Recommender Systems

程 健 研究员
jcheng@nlpr.ia.ac.cn
中国科学院自动化研究所
模式识别国家重点实验室

Index

- What is recommender system?
- Traditional Methods
- Deep Learning base Methods
- RS Systems
- Conclusion

Index

- What is recommender system?
- Traditional Methods
- Deep Learning base Methods
- RS Systems
- Conclusion

Everything is recommendation

Everything is personalized

京东商城

Everything is personalized

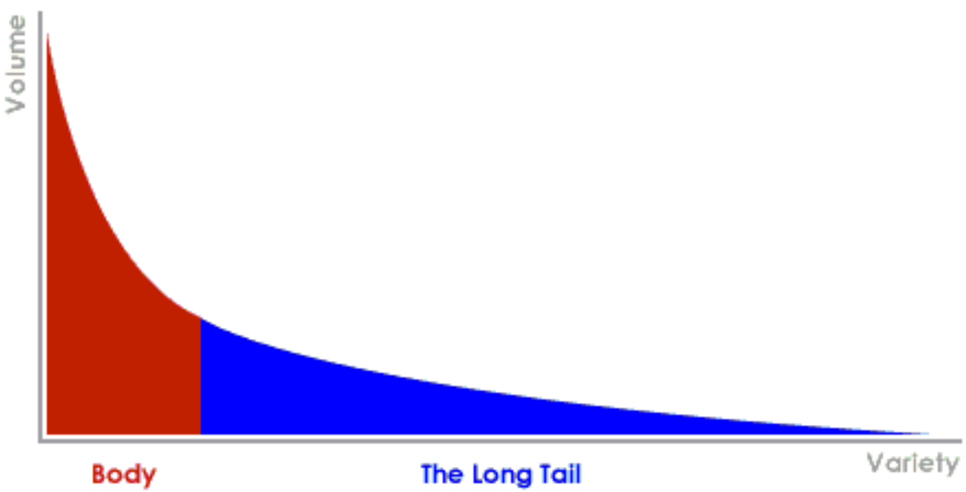
彼之砒霜 吾之蜜糖

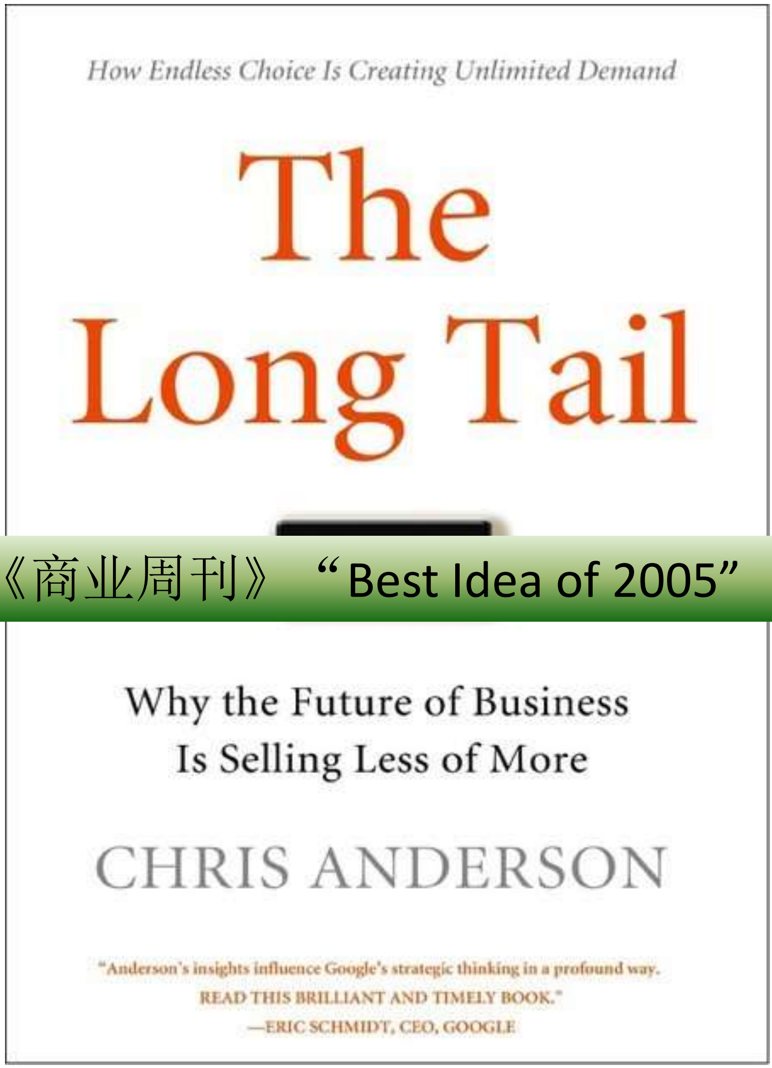
Everything is personalized

- 视频网站
  YouTube 土豆 Hulu 奇艺视频 等
- 电子商务网站
  淘宝，亚马逊等
- 社交网站
  Facebook 人人网 Twitter 微博 等
- ……

The Long Tail

Chris Anderson's
Web 2.0 Business Model:
The Long Tail





《商业周刊》 "Best Idea of 2005"

The Long Tail

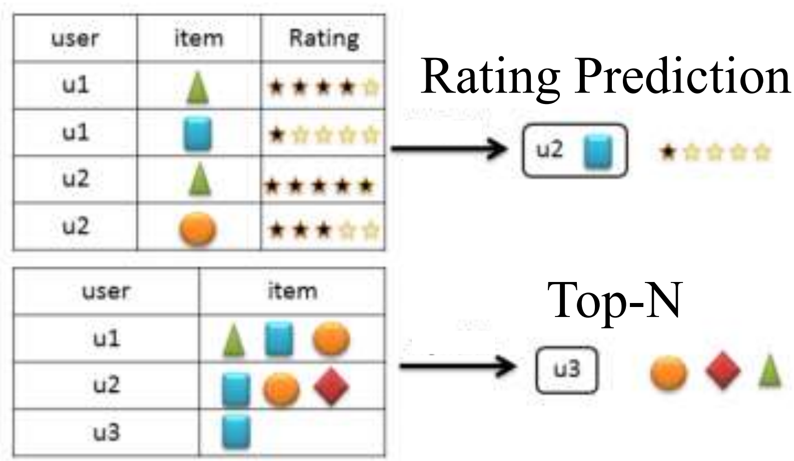
- Amazon: 35% 的销售来自推荐
- Google News: 推荐增加了38%的点击率
- Netflix:2/3的电影出租来自推荐

"We are leaving the age of information and entering the Age of Recommendation"

– The Long Tail (Chris Anderson)

RS Definition

- RS seen as a function
- Given:
  - ➢ User model (e.g. ratings, preferences, demographics, situational context)
  - ➢ Items (with or without description of item characteristics)
- Calculate:
  - ➢ Relevance score used for ranking
- Target:
  - ➢ Rating Prediction & Top-N Recommendation



- But:
  - ➢ Remember that relevance might be context dependent
  - ➢ Characteristics of the list itself might be important (diversity)

Performance Evaluation

- Measures for rating prediction
  - Mean absolute error

$$MAE = \frac{1}{|Test|} \times \sum_{(u,i) \in Test} |\hat{r}_{u,i} - r_{u,i}|$$

  - Root mean square error

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

Performance Evaluation

- Measures for top-N recommendation
  - NDCG(Normalized Discounted Cumulative Gain)

$$\mathrm{DCG_p} = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2(i)}$$

定义不唯一

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(1+i)}$$

$$NDCG_p = \frac{DCG_p}{IDCG_p}$$

Ideal DCG

  - F_1 Score

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

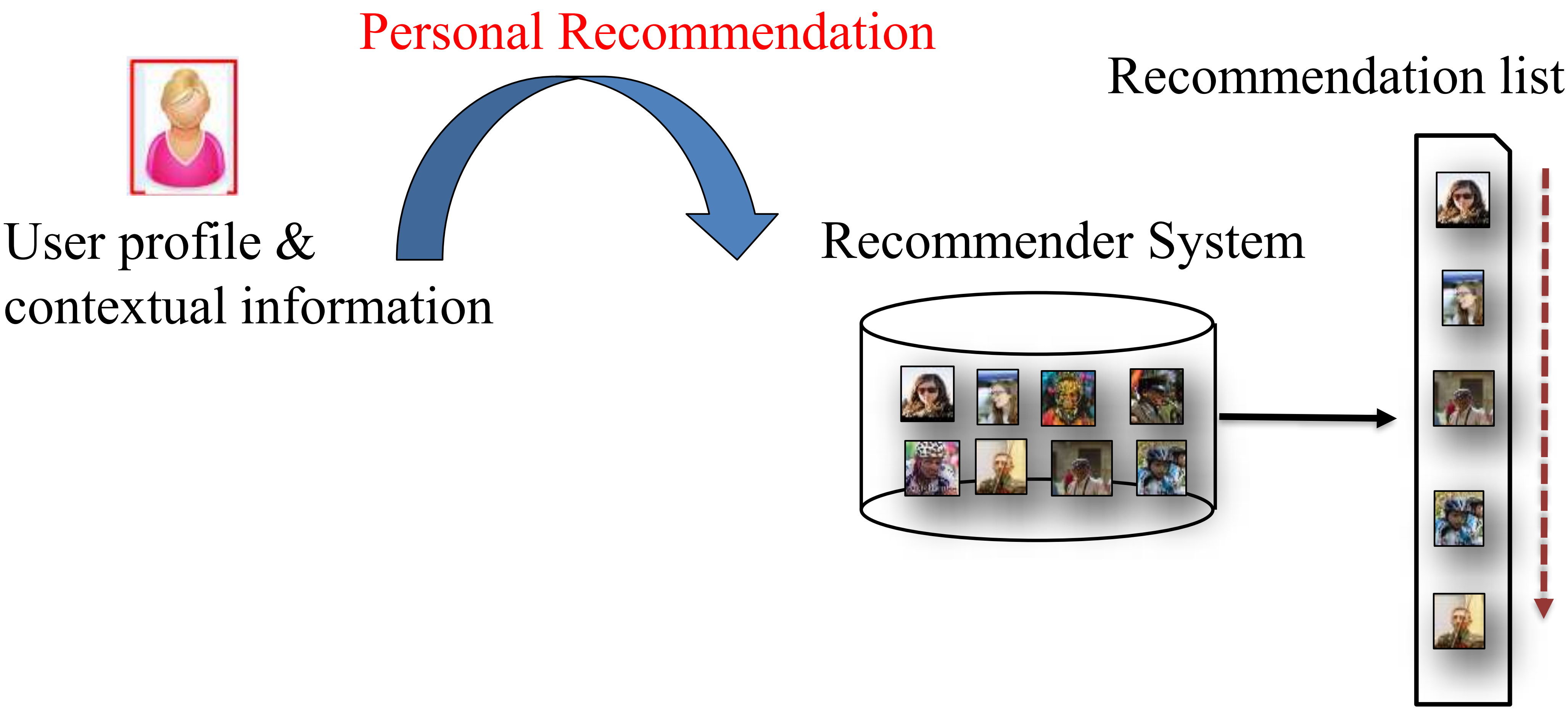
  - Mean Average Precision (MAP)

