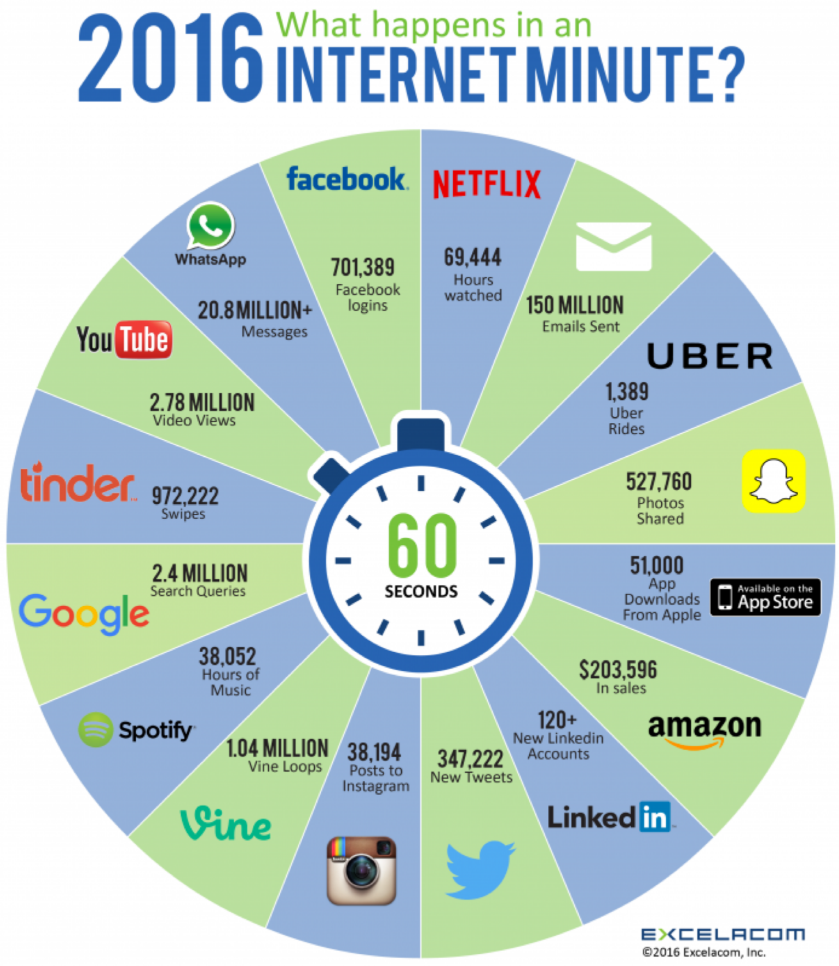
Social Recommendation

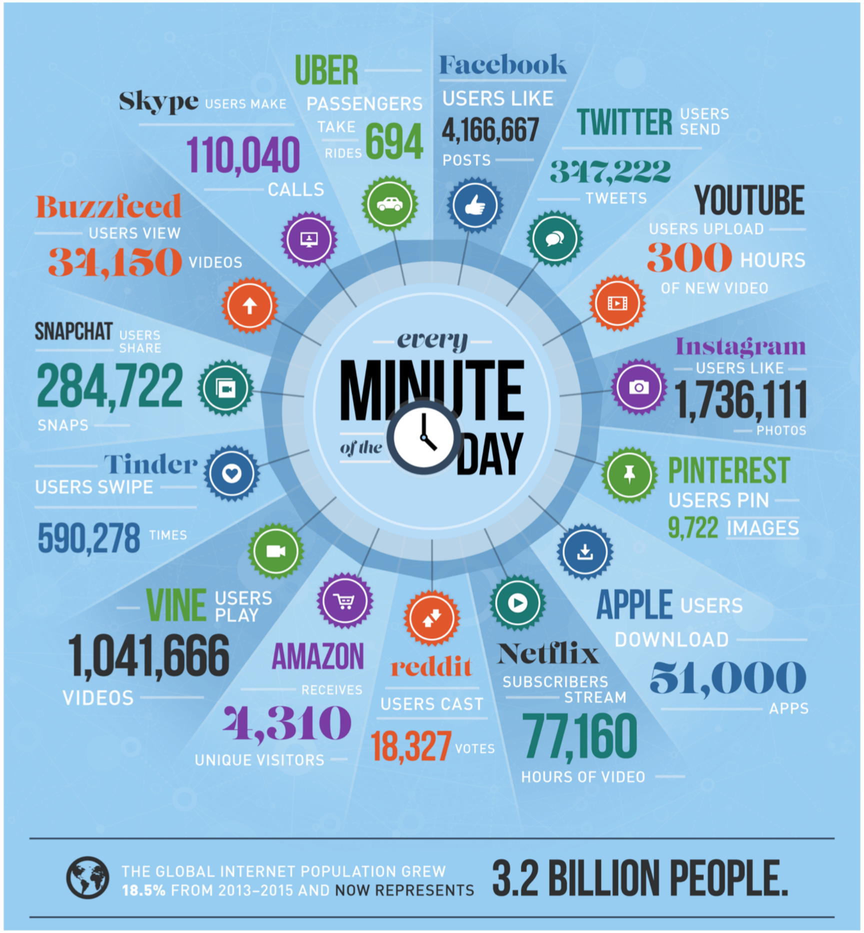
Dingqi Yang
Senior Researcher
eXascale Infolab, University of Fribourg, Switzerland
Homepage: https://sites.google.com/site/yangdingqi/

