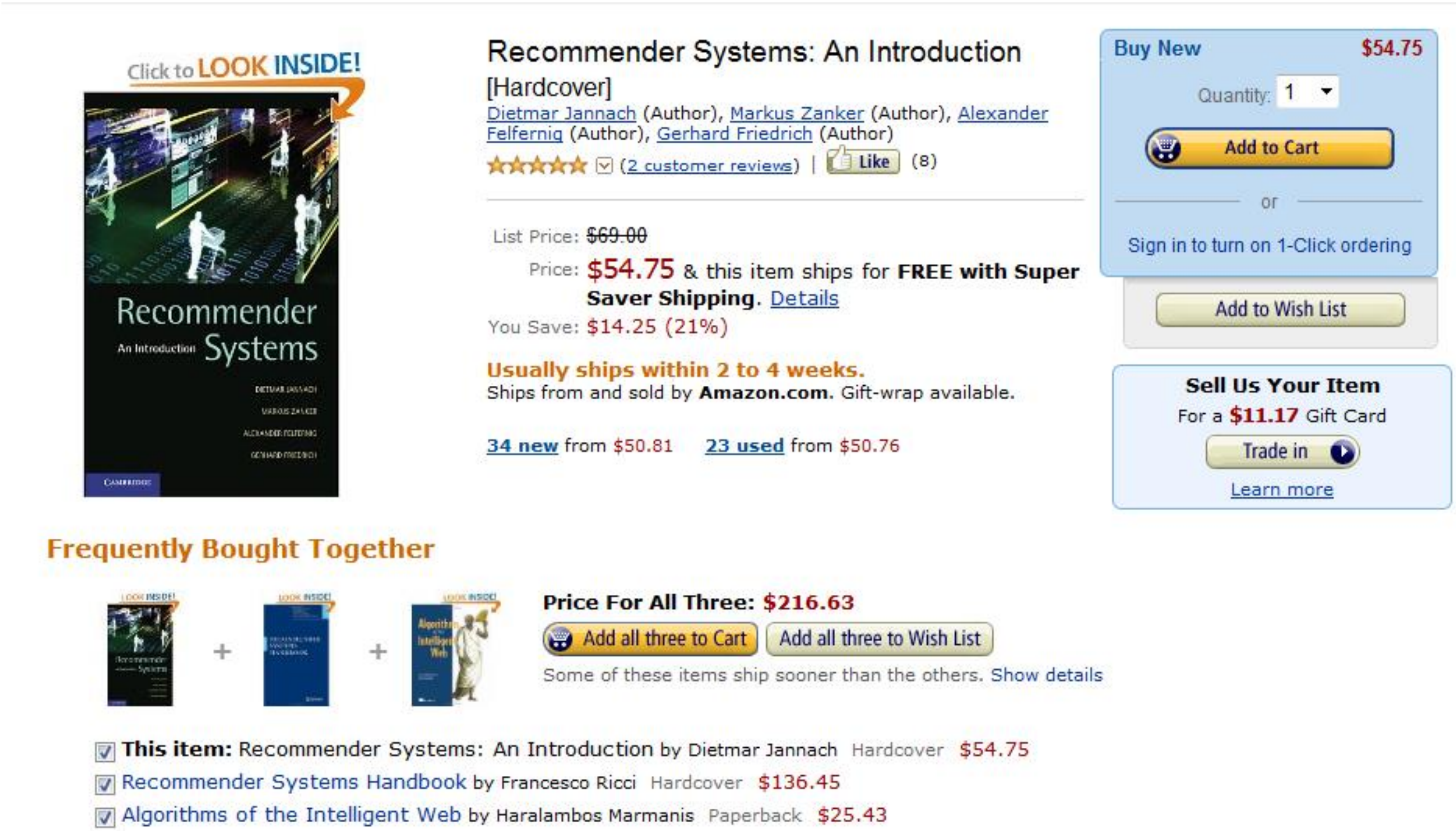
Outline

1. Introduction
2. Collaborative Filtering
3. Content-based Recommendation
4. Context-aware Recommendation
5. Other Approaches
6. Concluding Remarks

1. Introduction

2. Collaborative Filtering
3. Content-based Recommendation
4. Context-aware Recommendation
5. Other Approaches
6. Concluding Remarks

Recommender Systems



Recommender Systems

- **Netflix:**
  - 2/3 of the movies watched are recommended
- **Google News:**
  - Recommendations generate 38% more clickthrough
- **Amazon:**
  - 35% sales from recommendations
- **Choicestream:**
  - 28% of the people would buy more music if they found what they liked

Definition of Recommender Systems

- **Given**
  - User profile (usage history, demographics, …)
  - Items (with or without additional information)

- **Goal**
  - Relevance scores of unseen items
  - List of unseen items

- **By using a number of technologies**
  - Information Retrieval: document models, similarity, ranking
  - Machine Learning & Data Mining: classification, clustering, regression, probability, association
  - Others: user modeling, HCI

Approaches

- **Collaborative Filtering**
  - Memory based CF
    - User-based CF, Item-based CF
  - Model based CF
    - Dimension reduction, Clustering, Association rules, restricted Boltzmann machine, Probabilistic approach, Other classifiers
- **Content-based Recommendation**
  - Content/User modeling & similarity
    - TF-IDF, Cosine similarity
- **Context-aware Recommendation**
  - Pre-filtering, Post-filtering
  - Contextual modeling
    - Extension of 2D model, Tensor factorization

Approaches

- **Other Approaches**
  - Combining Multiple Recommendation Approach
  - Combining Multiple Information
    - Hybrid Information Network based CF
    - Collective matrix factorization
  - Diversity in Recommendation
  - Division of Profiles into Sub-Profiles
  - Recommendation for group users