$$\mathrm{MAP} = \frac{\sum_{q=1}^{Q} \mathrm{AveP(q)}}{Q}$$

Index

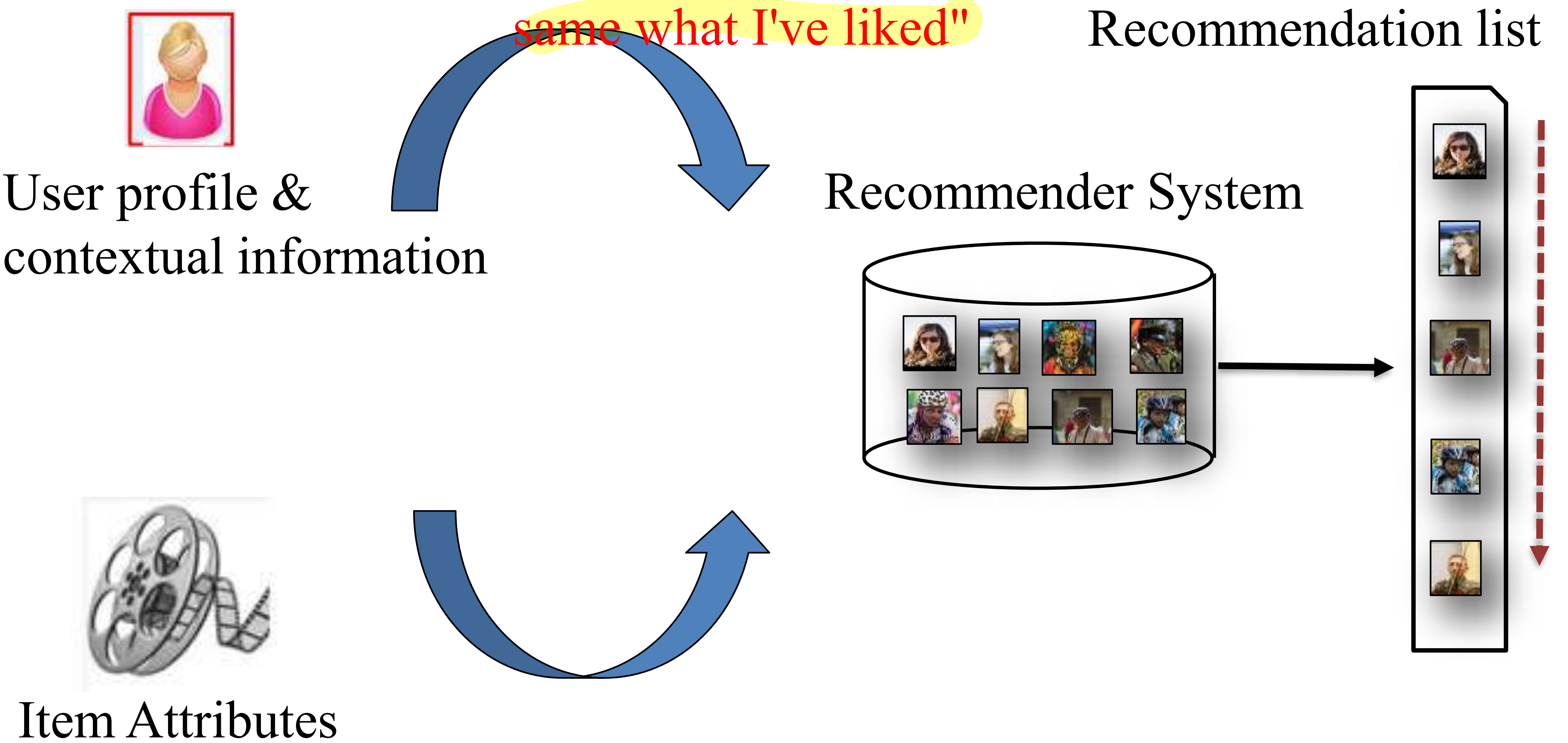
- What is recommender system?
- **Traditional Methods**
- Deep Learning base Methods
- RS Systems
- Conclusion

A glance of Paradigms for RS



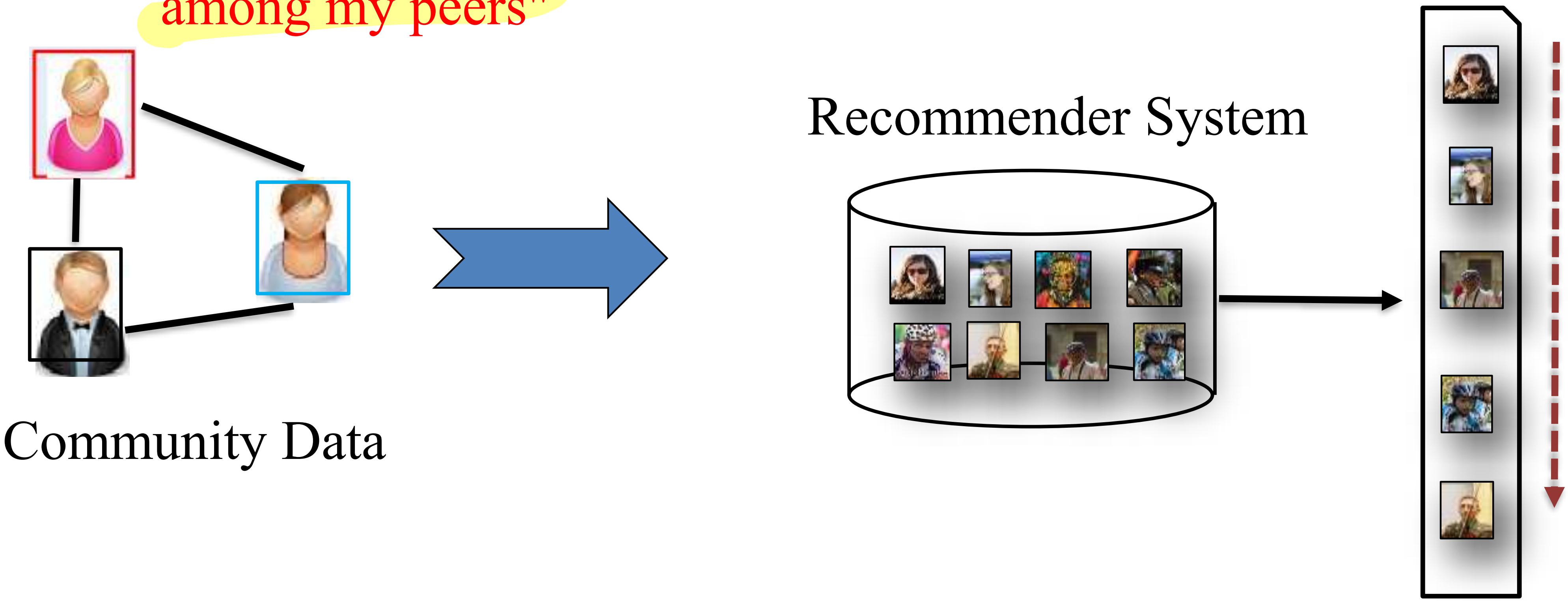
A glance of Paradigms for RS

Content-based: "Show me more of the same what I've liked"

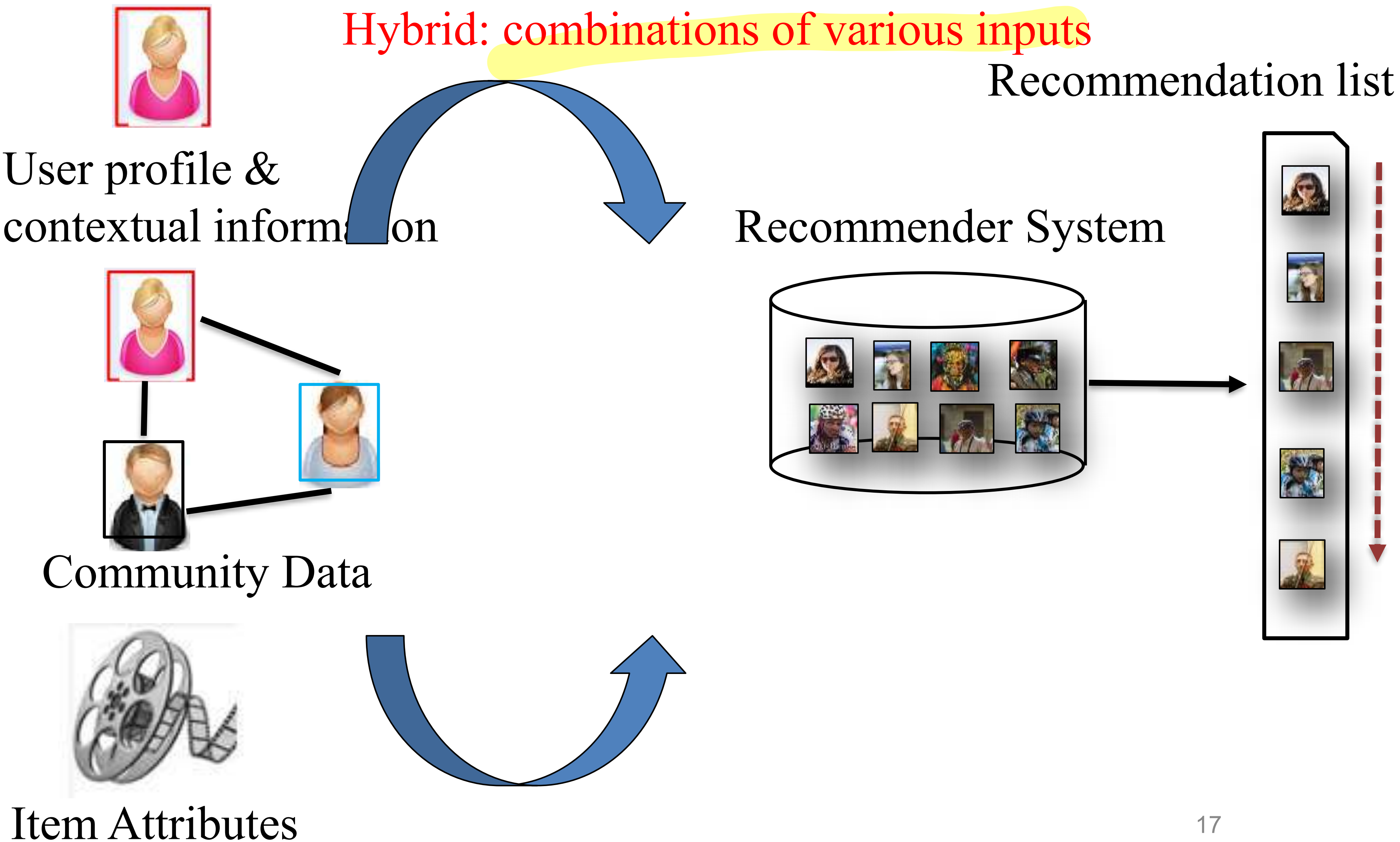


A glance of Paradigms for RS

Collaborative: "Tell me what's popular among my peers"



A glance of Paradigms for RS



Content-Based Recommendation

- Recommendations based on content of items rather than on other users' opinions/interactions
- Goal: recommend items similar to those the user liked
- Common for recommending text-based products (web pages, news messages)
- Items to recommend are "described" by their associated features (e.g. keywords)
- User Model structured in a "similar" way as the content: features/keywords more likely to occur in the preferred documents (lazy approach)
- The user model can be a classifier based on whatever technique (Neural Networks, Naïve Bayes...)

Content-Based Recommendation

- Content representation and item similarities

Express item features as:

- TF-IDF
- N-Gram
- LDA
- Word2Vec

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, Murder, Neo-nazism
...					

Title	Genre	Author	Type	Price	Keywords
...	Fiction, Suspense	Brunonia Barry, Ken Follet, ..	Paperback	25.65	detective, murder, New York

- Compute the similarity of an unseen item with the user profile based on the keyword features

Content-Based Recommendation

- Pros:
  - No need for data on other users: No cold-start or sparsity
  - Able to recommend to users with unique tastes
  - Able to recommend new and unpopular items
  - Can provide explanations by listing content-features

- Cons:
  - Requires content that can be encoded as meaningful features (difficult in some domains/catalogs)
  - Users represented as learnable function of content features
  - Difficult to implement serendipity
  - Easy to overfit (e.g. for a user with few data points)

Collaborative Filtering

- List of m Users and a list of n Items
- Each user has a list of items with associated opinion

opinion
- Explicit (e.g. ratings)
- Implicit (e.g. purchase records)

- Active user for whom the CF prediction task is performed
- Metric for measuring similarity between users
- Method for selecting a subset of neighbors
- Method for predicting a rating for items not currently rated by the active user.

Collaborative Filtering

users

items	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3	?		5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

Collaborative Filtering

- memory-based CF
  - User-based CF
  - Item-based CF

- model-based CF
  - First develop a model of user
  - Type of model:
    1. Probabilistic (e.g. Bayesian Network)
    2. Clustering
    3. Rule-based approaches (e.g. Association Rules)
    4. Classification/Regression
    5. …

User-based CF

The basic steps:

1. Identify set of ratings for the target/active user
2. Identify set of users most similar to the target/active user according to a similarity function (neighborhood formation)
3. Identify the products these similar users liked
4. Generate a prediction
5. Based on this predicted rating recommend a set of top N products

User-based CF

- A collection of user $u_i, i = 1 \ldots m$ and a collection of products $p_j, j = 1, \ldots, n$
- An m $\times$ n matrix of ratings , with $r_{ij} = ?$ if user i did not rate product j
- Prediction for user i and product j is computed as $r_{ij}^{\ *} = K \sum_{r_{kj \neq ?}} u_{ik} r_{kj}$
- Similarity can be computed by Pearson correlation

$$u_{ik} = \frac{\sum_j (r_{ij} - r_i)(r_{kj} - r_k)}{\sqrt{\sum_j (r_{ij} - r_i)^2 \sum_j (r_{kj} - r_k)^2}}$$

User-based CF Example

						sim(u,v)
2		2	4	5		NA
5		4			1	
		5		2		
	1		5		4	
		4			2	
4	5		1			NA

User-based CF Example

						sim(u,v)
2		2	4	5		NA
5		4			1	0.87
		5		2		
	1		5		4	
		4			2	
4	5		1			NA

User-based CF Example

						sim(u,v)
2		2	4	5		NA
5		4			1	0.87
		5		2		1
	1		5		4	
		4			2	
4	5		1			NA

User-based CF Example

	SHERLOCK	HOUSE of CARDS	AVENGERS	ARRESTED DEVELOPMENT	Breaking Bad	WALKING DEAD	sim(u,v)
	2		2	4	5		NA
	5		4			1	0.87
			5		2		1
		1		5		4	-1
			4			2	
	4	5		1			NA

Item-based CF Example

The basic steps:

1. Look into the items the target user has rated
2. Compute how similar they are to the target item
3. Select k most similar items
4. Compute Prediction by taking weighted average on the target user's ratings on the most similar items

Item Similarity Computation

- Similarity: find users who have rated items and apply a similarity function to their ratings

- Cosine-based Similarity (difference in rating scale between users is not taken into account)

$$sim(\mathrm{a},\mathrm{b}) = \frac{a \bullet b}{|\mathrm{a}| \times |b|}$$

- Adjusted Cosine Similarity (takes care of difference in rating scale)

$$S(\mathrm{i},\mathrm{j}) = \frac{\sum_u (\mathrm{r}_{ui} - \mathrm{r}_u)(\mathrm{r}_{uj} - \mathrm{r}_u)}{\sqrt{\sum_u (\mathrm{r}_{ui} - \mathrm{r}_u)^2 \sum_u (\mathrm{r}_{uj} - \mathrm{r}_u)^2}}$$

Item Similarity Computation

- Alternative similarity metric

Correlation based	Cosine, Pearson Correlation, Adjusted Cosine,OLS coefficient
Distance based	Euclidean distance, Manhattan distance,Minkowski distance
Hash based	Mini Hash, Sim Hash
Topic based	PLSA, LDA
Graph based	Shortest Path, Random Walk, Item Rank

Model-based CF

Motivated by Netflix Prize (launched in Oct. 2006)

- Task:
  High quality recommendations for cinematch (RMSE=0.9525)
- Dataset:
  users: 480,000
  movies: 17,770
  rates ratio <1%

Improve by 10% = $1million!