Outline

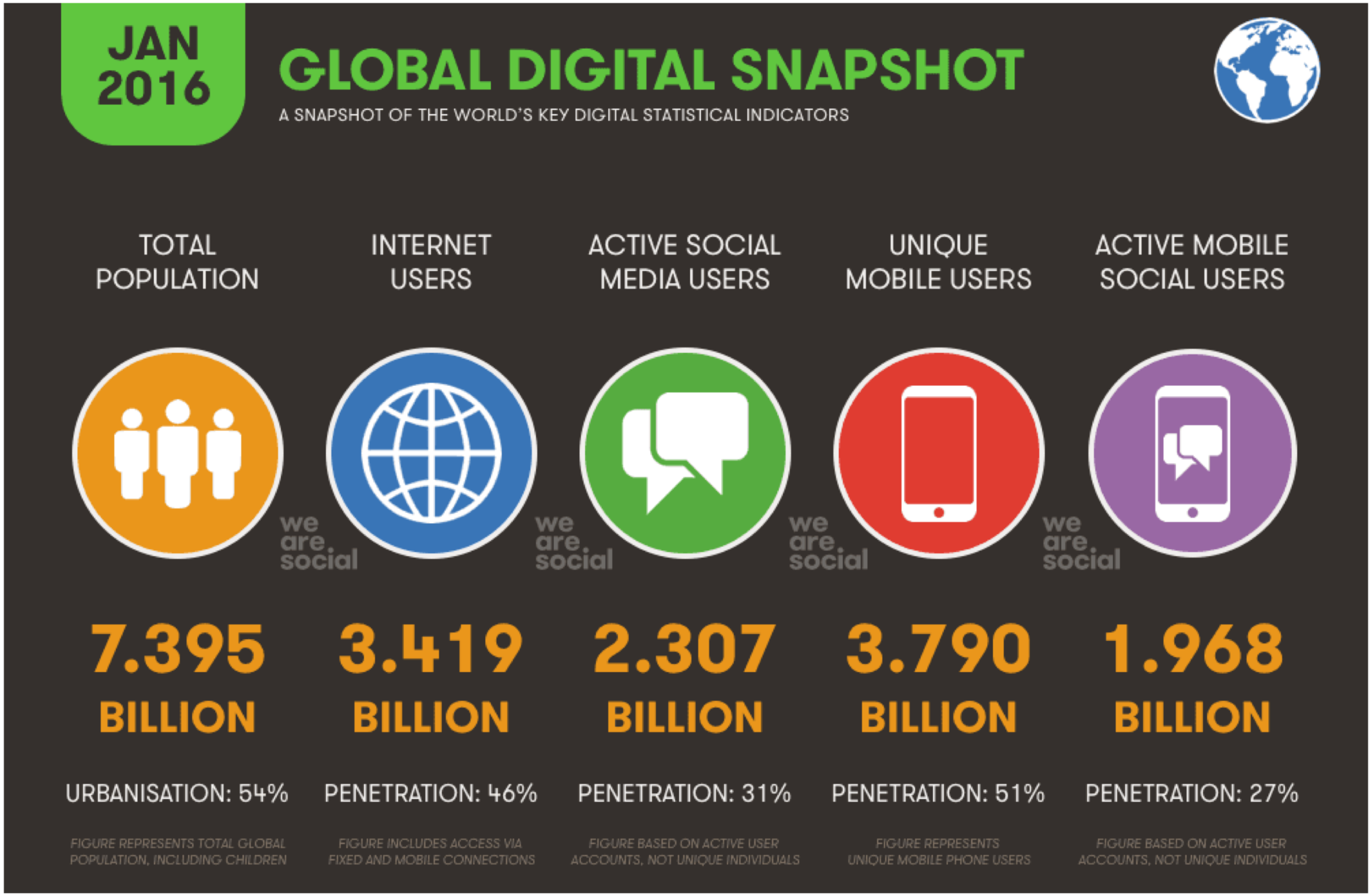
- **Introduction to Recommendation Systems**
- **Recommendation Techniques**
  - Content-based filtering
  - Collaborative filtering
    - Memory-based methods
    - Model-based methods (Matrix factorization)
- **Social Recommendation**
  - Social relationship in recommendation
- **Concluding remarks**