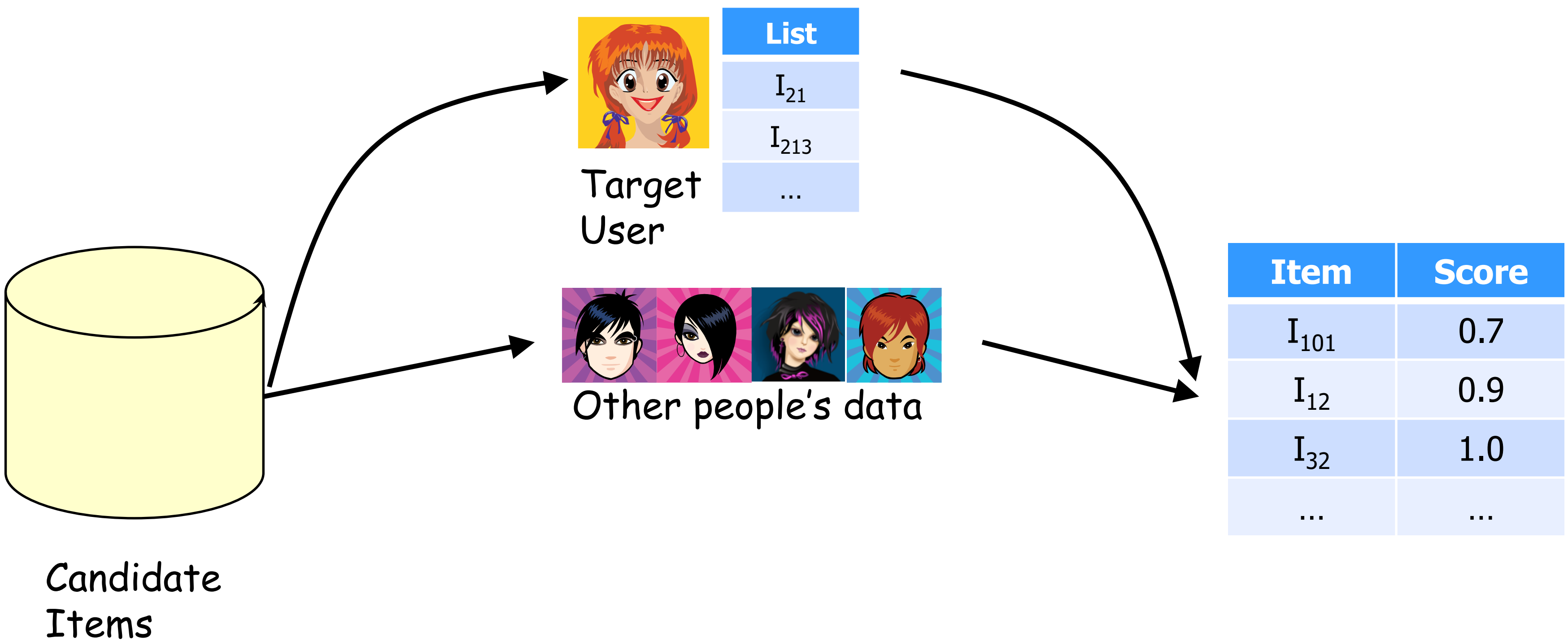
1. Introduction

2. Collaborative Filtering

3. Content-based Recommendation
4. Context-aware Recommendation
5. Other Approaches
6. Concluding Remarks

Overview

- **Collaborative Filtering**



List
I_{21}
I_{213}
…

Item	Score
I_{101}	0.7
I_{12}	0.9
I_{32}	1.0
…	…

Overview

- **Basic assumption and idea**
  - Customers who had similar tastes in the past, will have similar tastes in the future
  - Implicit or explicit user ratings to items are available

- **Easy to apply any domain**
  - Based on big data: commercial e-commerce sites
  - Easy to explain: wisdom of the crowd
  - Flexible: various algorithms exist
  - Example: book, movies, DVDs, ..

Collaborative Filtering

- **Memory based (k-NN approach)**
  - User-based CF
  - Item-based CF
- **Model based (User model construction)**
  - Dimension reduction (Matrix Factorization)
  - Clustering
  - Association rule mining
  - Restricted Boltzmann machine
  - Probabilistic models
  - Various machine learning approaches

User-based Collaborative Filtering

- **How much target user likes I3?**

	I1	I2	I3	I4	I5
Active	4	3	?	5	4
U1	2	2	2	3	3
U2	3	2	4	5	4
U3	2	3	3	2	5
U4	1	5	1	4	2

  – Predict the ratings of active user based on the ratings of similar users

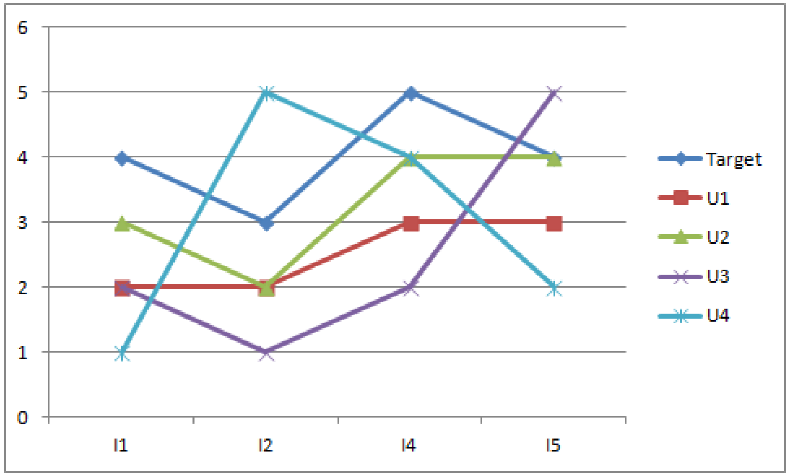
User-based Collaborative Filtering

- **User Similarity**

$$sim(u_1,u_2)=\frac{\sum_{i\in I}\left(r_{u_1,i}-\bar{r}_{u_1}\right)\left(r_{u_2,i}-\bar{r}_{u_2}\right)}{\sqrt{\sum_{i\in I}\left(r_{u_1,i}-\bar{r}_{u_1}\right)^2}\sqrt{\sum_{i\in I}\left(r_{u_2,i}-\bar{r}_{u_2}\right)^2}}$$

  - $r_{u,i}$: rating of user **u** for item **i**
  - $\bar{r}_u$: user **u**'s average ratings

	I1	I2	I3	I4	I5
Active	4	3	?	5	4
U1	2	2	2	3	3
U2	3	2	4	5	4
U3	2	3	3	2	5
U4	1	5	1	4	2



User-based Collaborative Filtering

- **Prediction**

$$pred(u,i) = \bar{r}_u + \frac{\sum_{v \in U} sim(u,v) \cdot (r_{v,i} - \bar{r}_v)}{\sum_{v \in U} sim(u,v)}$$

	I1	I2	I3	I4	I5	Sim.
Active	4	3	?	5	4	
U1	2	2	2	3	3	**0.71**
U2	3	2	4	5	4	**0.85**
U3	2	3	3	2	5	**0.24**
U4	1	5	1	4	2	**-0.22**

$$pred(\text{Target}, \text{I3}) = 0.43$$

User-based Collaborative Filtering

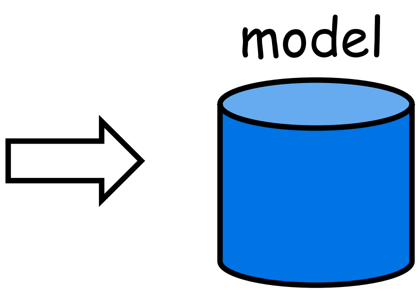
- **Some Problems**
  - Sparsity
    - Large item sets: users purchases are under 1%
    - Few common ratings between two users
    - Reliability of user-user similarity decreases
  - Scalability (m = |users|, n = |items|)
    - Large computation for finding NNs
    - Time complexity for computing Pearson $O(m^2n)$
    - Space complexity $O(m^2)$ for pre-computing
  - Solution
    - Model-based CF

Model-based Collaborative Filtering

- **Lazy Learning vs Eager Learning**
  - Lazy learning: User/Item-based collaborative filtering
  - Eager learning: Model-based collaborative filtering

- **Model-based CF**
  - Build preference model from rating matrix
  - Use the models for predictions
  - Possibly computationally expensive

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
abnormalities	0	0	0	0	0	0	0	1	0	1	0	0	0	0
age	1	0	0	0	0	0	0	0	0	0	0	1	0	0
behavior	0	0	0	0	1	1	0	0	0	0	0	0	0	0
blood	0	0	0	0	0	0	0	1	0	0	1	0	0	0
close	0	0	0	0	0	0	1	0	0	0	1	0	0	0
culture	1	1	0	0	0	0	0	1	1	0	0	0	0	0
depressed	1	0	1	1	1	0	0	0	0	0	0	0	0	0
discharge	1	1	0	0	0	1	0	0	0	0	0	0	0	0
disease	0	0	0	0	0	0	0	0	1	0	1	0	0	0
fast	0	0	0	0	0	0	0	0	0	1	0	1	1	1
generation	0	0	0	0	0	0	0	0	1	0	0	0	1	0
oestrogen	0	0	1	1	0	0	0	0	0	0	0	0	0	0
patients	1	1	0	1	0	0	0	1	0	0	0	0	0	0
pressure	0	0	0	0	0	0	0	0	0	0	1	0	0	1
rats	0	0	0	0	0	0	0	0	0	0	0	0	1	1
respect	0	0	0	0	0	0	0	1	0	0	0	1	0	0
rise	0	0	0	1	0	0	0	0	0	0	0	0	0	1
study	1	0	1	0	0	0	0	0	1	0	0	0	0	0