Model-based CF

Motivated by Netflix Prize (launched in Oct. 2006)

- Measure:

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Model-based CF

Leaderboard

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace_	0.8612	9.59	2009-07-24 17:18:43

Model-based CF

2009 Netflix Prize Results

- Top 2 single algorithms:

  SVD/MF - Prize RMSE: 0.8914

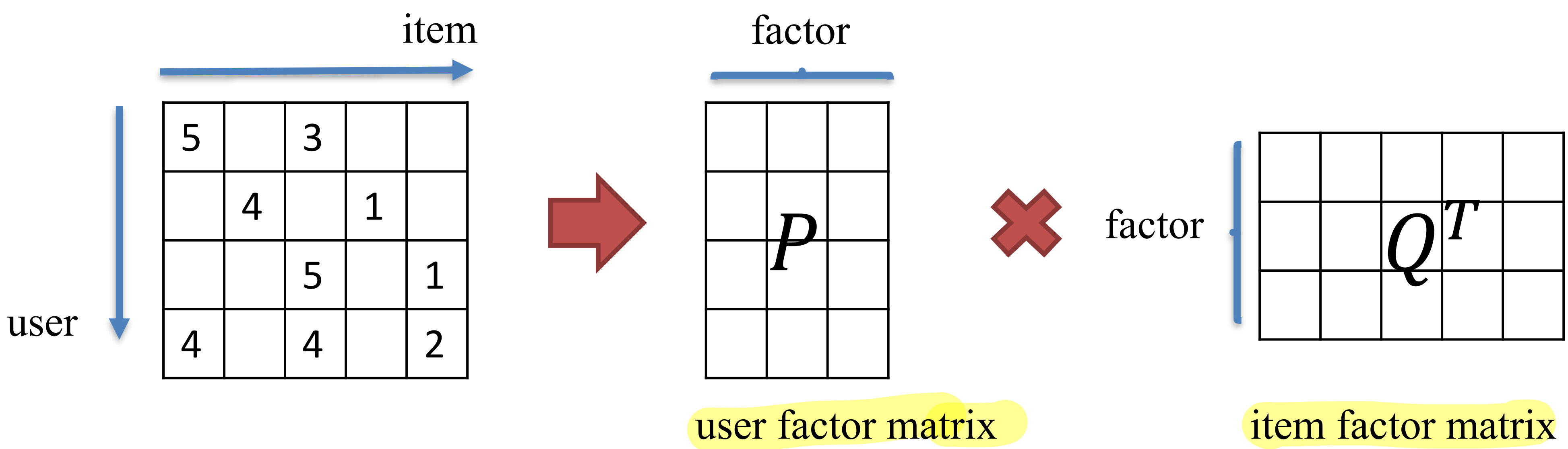
  RBM - Prize RMSE: 0.8990

- Linear blend Prize RMSE: 0.88

- Currently in use as part of Netflix' rating prediction component

Matrix Factorization

- Basic idea



- factor size << dim of user/item

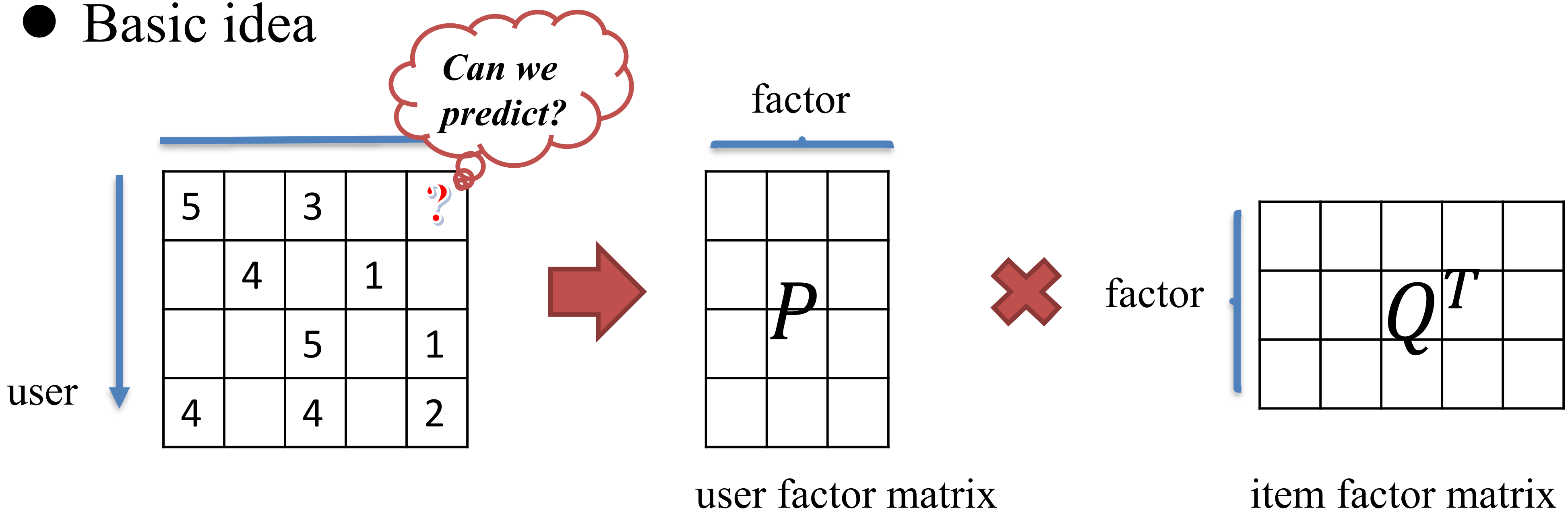
$$p_u \quad \boxed{f_1 \mid f_2 \mid f_3 \mid \dots \mid f_k}$$

$$q_v \quad \boxed{f'_1 \mid f'_2 \mid f'_3 \mid \dots \mid f'_k}$$

- User factor vectors $p_u \in R^f$ and item factor vector $q_v \in R^f$

Matrix Factorization

- Basic idea



- factor size << dim of user/item

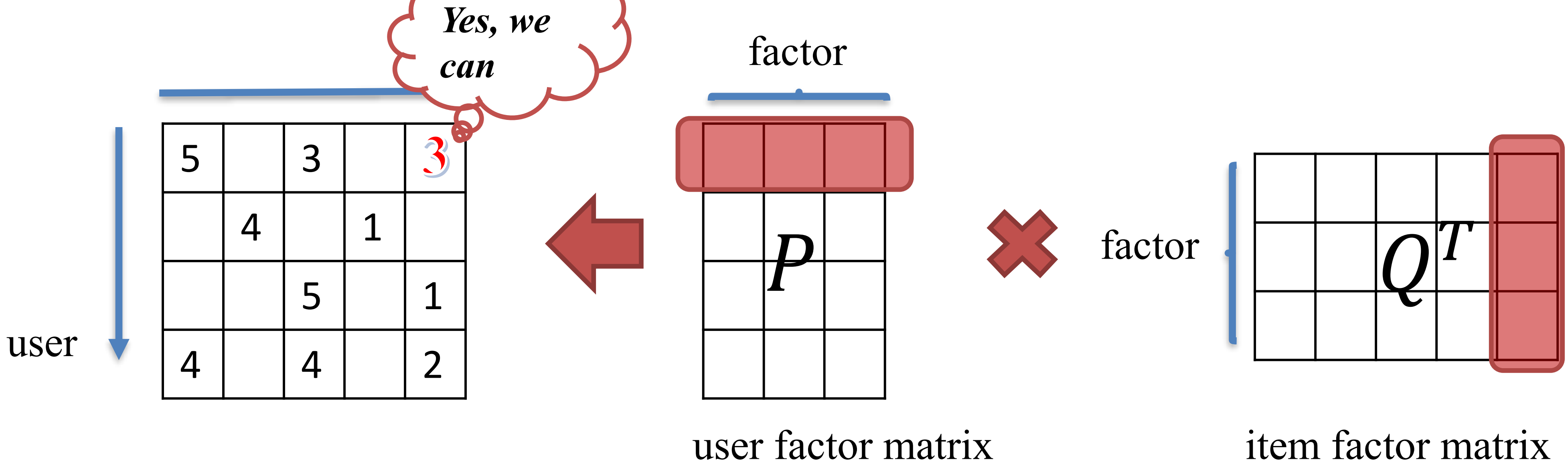
$$p_u \; \boxed{f_1 \,|\, f_2 \,|\, f_3 \,|\, \ldots \,|\, f_k}$$

$$q_v \; \boxed{f_1' \,|\, f_2' \,|\, f_3' \,|\, \ldots \,|\, f_k'}$$

- User factor vectors $p_u \in R^f$ and item factor vector $q_v \in R^f$

Matrix Factorization

- Basic idea



- factor size << dim of user/item

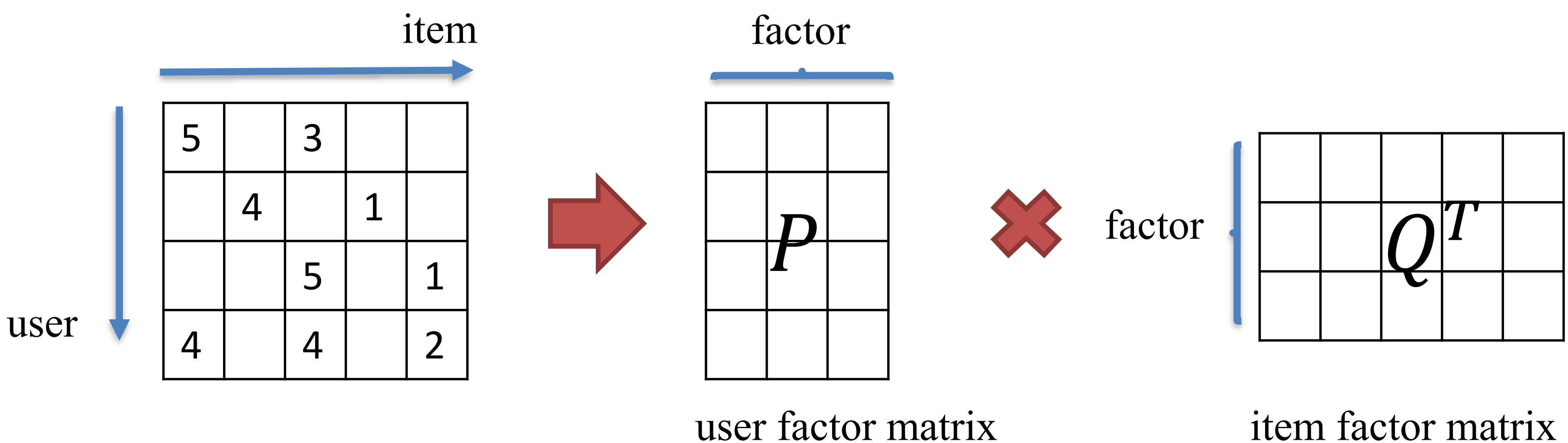
$$p_u \;\; \boxed{f_1 \mid f_2 \mid f_3 \mid \dots \mid f_k}$$

$$q_v \;\; \boxed{f'_1 \mid f'_2 \mid f'_3 \mid \dots \mid f'_k}$$

- User factor vectors $p_u \in R^f$ and item factor vector $q_v \in R^f$

Non-negative Matrix Factorization

- Both entries in factorized P and Q should be >= 0



- Explanation: real world data, i.e. images, has often been represented as non-negative values, while negative ones doesn't have any meanings.

p_u $\boxed{f_1 \mid f_2 \mid f_3 \mid \dots \mid f_k}$ >=0

q_v $\boxed{f'_1 \mid f'_2 \mid f'_3 \mid \dots \mid f'_k}$ >=0

Non-negative Matrix Factorization

- 'Orthogonal NMF' == 'Kernel K-Means Clustering'

Orthogonal NMF

$$\min_{F,G} \| X - FG^T \|_F^2, \text{s.t.}\ G^T G = I, G \geq 0$$

is equivalent to K-means clustering. Where each row of $G \in R^{n \times r}$ can be viewed as a probability distribution of the factors(clusters) .