An Internet Minute





Social Media Statistics



Social Media Statistics

Social Media Statistics

- **Okay, so many users and much data, then what ?**

Benji · a year ago

Hooray for tons of data! 99% of which is total useless rubbish!

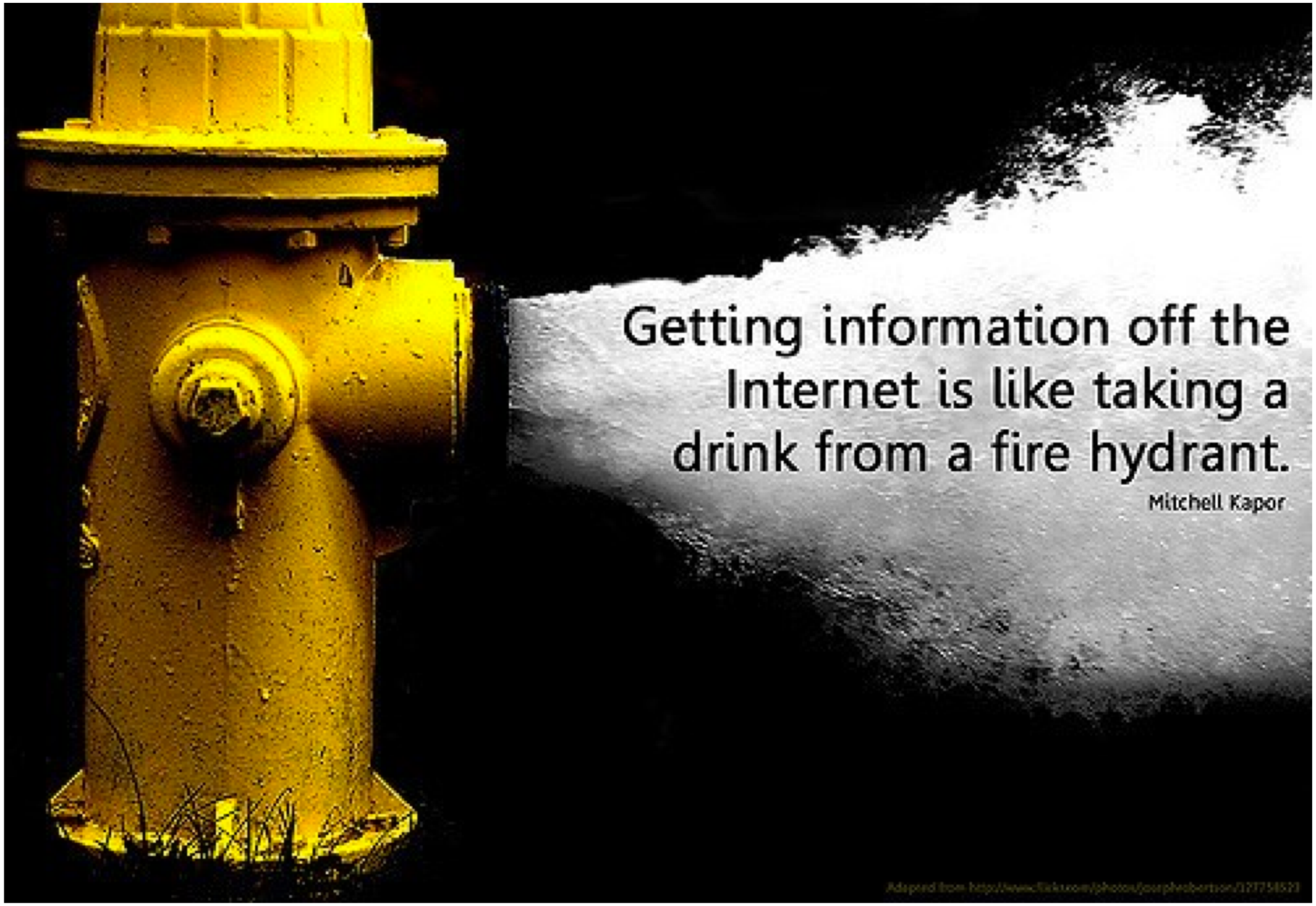
2 ^ | ˅ · Reply · Share ›

Pat Hennel · 20 days ago