Model-based Collaborative Filtering

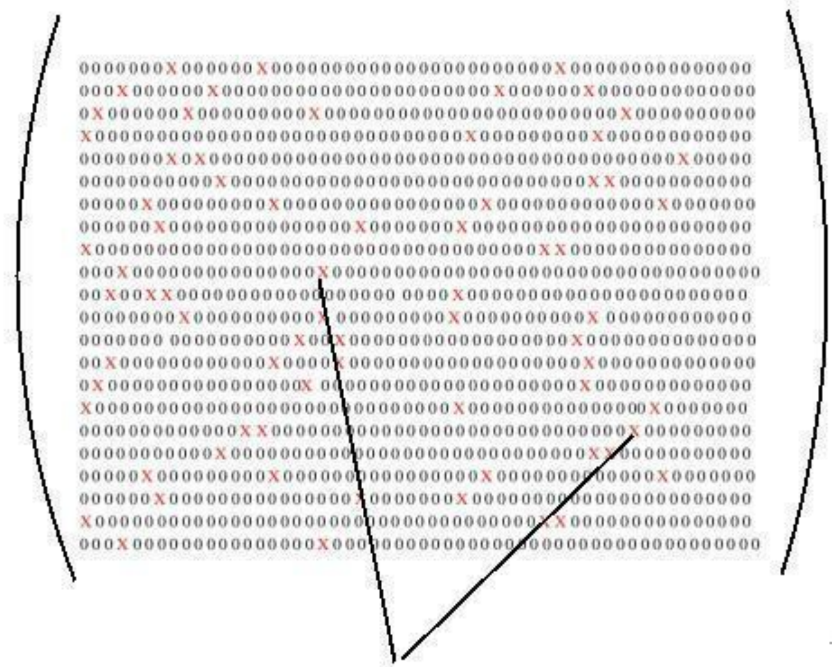
- **Basic Techniques**
  - Dimension reduction (Matrix Factorization)
  - Clustering
  - Association rule mining
  - Restricted Boltzmann machine
  - Probabilistic models
  - Various machine learning approaches

Matrix Factorization

- **Netflix 100M data**
  - Possibly 8,500M ratings (500,000 x 17,000)
  - But, there are only 100 M non-zero ratings

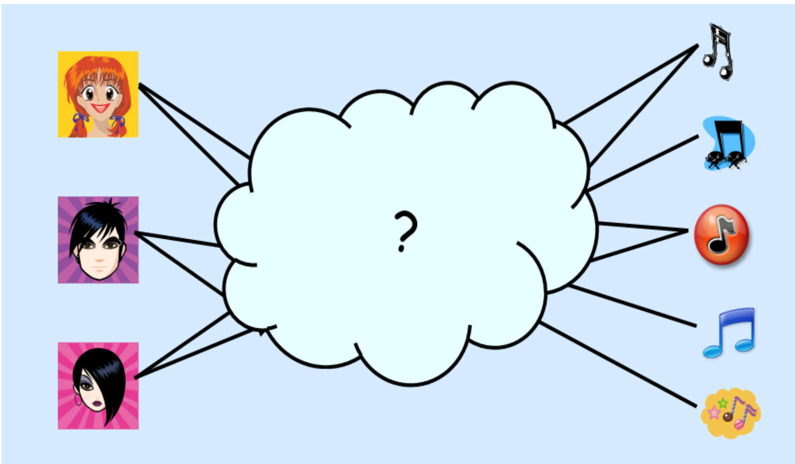
- **Methods of dimensionality reduction**
  - Matrix Factorization
  - Clustering
  - Projection (PCA…)

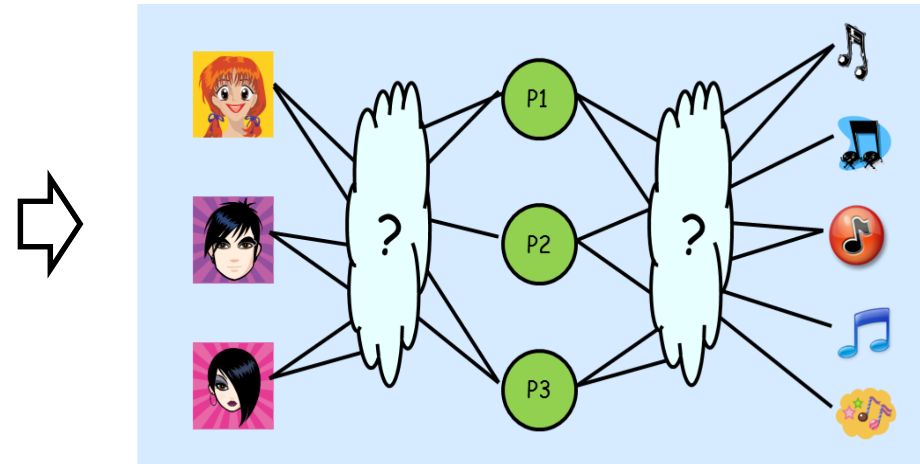
- **Space complexity**
  - Worst case: O(mn)
  - In practice: O(m + n)



Matrix Factorization

- **Assume some latent factors in user preference**





Matrix Factorization

- **Singular Value Decomposition**

$$
\underset{m \times n}{\overset{X}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} = \underset{m \times r}{\overset{U}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}}} \underset{r \times r}{\overset{S}{\begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}}} \underset{r \times n}{\overset{V^{\mathsf{T}}}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}}
$$

  - Obtain $\hat{X}$, a low-rank approximation of X
  - Purpose
    - To predict ratings by capturing latent relationships
    - To compute neighborhood in a low-dimensional representation

Matrix Factorization

- **Predict ratings by capturing latent relationships**
  - User-item matrix : x
  - Normalize x by subtracting b from x : y
    - to efficiently capture the interaction effect between users and items

$$b_{u,i} = \mu + b_u + b_i \qquad b_u = \frac{1}{|I_u|}\sum_{i \in I_u}(r_{u,i} - \mu)$$

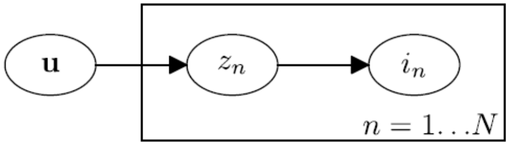
$$b_i = \frac{1}{|U_i|}\sum_{u \in U_i}(r_{u,i} - b_u - \mu)$$

  - Obtain a row-rank approximation of y: $\hat{y}$
  - Predict ratings for item *i* and user *u* as follows

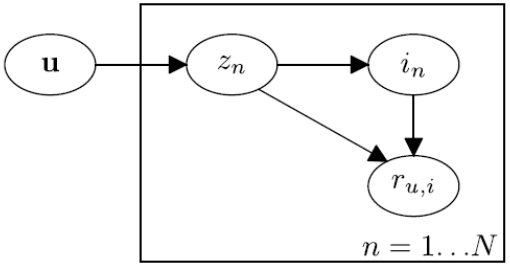
$$\hat{r}_{u,i} = b_{u,i} + \hat{y}_{u,i}$$

Matrix Factorization

- **Probabilistic Matrix Factorization**
  - PLSA (Probabilistic Latent Semantic Analysis)

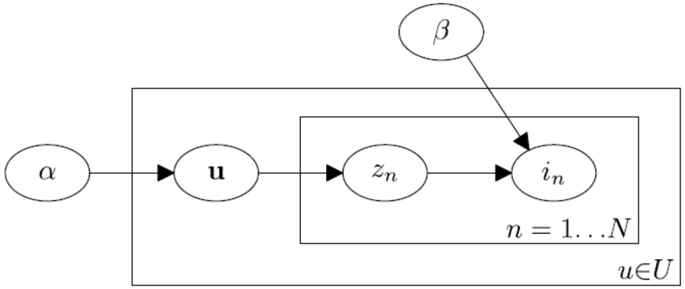


User purchase model



User rating model

  - LDA (Latent Dirichlet Allocation)