Proof

1. Kernel K-means clustering tries to minimize $J = \sum_{k=1}^{K} \sum_{i \in C_k} \| \mathrm{x}_i - m_k \|^2$

By utilizing an indicator matrix $G = (\mathrm{g}_1, \ldots, \mathrm{g}_K), \mathrm{g}_k^T g_l = \delta_{kl}$, where

$g_k = (0, .., 0, 1, ..., 1, 0, ..., 0)^T / n_k^{1/2}$, the above formulation can be transformed to

$$\max J(G) = \max Tr(G^T XXG), \text{s.t.}\ G^T G = I, G \geq 0$$

Non-negative Matrix Factorization

- ‘Orthogonal NMF’ == ‘Kernel K-Means Clustering’

2. We write the NMF formulation as

$$J = \| X - FG^T \|_F^2 = Tr(X^T X - 2F^T XG + F^T F)$$

the zero gradient condition $\partial J / \partial F = -2XG + 2F = 0$, given $F = XG$
then $J = Tr(X^T X - G^T XXG)$, the optimization can also be transformed to

$$\min_G Tr(-G^T XXG), \mathrm{s.t.} G^T G = I, G \geq 0$$

Further transform to

$$\max_G Tr(G^T XXG), \mathrm{s.t.} G^T G = I, G \geq 0$$

Which has the same form as Kernel K-means clustering

SVD for Rating Prediction

However,
- ➢ Some items are significantly higher rated…
- ➢ Some users rate substantially lower…
- ➢ All Ratings are high…

Thus,
- ➢ Add item offset…
- ➢ Add user offset…
- ➢ Add global offset…

- Baseline (bias) $b_{uv} = \mu + b_u + b_v$ (user & item deviation from average)
- Predict rating as $\hat{r}_{uv} = b_{uv} + p_u^T q_v$

SVD for Rating Prediction

- In order to prevent over-fitted problem, we add some regularized terms, such as:

$$SSE = \frac{1}{2}(r_{uv} - \hat{r}_{uv})^2 + \lambda\left(\sum_u |p_u|^2 + \sum_v |q_v|^2\right)$$

- SVD++ asymmetric variation with implicit feedback

$$\hat{r}_{uv} = b_{uv} + q_v^T\left(|R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_{uj}) x_j + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j\right)$$

Where

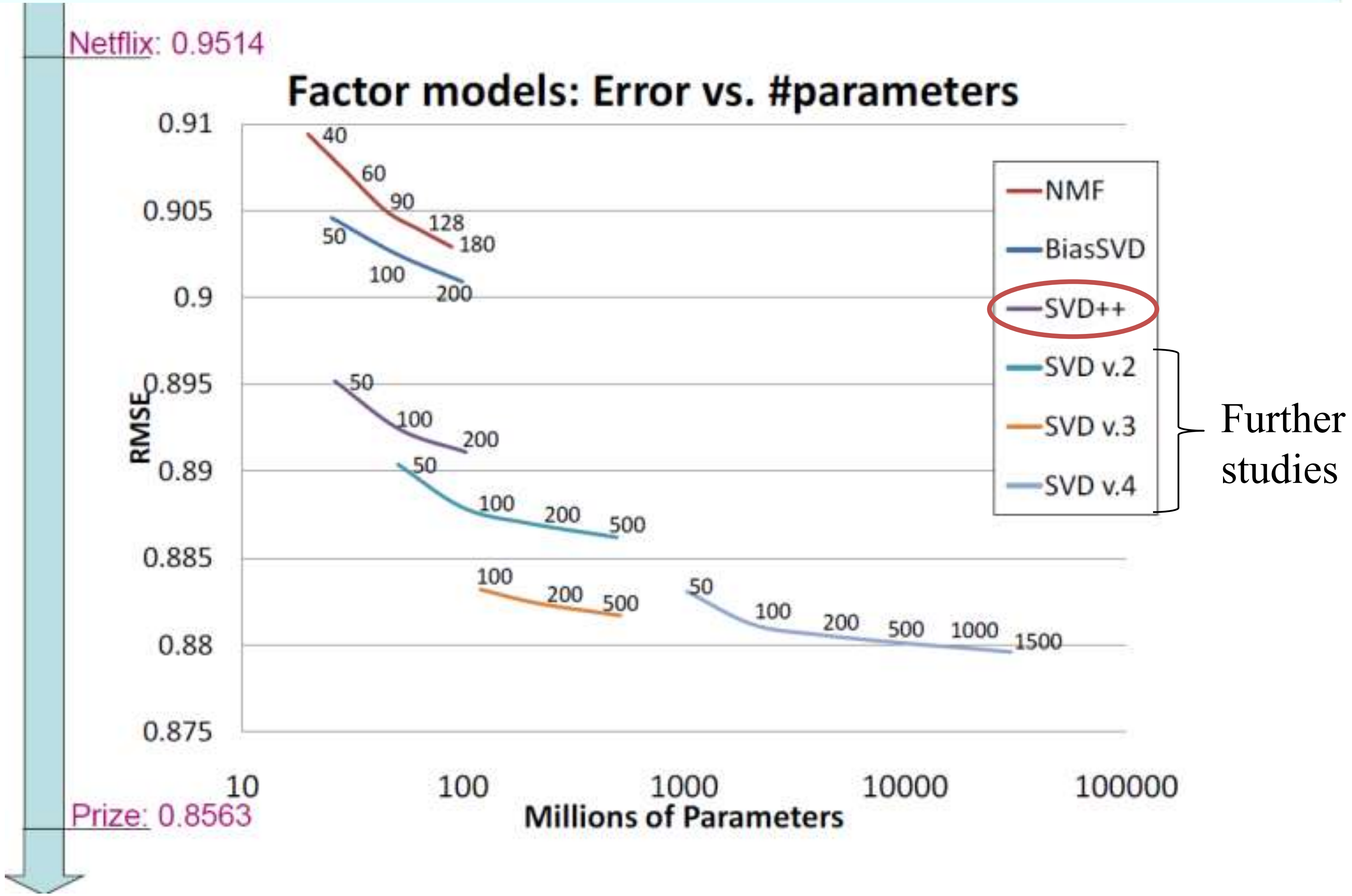
$q_v, x_v, y_v \in R^f$ are three item factor vectors

$R(u)$ items rated by user u

$N(u)$ items for which the user has given implicit preference

Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer, 2009.

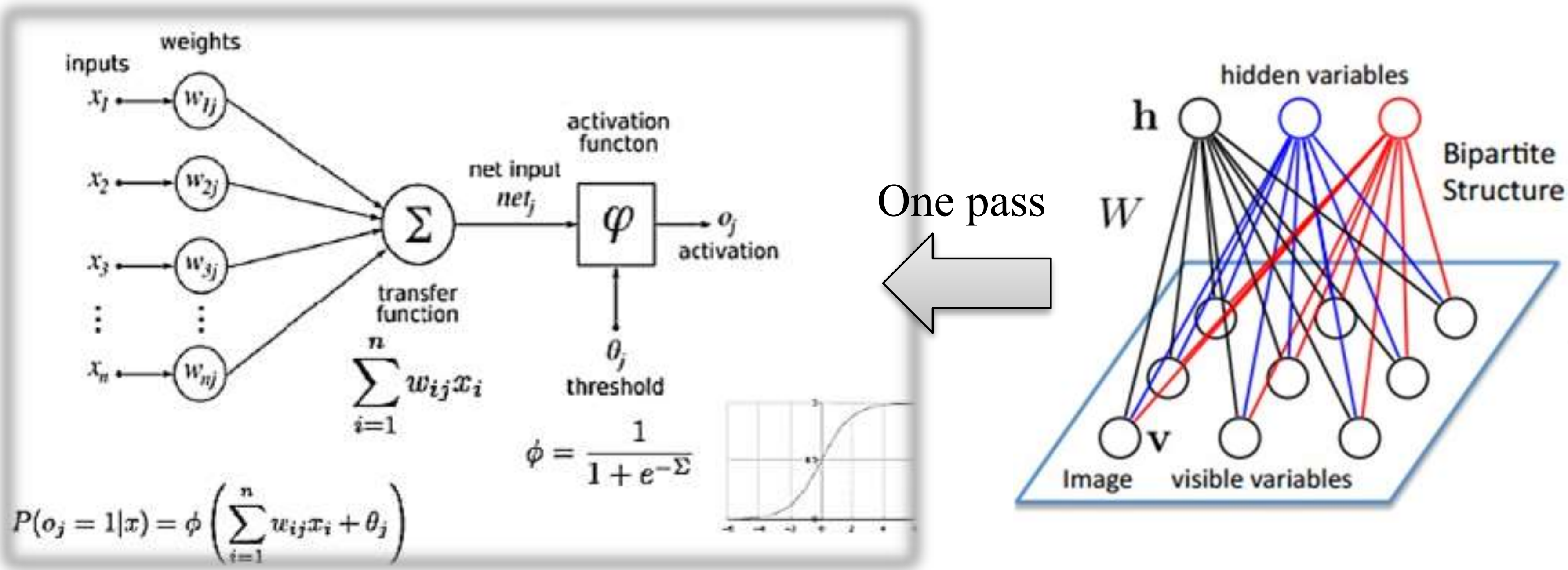
SVD for Rating Prediction



Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer, 2009.

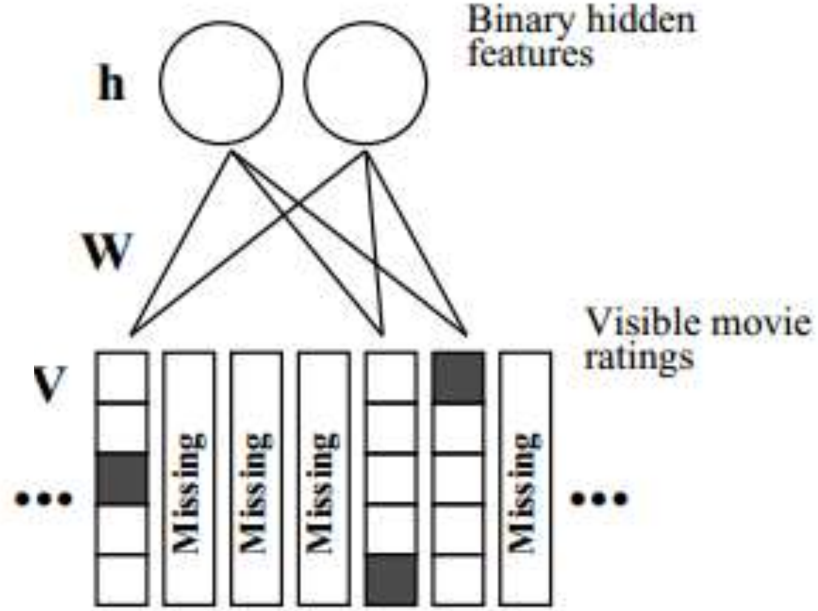
Restricted Boltzmann Machines

- Each unit is a state that can be active or not active
- Each input to a unit is associated to a weight
- The transfer function $\sum$ calculates a score for every unit based on the weighted sum of inputs
- Score is passed to the activation function φ that calculates the probability of the unit to be active



RBM for CF

- Each visible unit = an item
- Num of hidden units is a parameter
- In training phase, for each user:
  - If user rated item, v_i is activated
  - Activation states of v_i = inputs to h_j
  - Based on activation, h_j is computed
  - Activation state of h_j becomes input to v_i
  - Activation state of v_i is recalculated
  - Difference between current and past activation state for v_i used to update weights w_{ij} and thresholds
- In prediction phase:
  - For the items of the user the v_i are activated
  - Based on this the state of the h_j is computed
  - The activation of h_j is used as input to recompute the state of v_i
  - Activation probabilities are used to recommend items



Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering. ICML, 2007

Probabilistic Matrix Factorization

- From the view of probability to predict ratings, we assume factorized vectors of users and items are in line with the Gaussian distribution, user's preference for items is a combination of the probability of a series of problems, such as

$$p(R|U,V,\sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}(R_{ij}|U_i^T V_j,\sigma^2)\right]^{I_{ij}}$$

where

$$p(U|\sigma_U^2) = \prod_{i=1}^{N}\mathcal{N}(U_i|0,\sigma_U^2\mathbf{I}), \qquad p(V|\sigma_V^2) = \prod_{j=1}^{M}\mathcal{N}(V_j|0,\sigma_V^2\mathbf{I}).$$

$$\ln p(U,V|R,\sigma^2,\sigma_V^2,\sigma_U^2) = -\frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij}-U_i^T V_j)^2 - \frac{1}{2\sigma_U^2}\sum_{i=1}^{N} U_i^T U_i - \frac{1}{2\sigma_V^2}\sum_{j=1}^{M} V_j^T V_j$$
$$-\frac{1}{2}\left(\left(\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}\right)\ln\sigma^2 + ND\ln\sigma_U^2 + MD\ln\sigma_V^2\right) + C,$$

Salakhutdinov R, Mnih A. Probabilistic matrix factorization. NIPS, 2008.