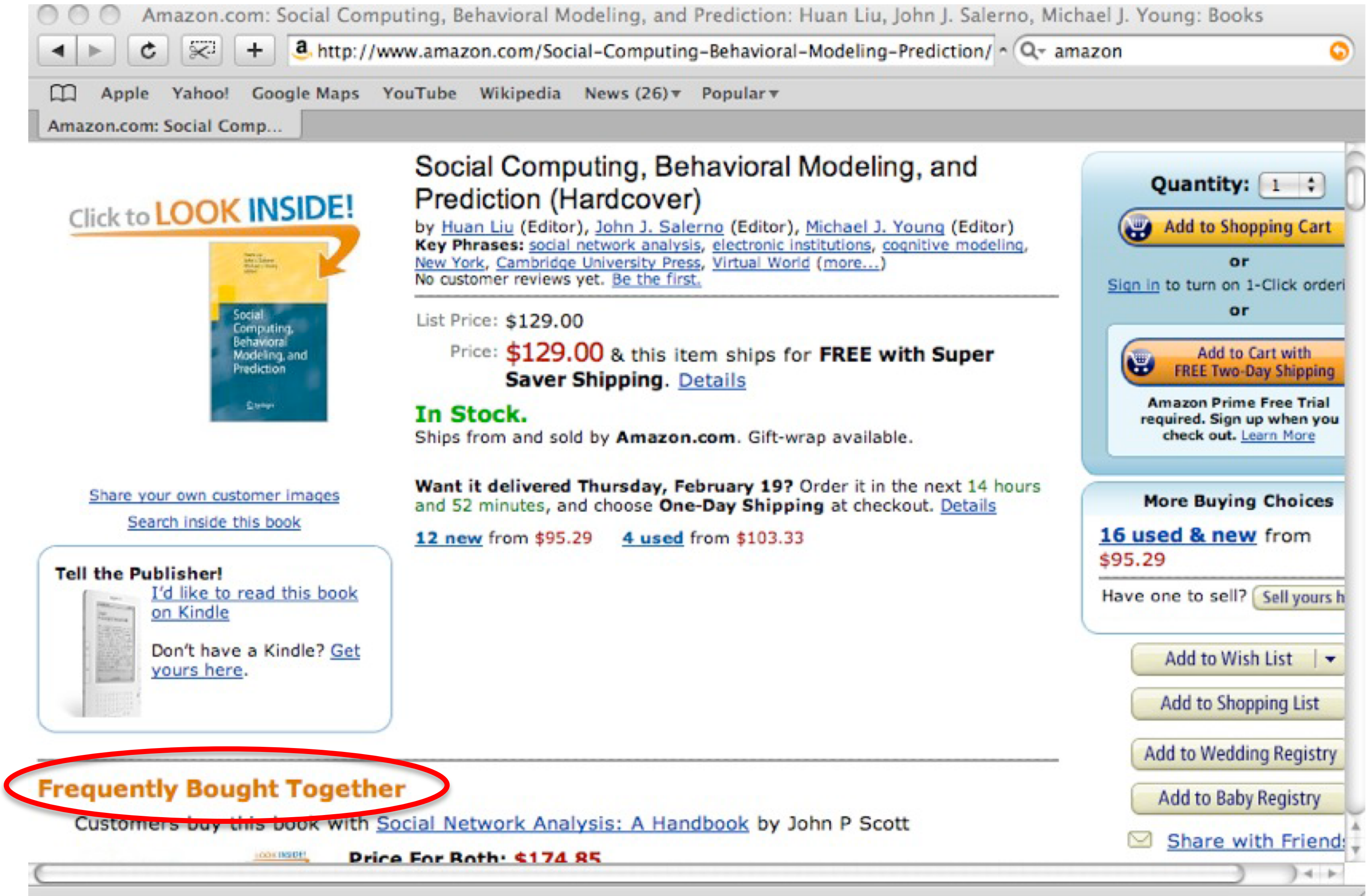
Astounding amounts of data are generated every minute, which means that the possibilities are endless for those who use this data for BI. The goal for any enterprise is to zero in on the data that is meaningful and relevant. Failure to do that could result in poor quality insights, which could also result in a loss of revenue.