Matrix Factorization

- **Probabilistic Latent Semantic Analysis**
  - Interpreting as probabilities of user-item

$$p(i|u) = \sum_{z} p(i|z)p(z|u)$$

  - Decompose the probability matrix ***P*** using an EM approach

$$\boldsymbol{P} = \widehat{\boldsymbol{U}}\boldsymbol{\Sigma}\widehat{\boldsymbol{T}}^{\boldsymbol{T}}$$

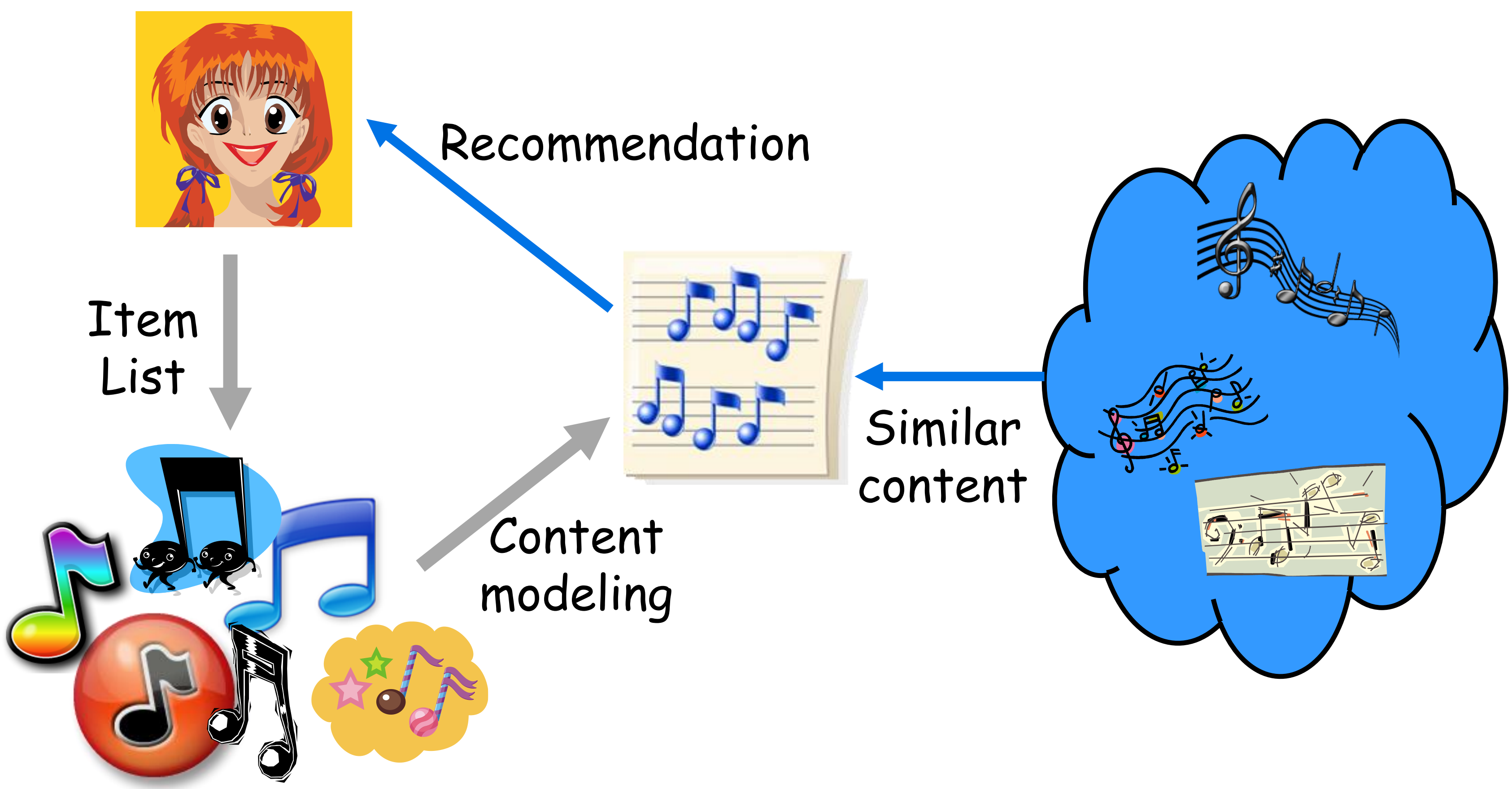
  - Comparison to SVD
    - SVD :minimizing error, decomposition with geometric model
    - PLSA : maximizing the predictive power, decomposition with stochastic model

Collaborative Filtering

- **Pros**
  - Requires minimal knowledge engineering efforts
  - No need of any internal structure or characteristics
- **Cons**
  - Requires a large number of reliable ratings
  - Assumes that prior behavior determines current behavior
  - Cold start problems: New user, new items
  - Sparsity problems

1. Introduction
2. Collaborative Filtering
3. **Content-based Recommendation**
4. Context-aware Recommendation
5. Other Approaches
6. Concluding Remarks

Overview


Recommendation
Item
List
Content
modeling
Similar
content

Overview

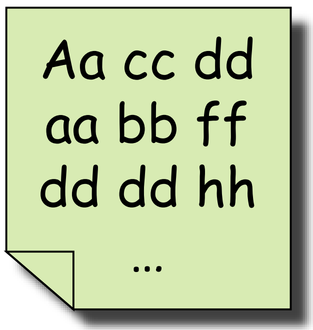
- **What's content?**
  - Explicit attributes or chracteristics (Eg for a movie)
    - Genre : Action / adventure
    - Feature : Bruce Willis
    - Year : 1995
  - Textual content (Eg for a book)
    - Title
    - Description
    - Table of content
  - **Any features or keywords which can describe items**

Overview

- **Basic assumption and idea**
  - Customers will like similar content which they liked in the past
- **Suitable for text-based products (web pages, book)**
  - Items are "described" by their features (e.g. keywords)
  - Users are described by the keywords in the items they bought
- **Characteristic**
  - Easy to apply to text-based products or products with text description
  - Based on match between the content (item keywords) and user keywords
  - Many machine learning approaches are applicable
    - Neural Networks, Naive Bayesian, Decision Tree, …