Probabilistic Matrix Factorization

- By adding regularized terms, the formulation can be shown as:

$$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}\left(R_{ij} - U_i^T V_j\right)^2 + \frac{\lambda_U}{2}\sum_{i=1}^{N} \| U_i \|_{Fro}^2 + \frac{\lambda_V}{2}\sum_{j=1}^{M} \| V_j \|_{Fro}^2$$

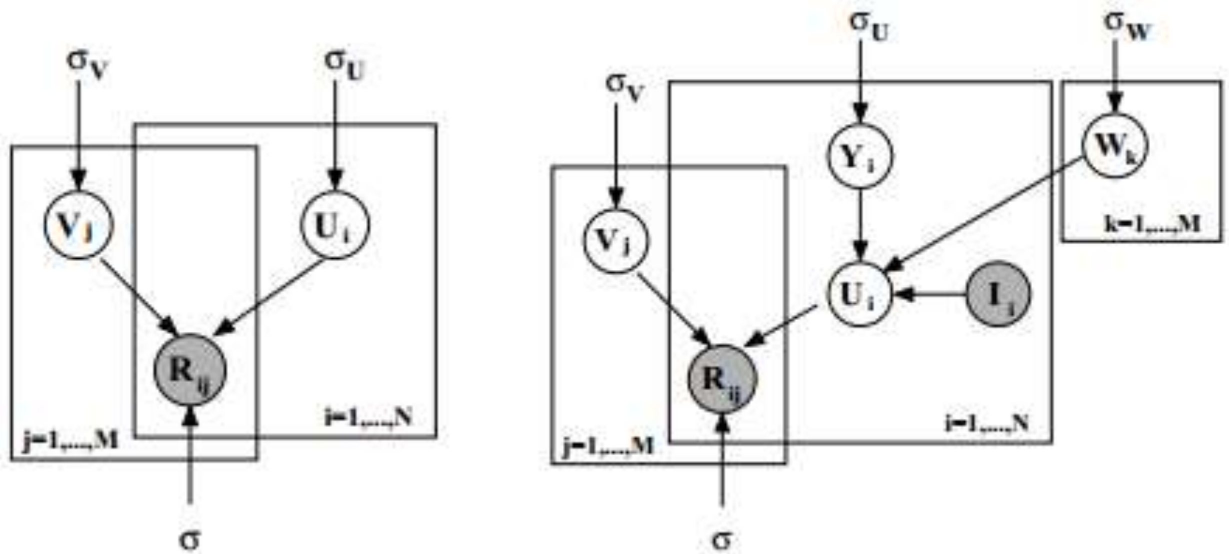


Figure 1: The left panel shows the graphical model for Probabilistic Matrix Factorization (PMF). The right panel shows the graphical model for constrained PMF.

Salakhutdinov R, Mnih A. Probabilistic matrix factorization. NIPS, 2008.

Probabilistic Matrix Factorization

- In order to normalize the scores (i.e. 1-5), the paper uses the following approach

$$g(x) = 1/(1 + \exp(-x))$$

- Thus, the final formulation can be written as:

$$p(R|U,V,\sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}(R_{ij}|g(U_i^T V_j),\sigma^2)\right]^{I_{ij}}$$

- The implementation is adopted Gibbs Sampling Strategy

Salakhutdinov R, Mnih A. Probabilistic matrix factorization. NIPS, 2008.

Probabilistic Matrix Factorization

- Once a PMF model has been fitted, users with very few ratings will have feature vectors that are close to the prior mean so the predicted ratings for those users will be close to the movie average ratings.
- Let $W \in R^{D \times M}$ be a latent similarity constraint matrix. We define the feature vector for user i as

$$U_i = Y_i + \frac{\sum_{k=1}^{M} I_{ik} W_k}{\sum_{k=1}^{M} I_{ik}}.$$

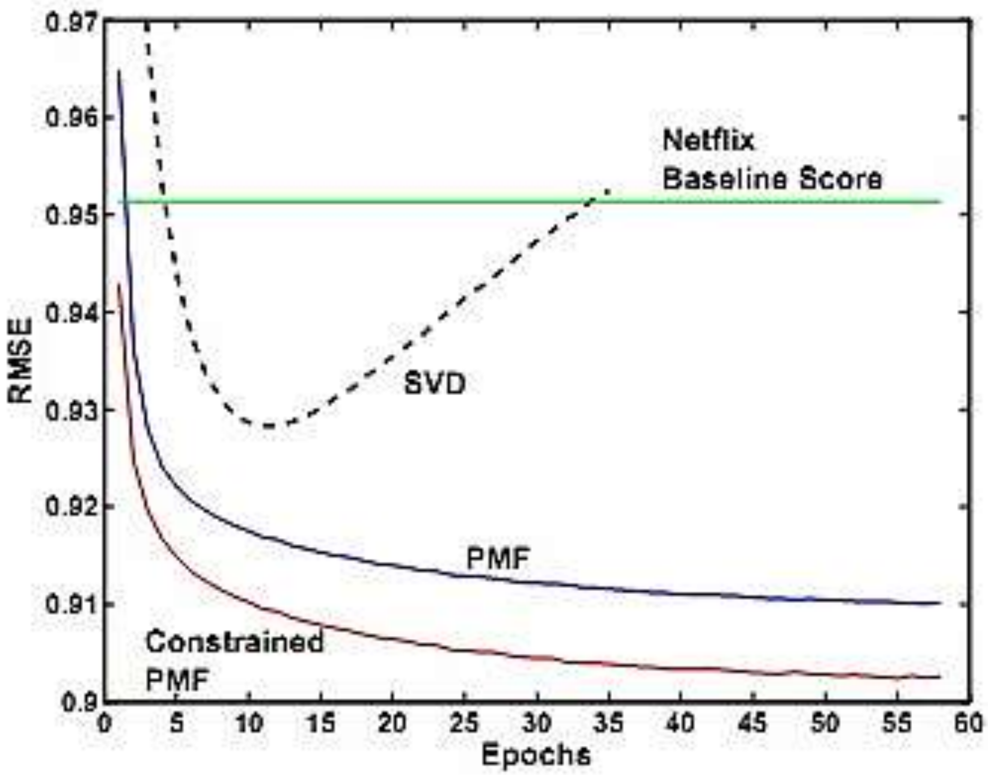
- The corresponding Constrained PMF formulation can be shown as:

$$p(R|Y,V,W,\sigma^2) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left[\mathcal{N}(R_{ij} | g([Y_i + \frac{\sum_{k=1}^{M} I_{ik} W_k}{\sum_{k=1}^{M} I_{ik}}]^T V_j), \sigma^2) \right]^{I_{ij}}$$

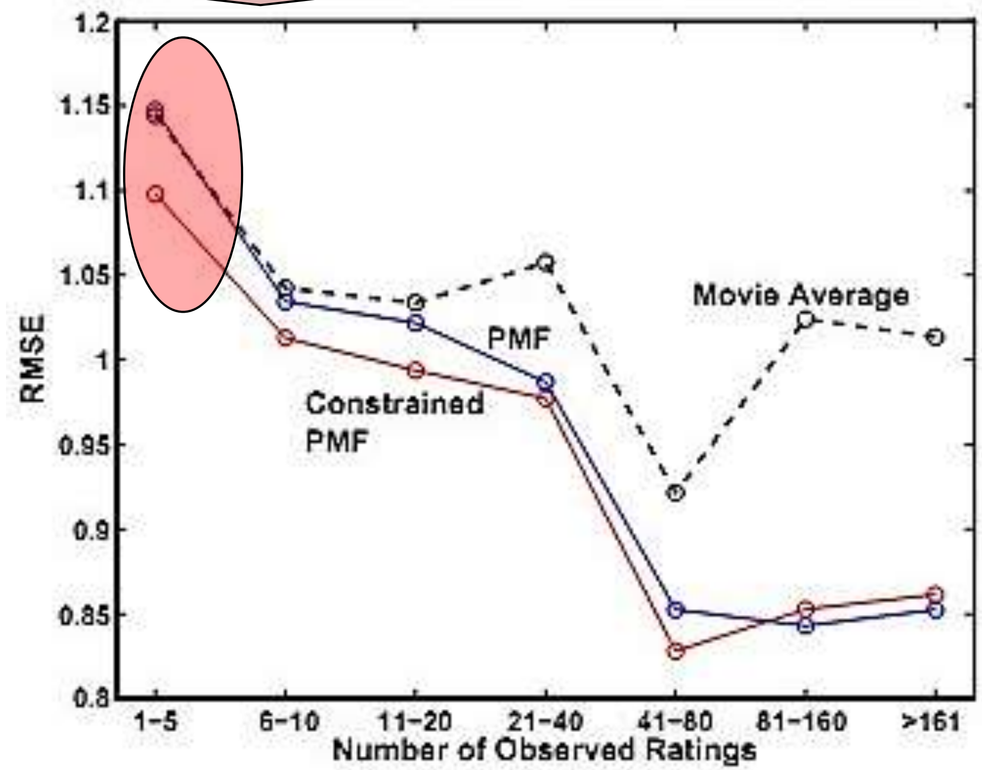
Salakhutdinov R, Mnih A. Probabilistic matrix factorization. NIPS, 2008.

Probabilistic Matrix Factorization

- Experiments



Very helpful for infrequent users



Salakhutdinov R, Mnih A. Probabilistic matrix factorization. NIPS, 2008.

Self-Representation Model

- Beyond matrix factorization, there is another form of modeling users'preference:

item →

user ↓

A

5		3		
	4		1	
		5		1
4		4		2

$\Rightarrow$

A

5		3		
	4		1	
		5		1
4		4		2

original matrix

$\times$

W

0	.2	0	1	.3
.1	0	.7	0	.4
.3	.8	0	0	.1
0	0	0	0	.2
.6	0	.3	0	0

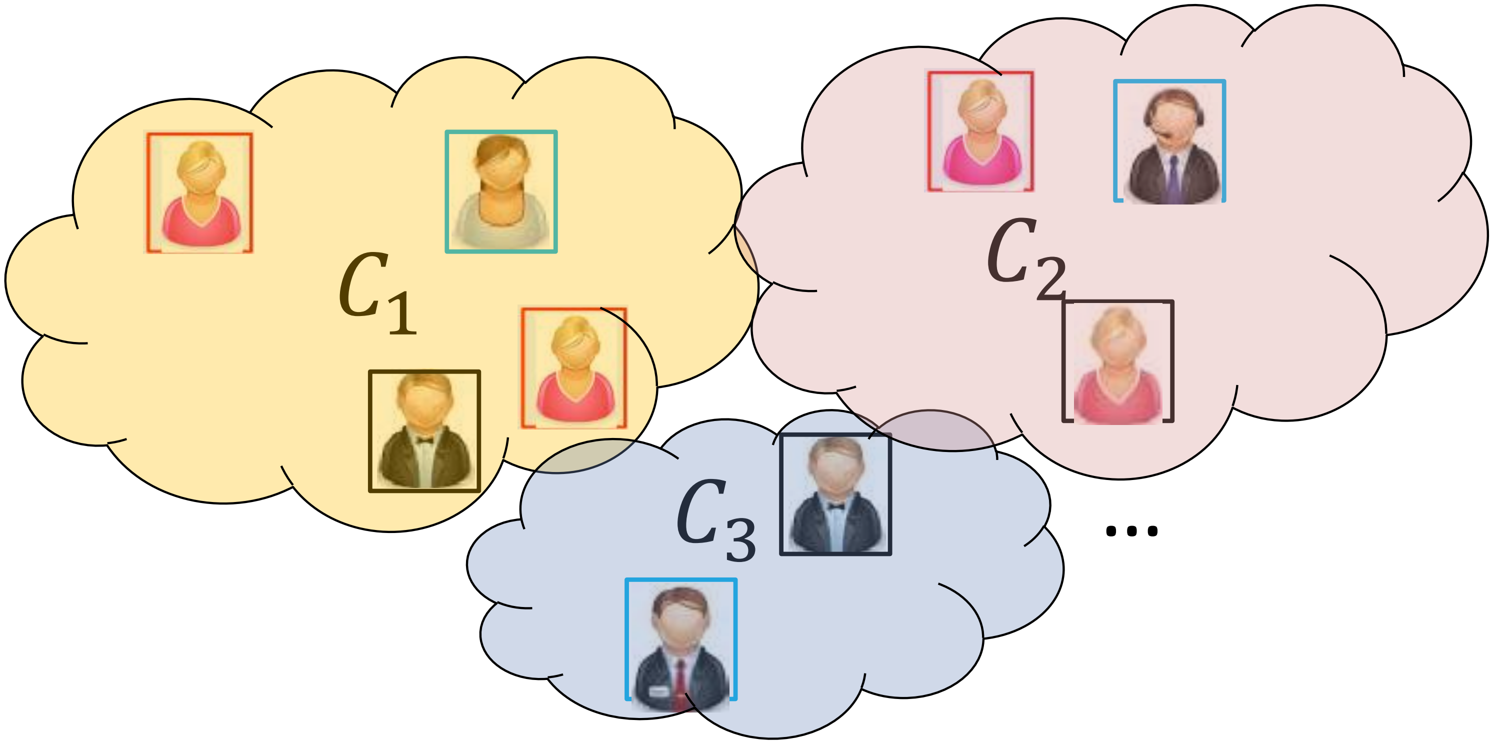
weight matrix

$$
\begin{aligned}
\underset{W}{\text{minimize}} \quad & \frac{1}{2}\|A - AW\|_F^2 + \frac{\beta}{2}\|W\|_F^2 + \lambda\|W\|_1 \\
\text{subject to} \quad & W \geq 0 \\
& \operatorname{diag}(W) = 0.
\end{aligned}
$$

Ning X, Karypis G. Slim: Sparse linear methods for top-n recommender systems. ICDM 2011

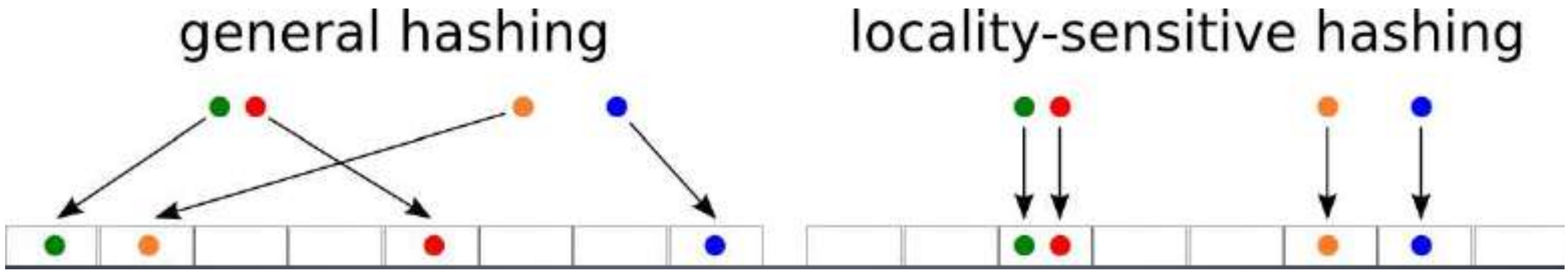
Clustering Based CF

- Goal: cluster users and compute per-cluster "typical" preferences
- Users receive recommendations computed at the cluster level



LSH for clustering

- Method for grouping similar items in highly dimensional spaces
- Find a hashing function s.t. similar items are grouped in the same buckets
- Main application is Nearest-neighbors
  - Hashing function is found iteratively by concatenating random hashing functions
  - Addresses one of NN main concerns: performance



Classifiers for CF

- Classifiers are general computational models trained using positive and negative examples
- They may take in inputs:
  - Vector of item features (action / adventure)
  - Preferences of customers (like action / adventure)…
  - Relations among item
- E.g. Logistic Regression, Bayesian Networks, Support Vector Machines, Decision Trees, etc...
- Pros:
  - Versatile
  - Can be combined with other methods to improve accuracy
- Cons:
  - Need a relevant training set
  - May overfit

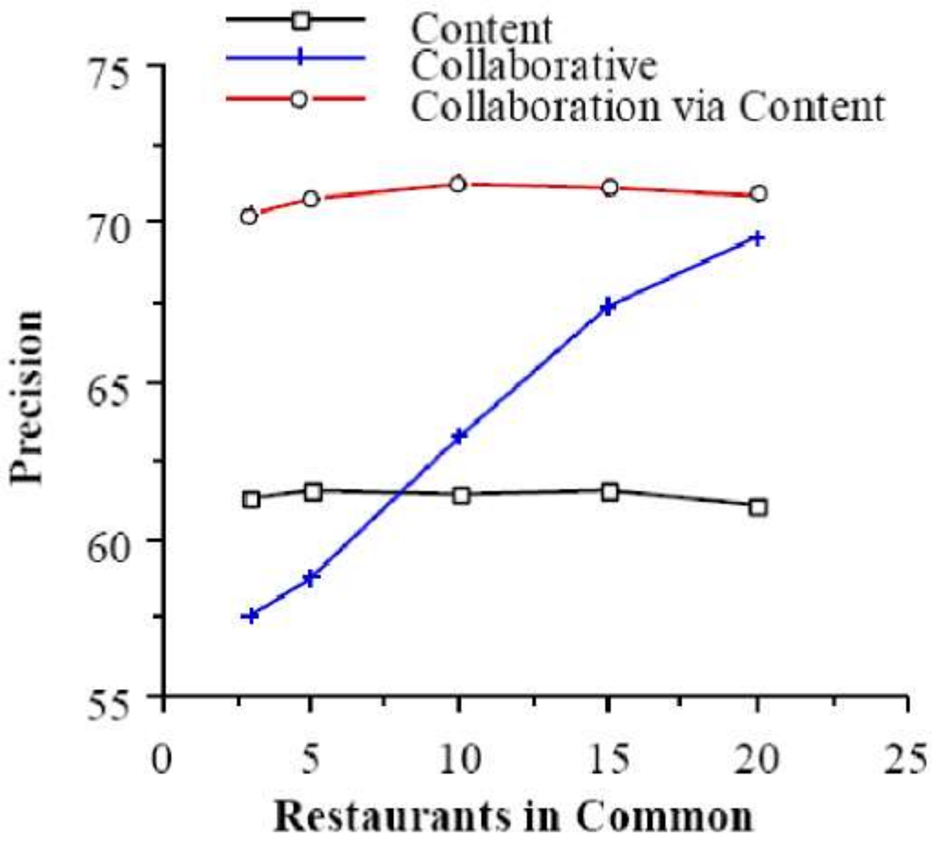
Limitations of CF

- Cold Start: There needs to be enough other users already in the system to find a match. New items need to get enough ratings.

- Popularity Bias: Hard to recommend items to someone with unique tastes. Tends to recommend popular items

Hybrid Approaches

- Content–based recommendation with Bayesian classifier
- Collaborative is standard using Pearson correlation:
- Collaboration via content uses the content-based user profiles

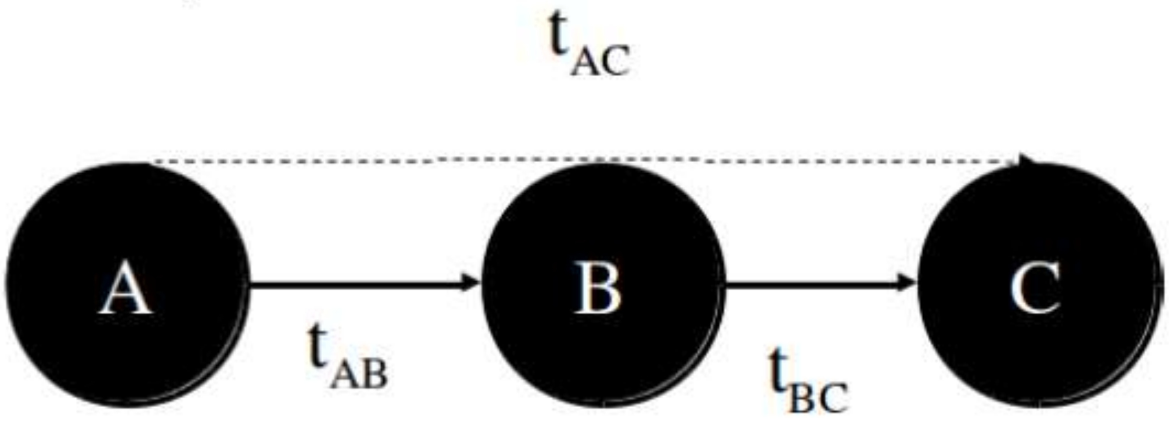


Hybridization Methods

Hybridization Method	Description
Weighted	Outputs from several techniques (in the form of scores or votes) are combined with different degrees of importance to offer final recommendations
Switching	Depending on situation, the system changes from one technique to another
Mixed	Recommendations from several techniques are presented at the same time
Feature combination	Features from different recommendation sources are combined as input to a single technique
Cascade	The output from one technique is used as input of another that refines the result
Feature augmentation	The output from one technique is used as input features to another
Meta-level	The model learned by one recommender is used as input to another

Social Recommendations

- A social recommender system recommends items that are "popular" in the social proximity of the user
- Social proximity = trust (can also be topic-specific)
- Given two individuals - the source (node A) and sink (node C) - derive how much the source should trust the sink.
- Algorithm: Advogato, Appleseed, MoleTrust, TidalTrust

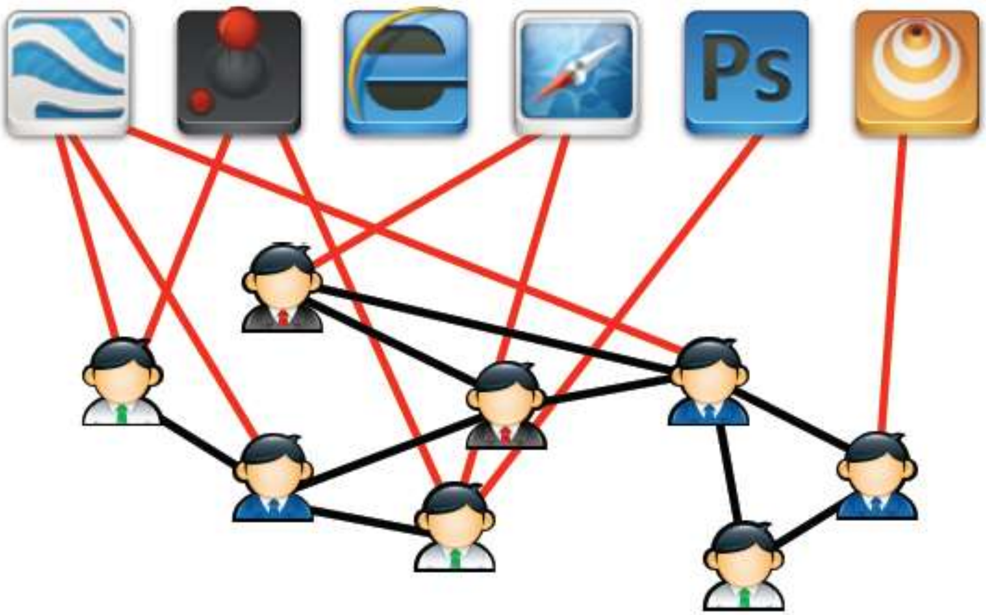


Social Recommendations

social network = friendship + interests

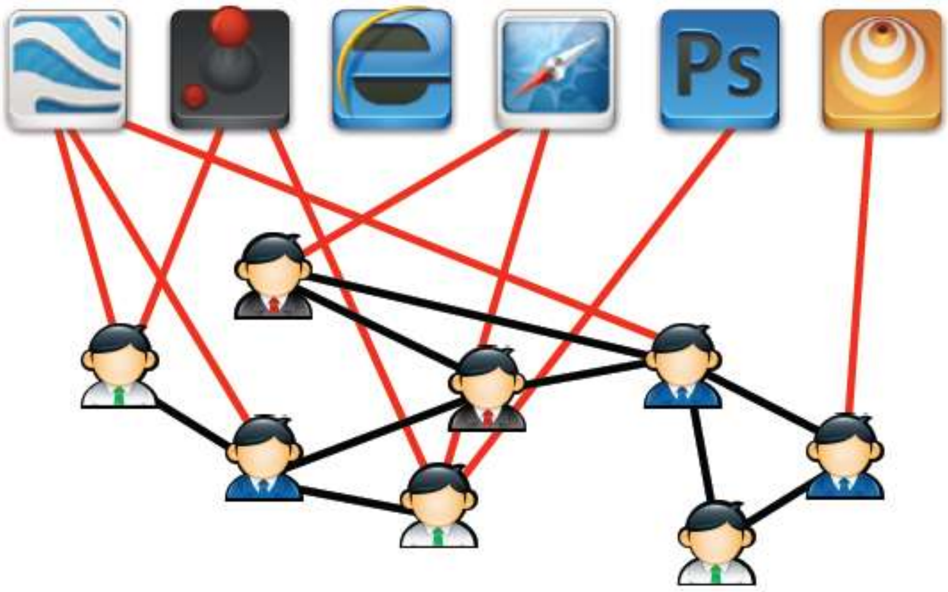
recommend users based on friendship & interests