Content/User Modeling

- **User Modeling (for documents)**
  - Usually, bag of words model is adopted



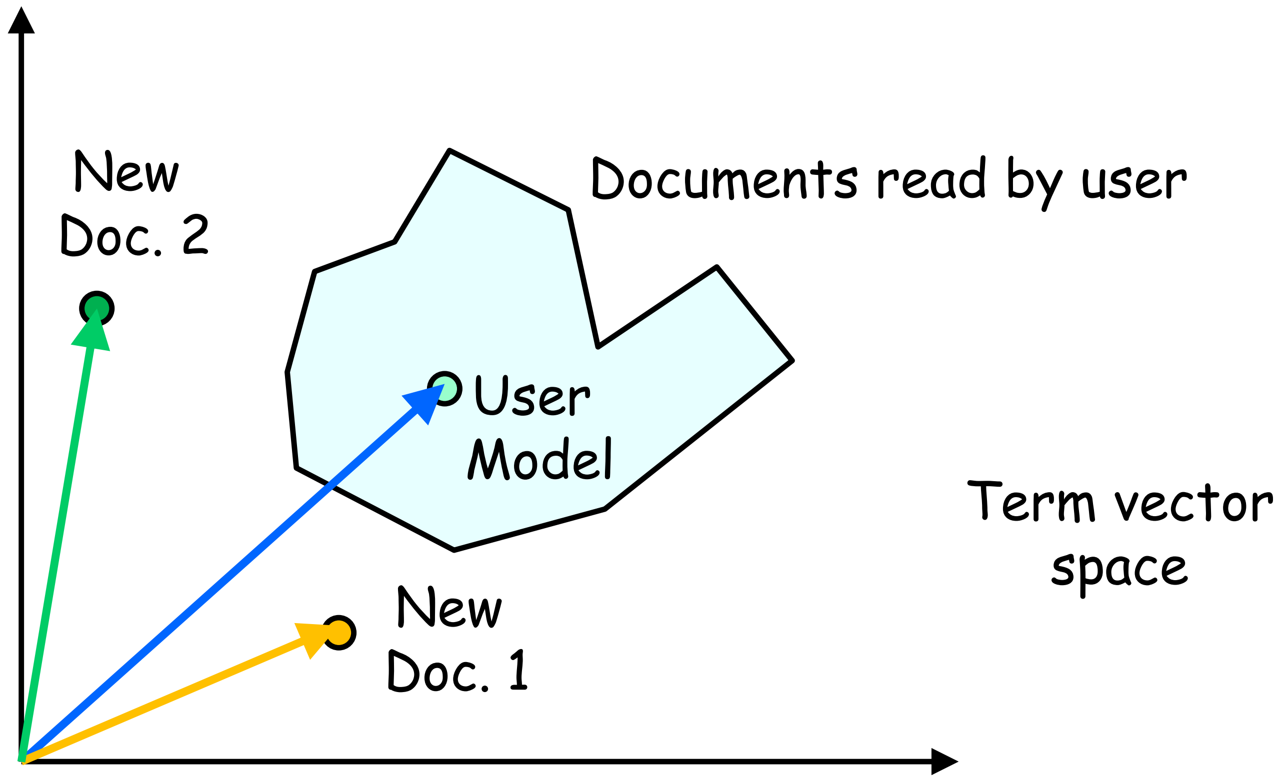
(aa, bb, cc, dd, ee, ff, gg, hh, …)

(2, 1, 1, 2, 0, 1, 0, 1, …)

  - Some important words can be selected
    - Based on Entropy or TF-IDF
  - User Modeling
    - Average of term vectors of documents in user profile

Content-User Matching

- **Similarity measure based**
  - Cosine similarity



Advantages of CBR

- **No need for data on other users**
  - No first-rater problem or sparsity problems
  - Able to recommend new and unpopular items
- **Able to recommend to users with unique preference**
- **Can provide explanations why it is recommended**
  - by listing content-features that caused an item to be recommended
- **Good to dynamically created items**
  - News, email, events, etc.

Disadvantages of CBR

- **Not easy to create content model for any products**
  - Book, web pages, news articles, music, video
- **Over-specialization**
  - Users are recommended with items similar to what they watched
  - no serendipity

Overview

- **Traditional Recommendations**
  - Are based on the ratings of user *u* for item *i*
  - Cumulate data of (User, Items, Rating)
  - Build a relation *R: Users* × *Items* → *Rating*, in order to estimate ratings for unseen items of a user
    - Two-dimensional recommendation framework

- **Extension for Recommendations with Context**
  - Data: <*user, item, rating, context*>
  - Relation: *Users* × *Items* × *Context*→ *Rating*
    - Three-dimensional recommendation framework

Overview

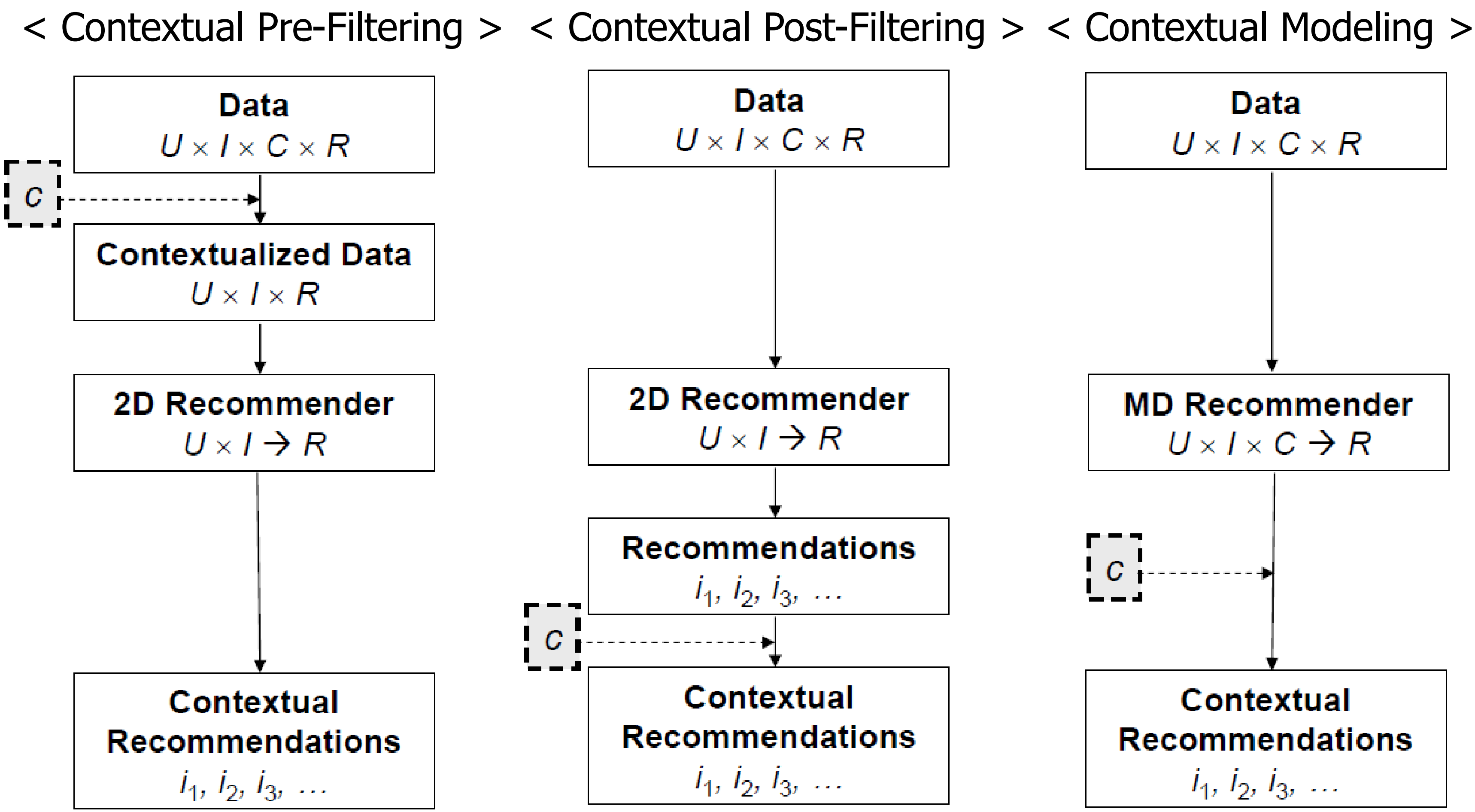
- **What context is**

> Context is any information or conditions that can influence the perception of the usefulness of an item for a user

- **Additional information**
  - Except users and items
  - Can be used for better recommendations

- **Example: Which context is helpful for recommending a book?**
  - For what purpose is the book bought? (Work, leisure, …)
  - When will the book be read? (Weekday, weekend, …)
  - Where will the book be read? (At home, at school, on a plane, …)