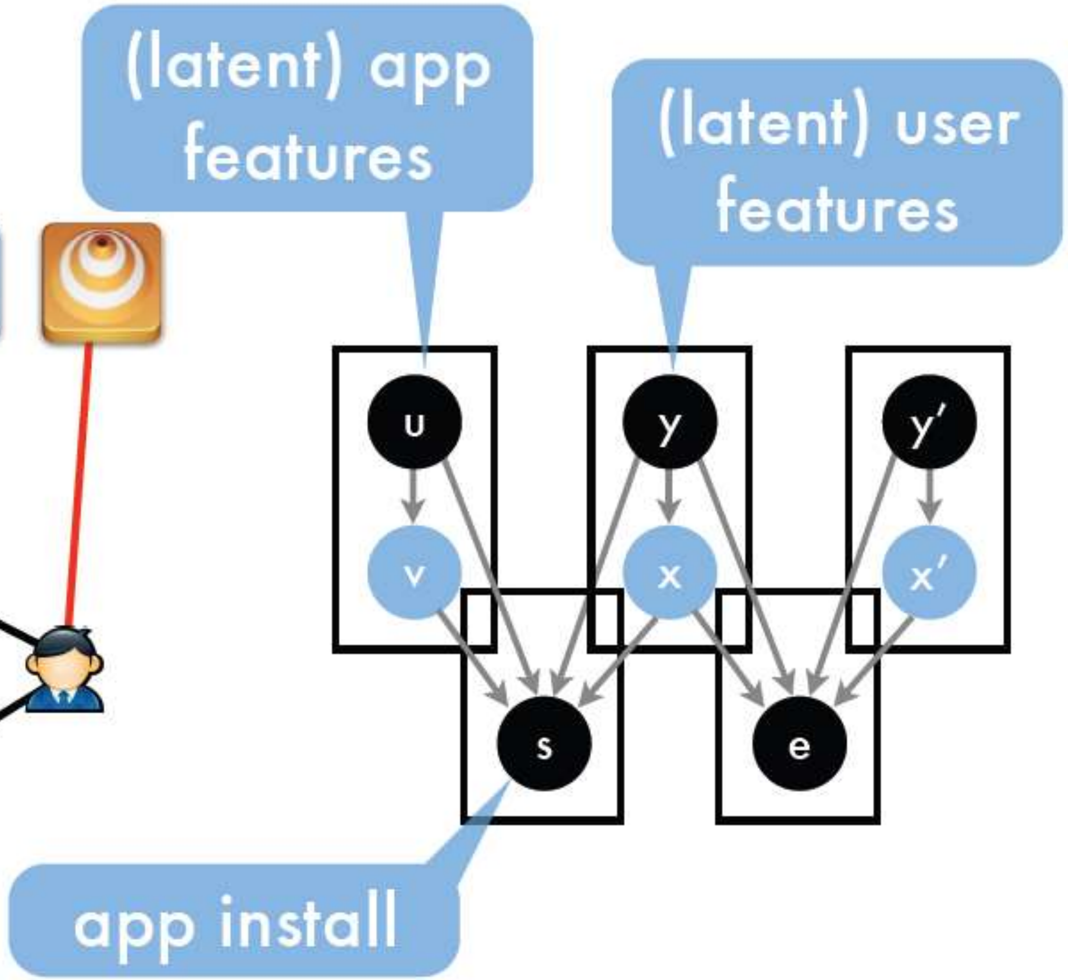
recommend apps based on friendship & interests



users with similar interests are more likely to connect

Social Recommendations





Social Recommendations

$$
\begin{aligned}
\textbf{minimize} \quad & \lambda_e \sum_{(i,j)} l(e_{ij}, x_i^\top x_j + y_i^\top W y_j) + \\
& \lambda_a \sum_{(i,j)} l(a_{ij}, x_i^\top v_j + y_i^\top M u_j) + \\
& \lambda_x \sum_{i} \gamma(x_i | y_i) + \lambda_v \sum_{i} \gamma(v_i | u_i) + \\
& \lambda_W \|W\|^2 + \lambda_M \|M\|^2 + \lambda_A \|A\|^2 + \lambda_B \|B\|^2
\end{aligned}
$$

social

app

reconstruction

regularizer

Social Recommendations

- Social connections can be used in combination with other approaches

- In particular, "friendships" can be fed into collaborative filtering methods in different ways

  - replace or modify user-user "similarity" by using social network information
  - use social connection as a part of the ML objective function as regularizer

Factorization Machines

- Generalization of regularized matrix (and tensor) factorization approaches combined with linear (or logistic) regression

- Problem: Each new adaptation of matrix or tensor factorization requires deriving new learning algorithms
  - Hard to adapt to new domains and add data sources
  - Hard to advance the learning algorithms across approaches
  - Hard to incorporate non-categorical variables

Rendle S. Factorization machines with libFM. ACM TIST, 2012.

Factorization Machines

- Approach: Treat input as a real-valued feature vector
  - Model both linear and pair-wise interaction of k features (i.e. polynomial regression)
  - Traditional machine learning will overfit
  - Factor pairwise interactions between features
  - Reduced dimensionality of interactions promote generalization

- Combines “generality of machine learning/regression with quality of factorization models”

Rendle S. Factorization machines with libFM. ACM TIST, 2012.

Factorization Machines

- Two categorical variables (u, i) encoded as real values:

Feature vector $\mathbf{x}$

	User				Movie				
	A	B	C	...	TI	NH	SW	ST	...
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...

一阶

二阶
交叉

- FM becomes identical to MF with biases:

$$f(\mathbf{x}) = b + w_u + w_i + \mathbf{v}_u^T \mathbf{v}_i$$

Rendle S. Factorization machines with libFM. ACM TIST, 2012.

Factorization Machines

- Makes it easy to add a time signal

Feature vector $\mathbf{x}$

	User				Movie					Time
	A	B	C	...	TI	NH	SW	ST	...	
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.2
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.6
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.61
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0.3
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0.5
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.1
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.8

- FM becomes as:

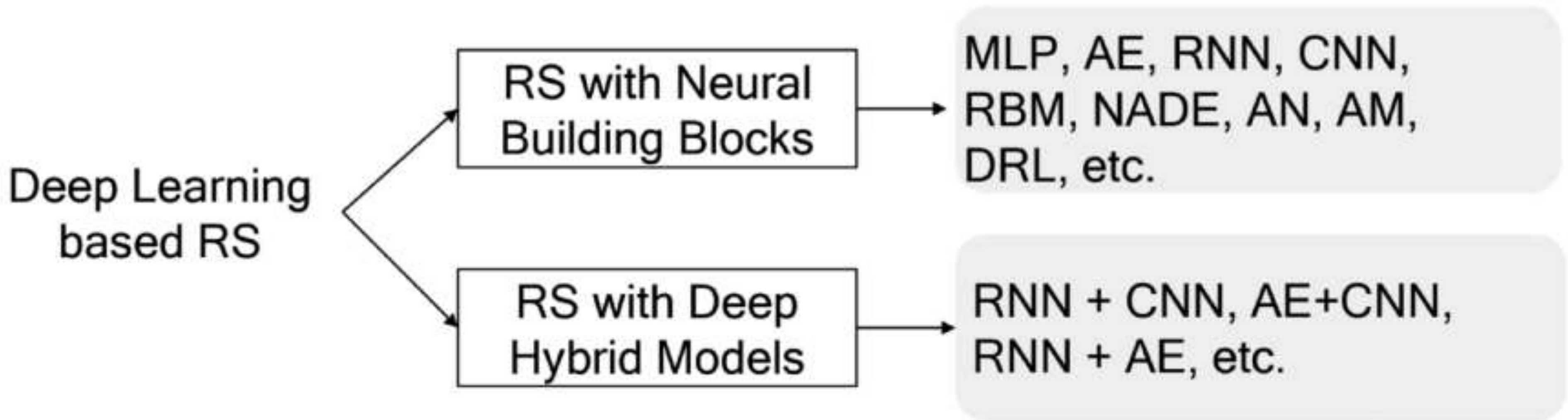
$$f(\mathbf{x}) = b + w_u + w_i + x_t w_t + \mathbf{v}_u^T \mathbf{v}_i + x_t \mathbf{v}_u^T \mathbf{v}_t + x_t \mathbf{v}_i^T \mathbf{v}_t$$

Rendle S. Factorization machines with libFM. ACM TIST, 2012.

Index

- What is recommender system?
- Traditional Methods
- **Deep Learning base Methods**
- RS Systems
- Conclusion

Deep Learning based RS



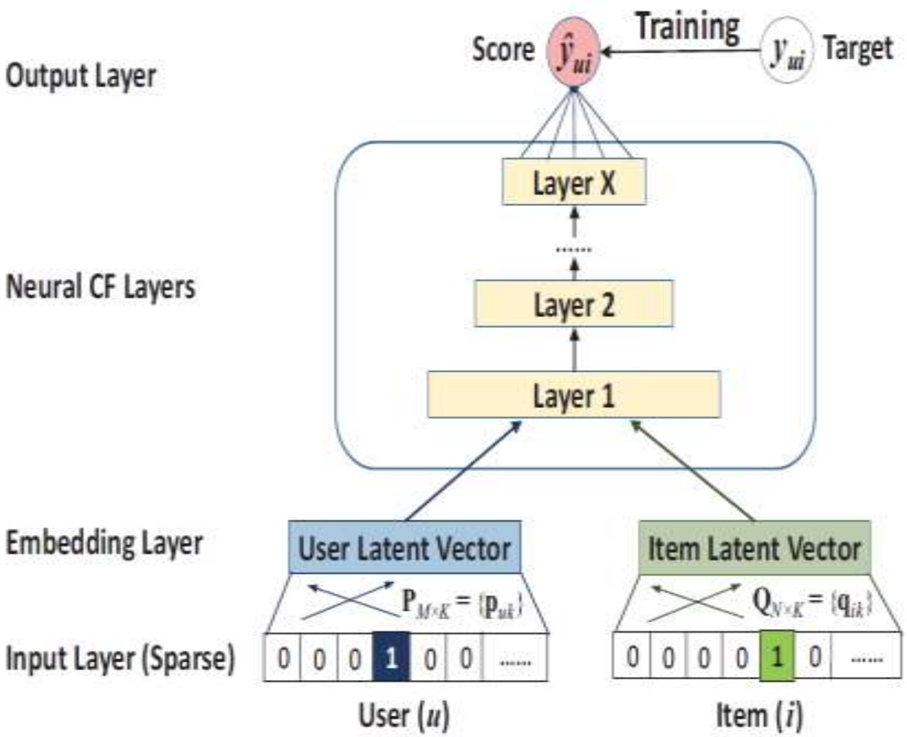
MLP: easily model the nonlinear interactions between users and items;

CNN: are capable of extracting local and global representations from heterogeneous data sources such as textual and visual information;

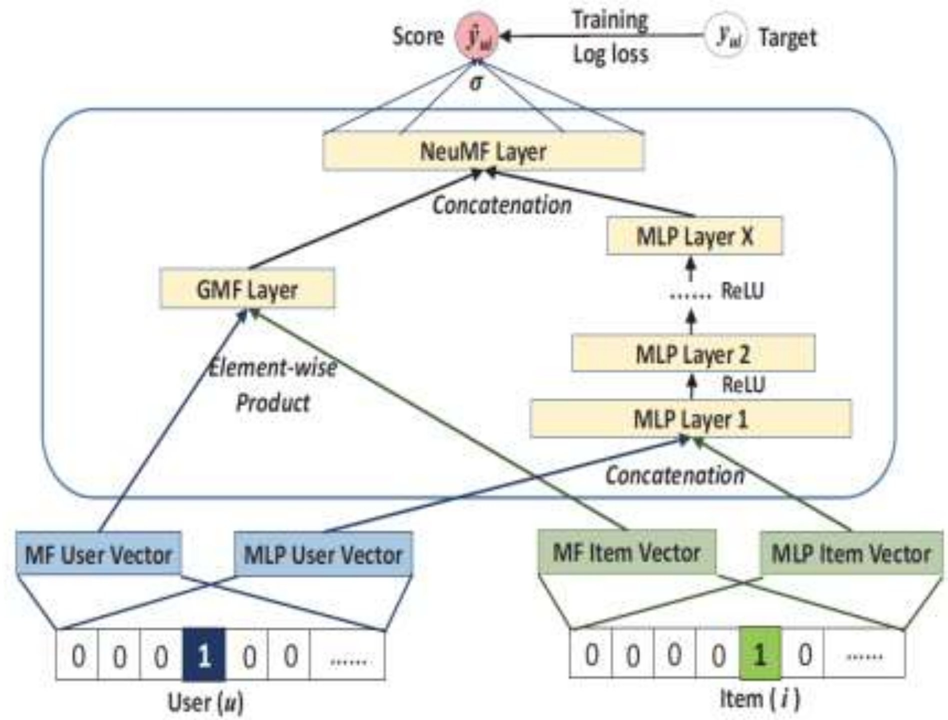
RNN: enable the recommender systems to model the temporal dynamics and sequential evolution of content information