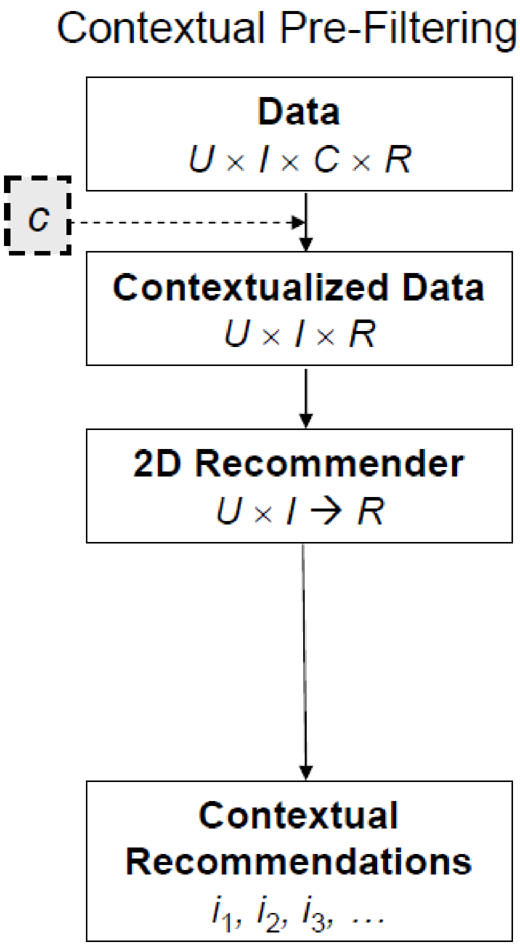
Architectural Models of Context Integration



Contextual Pre-Filtering

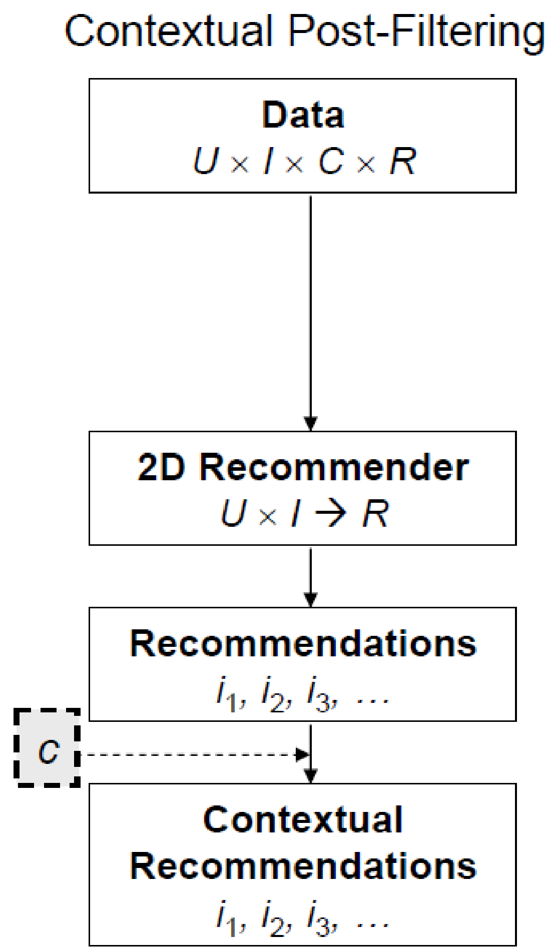
- **Steps**
  - Select the relevant data using given context
  - Generate recommendation based on the selected data using traditional recommendation approach

- **Issues**
  - How to efficiently extract relevant data
  - Exact filtering vs. Generalized filtering



Contextual Post-Filtering

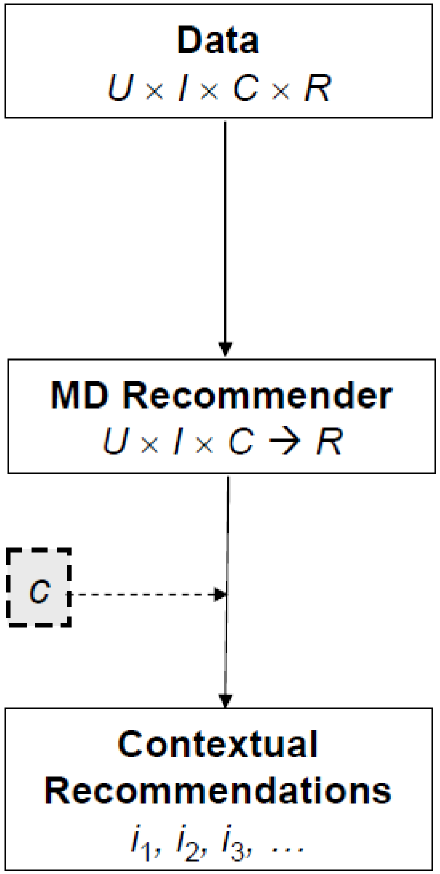
- **Overview**
  - Convert into two-dimensional data (drop out the context information)
  - Build two models
- **Steps**
  - Generate recommendation by the traditional recommendation approach
  - Adjust the obtained recommendation using contextual information
- **Issues**
  - How to adjust the recommendation
  - How to apply generalized context



Contextual Modeling

- **Based on the three-dimensional model**
- **Directly incorporating contextual information into the recommendation model**
  - Three-dimensional model
  - Rating = f (User, Item, Context)
- **Issues**
  - How to efficient build a model
  - How to apply generalized context

Contextual Modeling



Contextual Modeling

- **How to model three-dimensional information**

$$Users \times Items \times Context \rightarrow Rating$$

  - Extension of two-dimensional models
  - Tensor factorization (like SVD)

Extension of two-dimensional models

- **Extension of two-dimensional model**

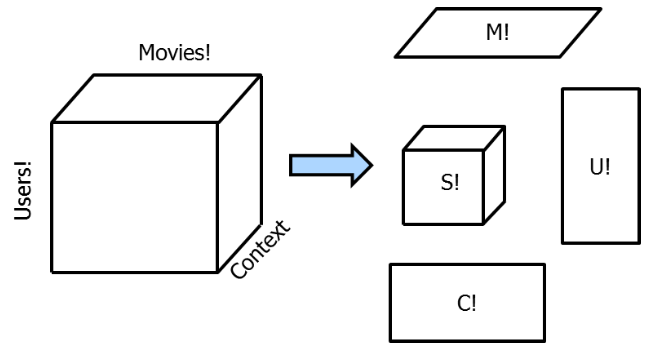
$$pred(u,i,c) = \bar{r}_{u,c} + \frac{\sum_{v \in U, k \in C} sim((u,c),(v,k)) \cdot (r_{v,i,k} - \bar{r}_{v,k})}{\sum_{v \in U, k \in C} sim((u,c),(v,k))}$$

  - Traditional user-based collaborative filtering:

$$pred(u,i) = \bar{r}_{u_1} + \frac{\sum_{v \in U} sim(u,v) \cdot (r_{v,i} - \bar{r}_v)}{\sum_{v \in U} sim(u,v)}$$

Tensor Factorization

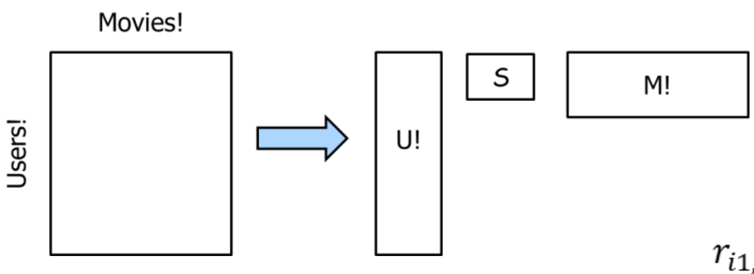
- **Also called HOSVD (High Order SVD)**



$$U \in \mathbb{R}^{n \times d_U},\ M \in \mathbb{R}^{m \times d_M} \text{ and } C \in \mathbb{R}^{c \times d_C}$$

$$S \in \mathbb{R}^{d_U \times d_M \times d_C}$$

$$r_{i1,i2,i3} = \sum_{j1} \sum_{j2} \sum_{j3} s_{j1,j2,j3} \times m_{i1,j2} \times u_{i1,j2} \times c_{j3,j3}$$



$$r_{i1,i2,i3} = \sum_{j1} \sum_{j2} s_{j1,j2} \times m_{i1,j1} \times u_{i1,j2}$$

Tensor Factorization

- **Optimization**
  - Loss function

$$L(F,Y) := \frac{1}{\|S\|_1} \sum_{i,j,k} D_{ijk} l(F_{ijk}, Y_{ijk}) \qquad l(f,y) = |f-y|$$

  - Regularization

$$\Omega[U,M,C] := \frac{1}{2}\left[\lambda_U \|U\|^2_{\text{Frob}} + \lambda_M \|M\|^2_{\text{Frob}} + \lambda_C \|C\|^2_{\text{Frob}}\right]$$

$$\Omega[S] := \frac{1}{2}\left[\lambda_S \|S\|^2_{\text{Frob}}\right]$$

  - Objective function

$$R[U,M,C,S] := L(F,Y) + \Omega[U,M,C] + \Omega[S]$$

Context-aware Recommendation

- **Pre-filtering**
  - Simple: using only the ratings in the same context
  - Works with large amounts of data
    - Increases sparseness
- **Post-filtering**
  - Simple: Averaging ratings under different context
  - Takes into account context interactions
    - Computationally expensive
- **Contextual modeling**
  - Extension of 2-D model
    - How to extend considering context
  - Tensor Factorization
    - Performance, Linear scalability

1. Introduction
2. Collaborative Filtering
3. Content-based Recommendation
4. Context-aware Recommendation

5. Other Approaches

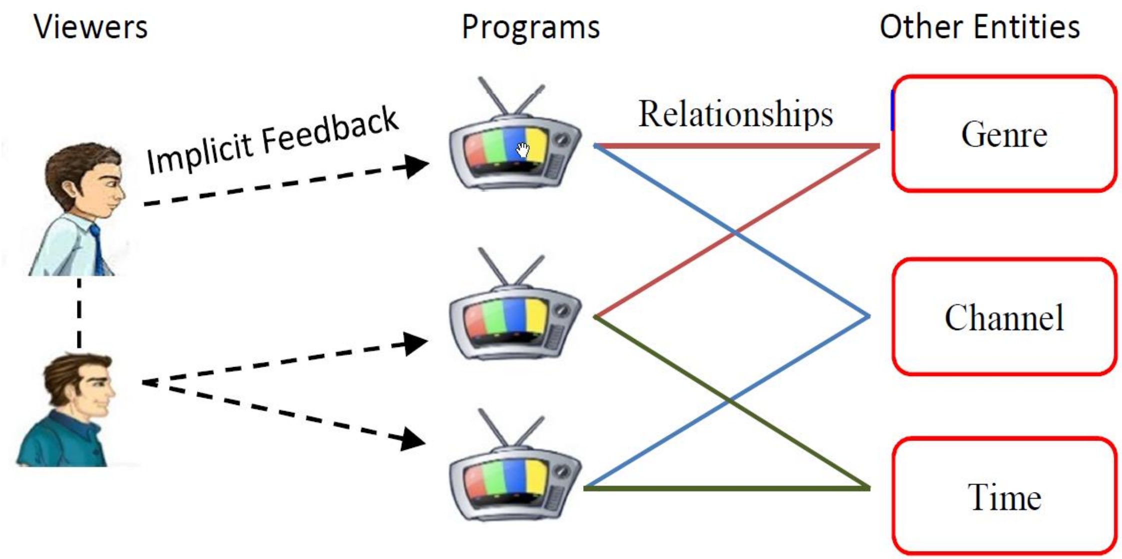
6. Concluding Remarks

Overview

- **Combining Multiple Information**
  - Hybrid Information Network based CF
  - Collective matrix factorization
- **Recommendation for group users**
  - Group profile based
  - Consensus function based

Combining Multiple Information

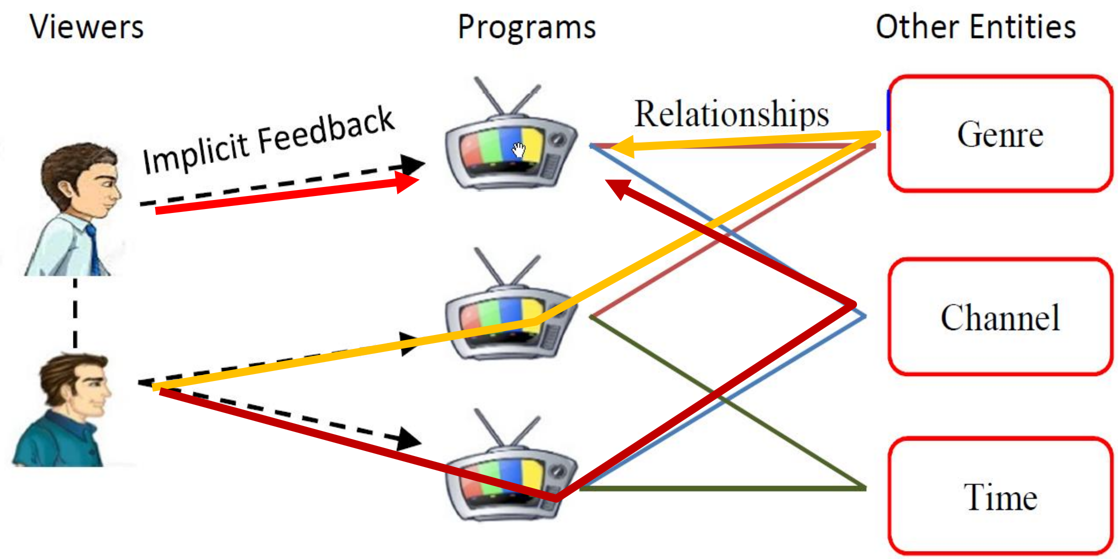
- **There are many kinds of information**
  - User-user relation
  - User-program relation
  - Program-genre/channel/time relations



- **Why do we use only user-program relation?**

Combining Multiple Information

- **Hybrid Information Network based CF**
  - Evaluate user-user similarity through multiple path
  - Recommend based on user-based CF



Combining Multiple Information

- **Hybrid Information Network based CF**
  - Predicted rating

$$r(u,e') = \sum_{P \in P^*} \theta_{u,P} \cdot s(u,e' \mid P)$$

    - Predicted ratings given path *P*

$$s(u,\mathrm{e}' \mid P) = \sum_{e \in I_u} \frac{2 \times W_{u,e} \times |\{p_{e \to \mathrm{e}'} : p_{e \to \mathrm{e}'} \in P'\}|}{|\{p_{e \to e} : p_{e \to e} \in P'\}| + |\{p_{\mathrm{e}' \to \mathrm{e}'} : p_{\mathrm{e}' \to \mathrm{e}'} \in P'\}|}$$