Deep Learning based Recommender System: A Survey and New Perspectives. ACM Computing Surveys 2019

MLP based RS



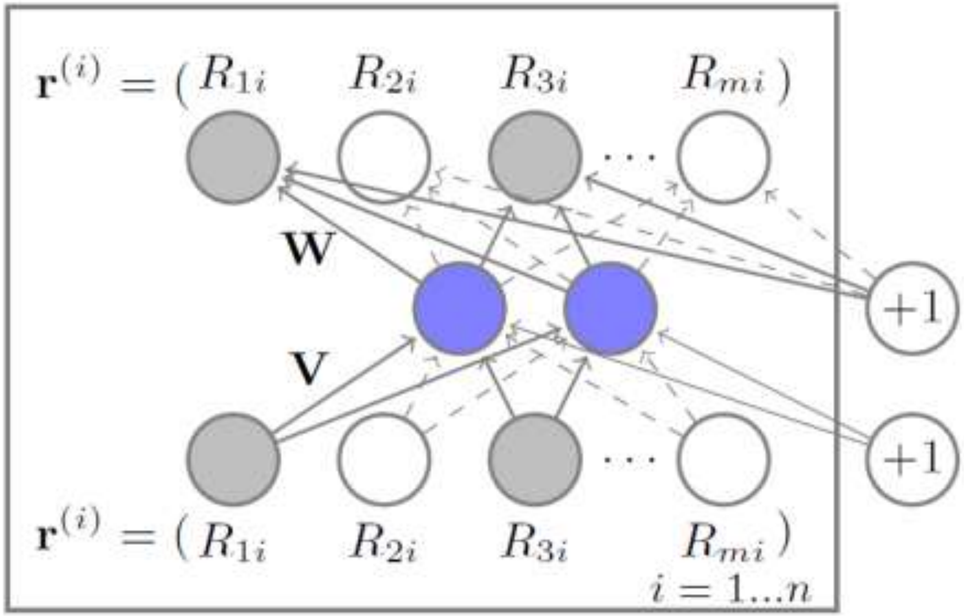
Neural collaborative filtering framework



Neural matrix factorization model fusing GMF and MLP

Xiangnan He, etc. Neural Collaborative Filtering. WWW2017

AutoEncoder based RS



Item-based AutoRec model

$$\min_{\theta} \sum_{i=1}^{n} ||\mathbf{r}^{(i)} - h(\mathbf{r}^{(i)}; \theta))||_{\mathcal{O}}^2 + \frac{\lambda}{2} \cdot (||\mathbf{W}||_F^2 + ||\mathbf{V}||_F^2)$$

where $h(\mathbf{r}; \theta) = f\left(\mathbf{W} \cdot g(\mathbf{V}\mathbf{r} + \boldsymbol{\mu}) + \mathbf{b}\right)$

Suvash Sedhain, etc. AutoRec: Autoencoders Meet Collaborative Filtering. WWW2015

CNN based RS

Music recommendation

Query	Most similar tracks (WMF)	Most similar tracks (predicted)
Jonas Brothers - Hold On	Jonas Brothers - Games Miley Cyrus - G.N.O. (Girl's Night Out) Miley Cyrus - Girls Just Wanna Have Fun Jonas Brothers - Year 3000 Jonas Brothers - BB Good	Jonas Brothers - Video Girl Jonas Brothers - Games New Found Glory - My Friends Over You My Chemical Romance - Thank You For The Venom My Chemical Romance - Teenagers
Beyoncé - Speechless	Beyoncé - Gift From Virgo Beyonce - Daddy Rihanna / J-Status - Crazy Little Thing Called Love Beyoncé - Dangerously In Love Rihanna - Haunted	Daniel Bedingfield - If You're Not The One Rihanna - Haunted Alejandro Sanz - Siempre Es De Noche Madonna - Miles Away Lil Wayne / Shanell - American Star
Coldplay - I Ran Away	Coldplay - Careful Where You Stand Coldplay - The Goldrush Coldplay - X & Y Coldplay - Square One Jonas Brothers - BB Good	Arcade Fire - Keep The Car Running M83 - You Appearing Angus & Julia Stone - Hollywood Bon Iver - Creature Fear Coldplay - The Goldrush
Daft Punk - Rock'n Roll	Daft Punk - Short Circuit Daft Punk - Nightvision Daft Punk - Too Long (Gonzales Version) Daft Punk - Aerodynamite Daft Punk - One More Time / Aerodynamic	Boys Noize - Shine Shine Boys Noize - Lava Lava Flying Lotus - Pet Monster Shotglass LCD Soundsystem - One Touch Justice - One Minute To Midnight

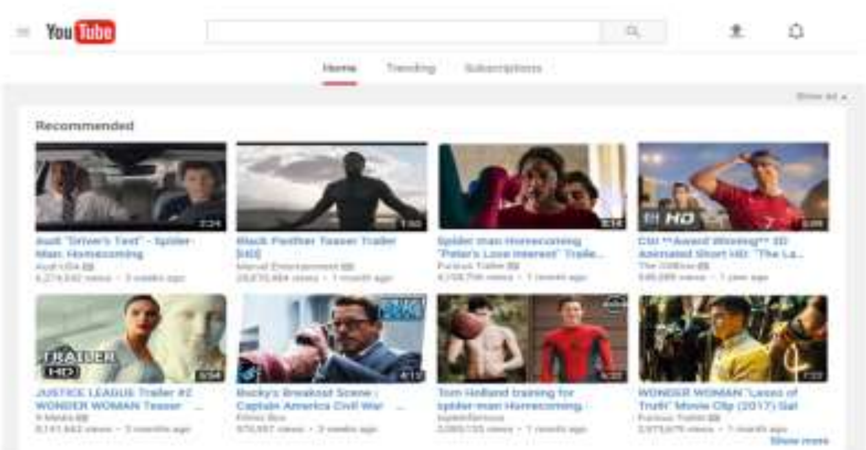
Table 4: A few songs and their closest matches in terms of usage patterns, using latent factors obtained with WMF and using latent factors predicted by a convolutional neural network.

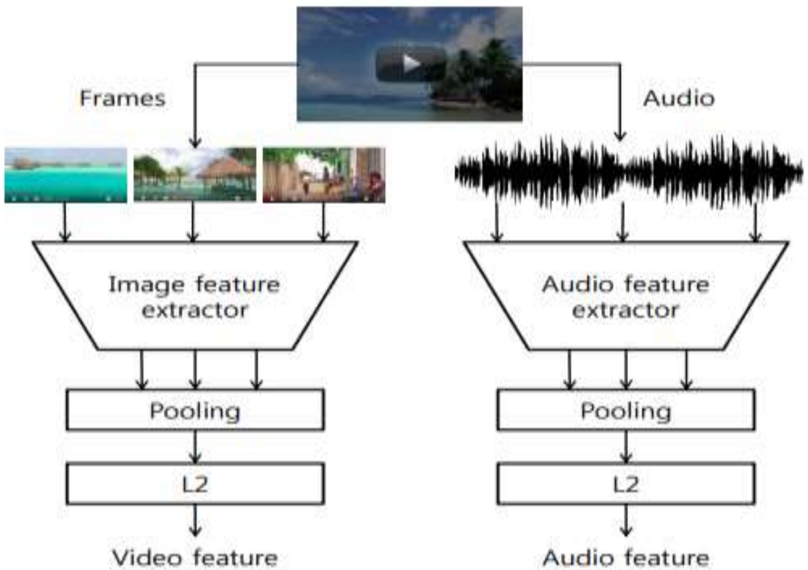
$$\min_{\theta} \sum_{i} ||y_i - y'_i||^2$$

$$\min_{\theta} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y'_i)^2$$

Aaron Van den Oord, NIPS2013

Video recommendation





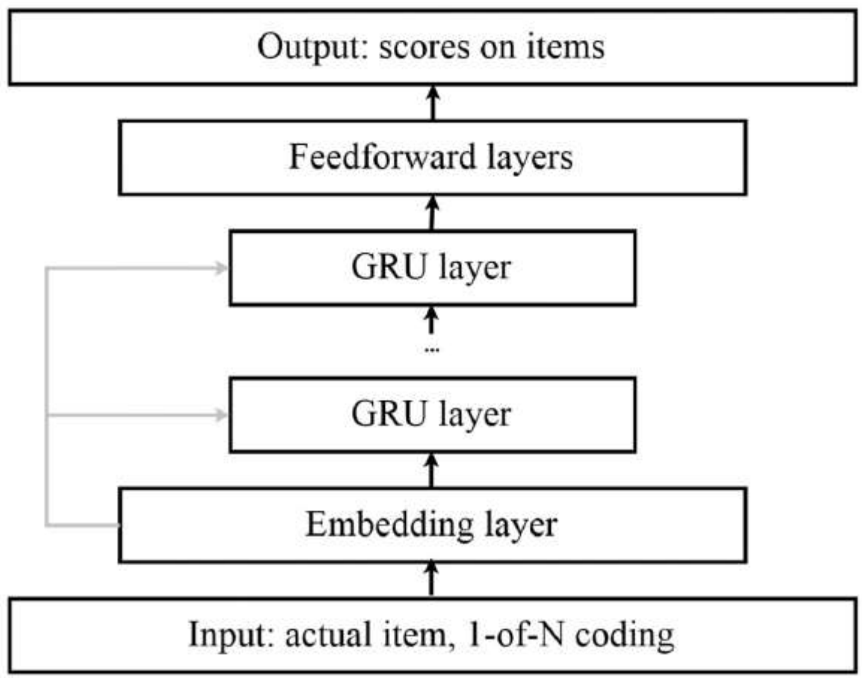
Joonseok Lee, KDD18

Deep content-based music recommendation. NIPS2013

Collaborative Deep Metric Learning for Video Understanding, KDD2018

RNN based RS

GRU4Rec:



$$\mathcal{L}_s = \frac{1}{S} \sum_{j=1}^{S} \sigma(\hat{r}_{sj} - \hat{r}_{si}) + \sigma(\hat{r}_{sj}^2)$$

Neural collaborative filtering framework

B. Hidasi. Session-based recommendations with recurrent neural networks. ICLR2016

Index

- What is recommender system?
- Traditional Methods
- Deep Learning base Methods
- **RS Systems**
- Conclusion

TOP 10 开源的推荐系统

- SVDFeature http://svdfeature.apexlab.org/wiki/Main_Page
- libMF http://www.csie.ntu.edu.tw/~cjlin/libmf/
- libFM http://www.libfm.org/
- Lenskit http://lenskit.grouplens.org/
- GraphLab GraphLab - Collaborative Filtering
- Mahout http://mahout.apache.org/
- Myrrix http://myrrix.com/
- EasyRec http://easyrec.org/
- Waffles http://waffles.sourceforge.net/
- RapidMiner http://rapidminer.com/

工业界的推荐系统

视频类：
Netflix：很多方法的融合
Hulu：主要是item based CF
Youtube：开始是random walk，后来改为类似item based CF的方法

图书类：
Amazon：好多方法都用了，主要是 item based CF

资讯类：
google news：用了CF和bayesian的方法。
digg：算法是 热门度+topic driven user based CF，

音乐类：
last.fm：用的是CF。
yahoo music：参考Koren的论文。
pandora：音乐基因项目，主要依赖专家标注。

社交类
facebook：算法叫Edgerank。
twitter：主要场景是推荐其它用户，参考官方介绍。

Widely used data

- Movie
  - MovieLens http://grouplens.org/datasets/movielens/
  - Netflix https://www.netflix.com/cn/
- Book
  - Amazon books http://www.amazon.com/b/ref=usbk_surl_books/?node=283155
  - Book-Crossing http://grouplens.org/datasets/book-crossing/
- Music
  - Last.fm http://www.last.fm/
- Food
  - Dianping http://www.dianping.com/
- Else…
  - Epinion http://www.datatang.com/data/11849

Index

- What is recommender system?
- Traditional Methods
- Deep Learning base Methods
- RS Systems
- **Conclusion**

Challenging Problems

- Data sparsity：
  - Netflix Dataset: nearly 48,000 users and 1,700 items, only 1% observations
- Curse of dimensionality
  - Users' features can be represented as many ways
- Cold start:
  - Many new users sign in and many new items are added
- Personalization:
  - Different user has different taste

Thanks!

Q&A