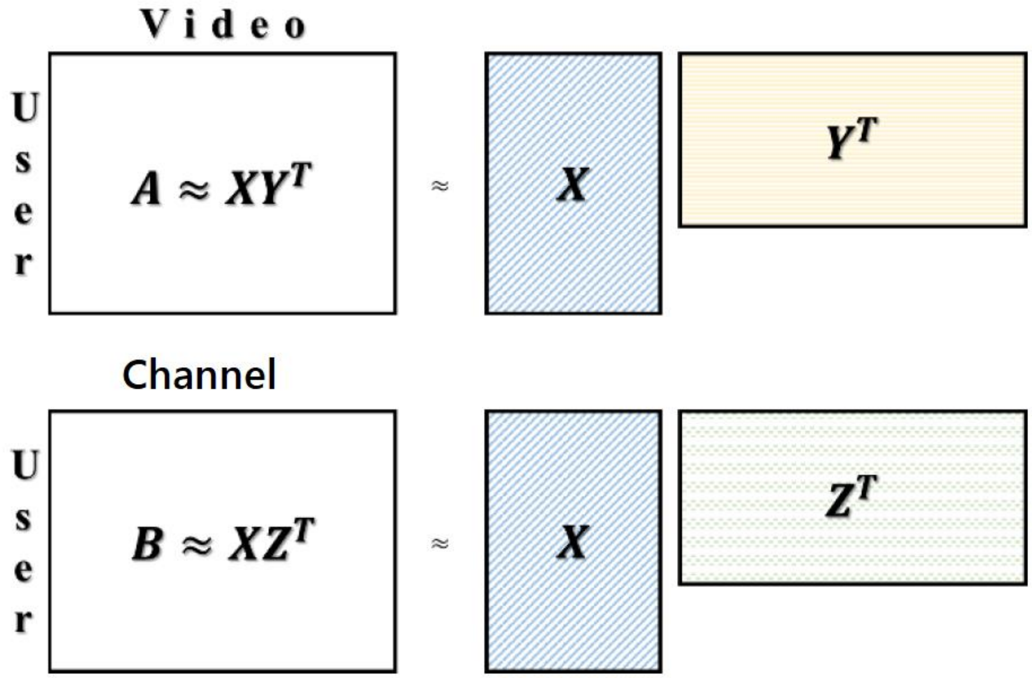
    - Normalized weight & weight of path *P* for *u*

$$\theta_{u,P} = \frac{Weight(u \mid P)}{\sum_{Q \in P^*} Weight(u \mid Q)}$$

$$Weight(u \mid P) = \sum_{e \in I_u} W_{u,e} \cdot s(u,e \mid P)$$

Combining Multiple Information

- **Collective Matrix Factorization**



$$L(X,Y,Z) = \frac{1}{2}\|I_1 \circ (A - XY^T)\|_F^2 + \frac{\alpha}{2}\|I_2 \circ (B - XZ^T)\|_F^2 + \frac{\beta}{2}(\|X\|_F^2 + \|Y\|_F^2 + \|Z\|_F^2),$$

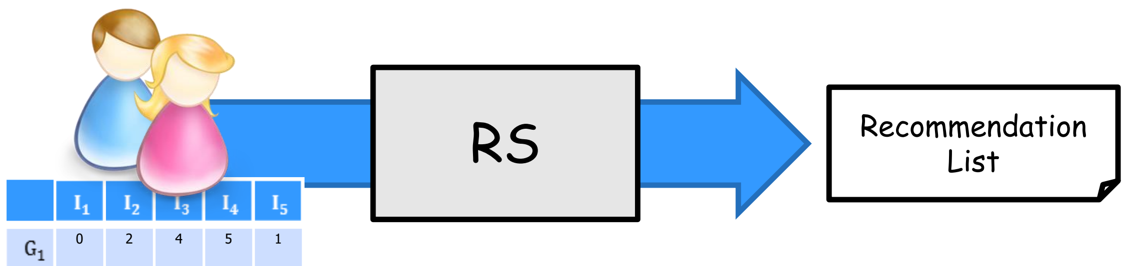
Group Recommendation

- **Group profile-based approach**
  - If group profile is available
  - Treats a group as a single user
  - Most existing recommender systems can be adopted easily, but it is difficult to obtain group profiles

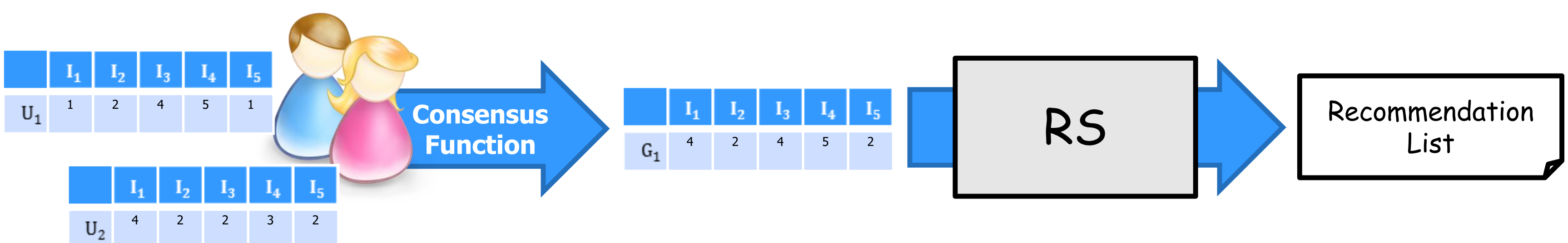
- **Consensus function-based approach**
  - If single user profile is available but group profile is not
  - Imitates decision-making process
  - It is easy to apply, but it needs domain knowledge to select consensus function

Group Recommendation

- **Group profile-based approach**
  - Regular recommender systems are applicable to group profiles



- **Consensus function-based approach**
  - Virtual group is generated through consensus function, regular recommender systems are applied



Group Recommendation

- **Consensus Functions**
  - **Least Misery Strategy**

	I_1	I_2	I_3	I_4	I_5	I_6
U_1	2	2	4	5	1	4
U_2	4	2	2	3	3	2

Min

	I_1	I_2	I_3	I_4	I_5	I_6
G_1	2	2	2	3	1	2

  - **Most Pleasure Strategy**

	I_1	I_2	I_3	I_4	I_5	I_6
U_1	2	2	4	5	1	4
U_2	4	2	2	3	3	2

Max

	I_1	I_2	I_3	I_4	I_5	I_6
G_1	4	2	4	5	3	4

  - **Average Strategy**

	I_1	I_2	I_3	I_4	I_5	I_6
U_1	2	2	4	5	1	4
U_2	4	2	2	3	3	2

Avg

	I_1	I_2	I_3	I_4	I_5	I_6
G_1	3	2	3	4	2	3

Group Recommendation

- **Procedure of Consensus Function-based Approach**
  - Consensus-Recommendation
    - It may reflect more of the group preference, or the consensus between group members

	I_1	I_2	I_3	I_4	I_5	I_6
U_1	1	2	4	5	1	-
U_2	4	2	2	3	2	-

Consensus Function

	I_1	I_2	I_3	I_4	I_5	I_6
G_1	4	2	4	5	2	-

RS

	I_1	I_2	I_3	I_4	I_5	I_6
G_1	4	2	4	5	2	**3**

  - Recommendation-Consensus
    - Recommendation list for the group may reflect more each group member's preference

	I_1	I_2	I_3	I_4	I_5	I_6
U_1	1	2	4	5	1	-
U_2	4	2	2	3	2	-

RS

	I_1	I_2	I_3	I_4	I_5	I_6
U_1	1	2	4	5	1	**2**
U_2	4	2	2	3	2	**4**

Consensus Function

	I_1	I_2	I_3	I_4	I_5	I_6
G_1	4	2	4	5	2	**4**

1. Introduction
2. Collaborative Filtering
3. Content-based Recommendation
4. Context-aware Recommendation
5. Other Approaches

6. Concluding Remarks

Summary

- **Recommendation**
  - Collaborative Filtering
  - Content-based Recommendation
  - Context-aware Recommendation
  - Others…

- **RS are fairly new but already grounded on well-proven technology**

- **However, there are still many open questions and a lot of interesting research to do**

Thank you for your attention

Q&A