^ | ˅ · Reply · Share ›

Information Overload

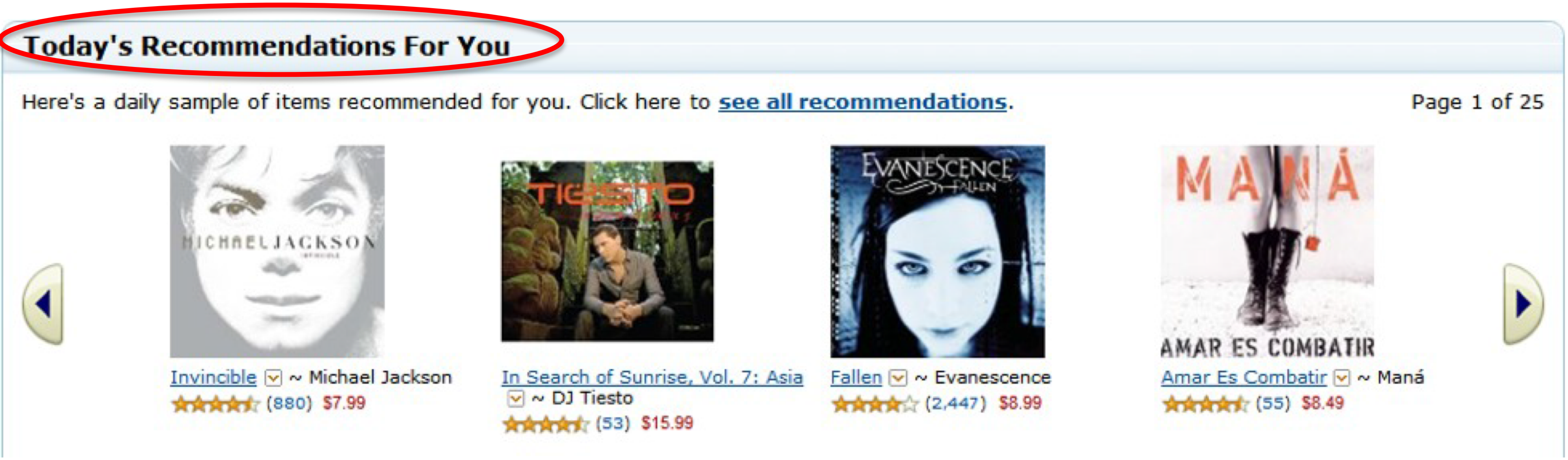


Real Life Examples



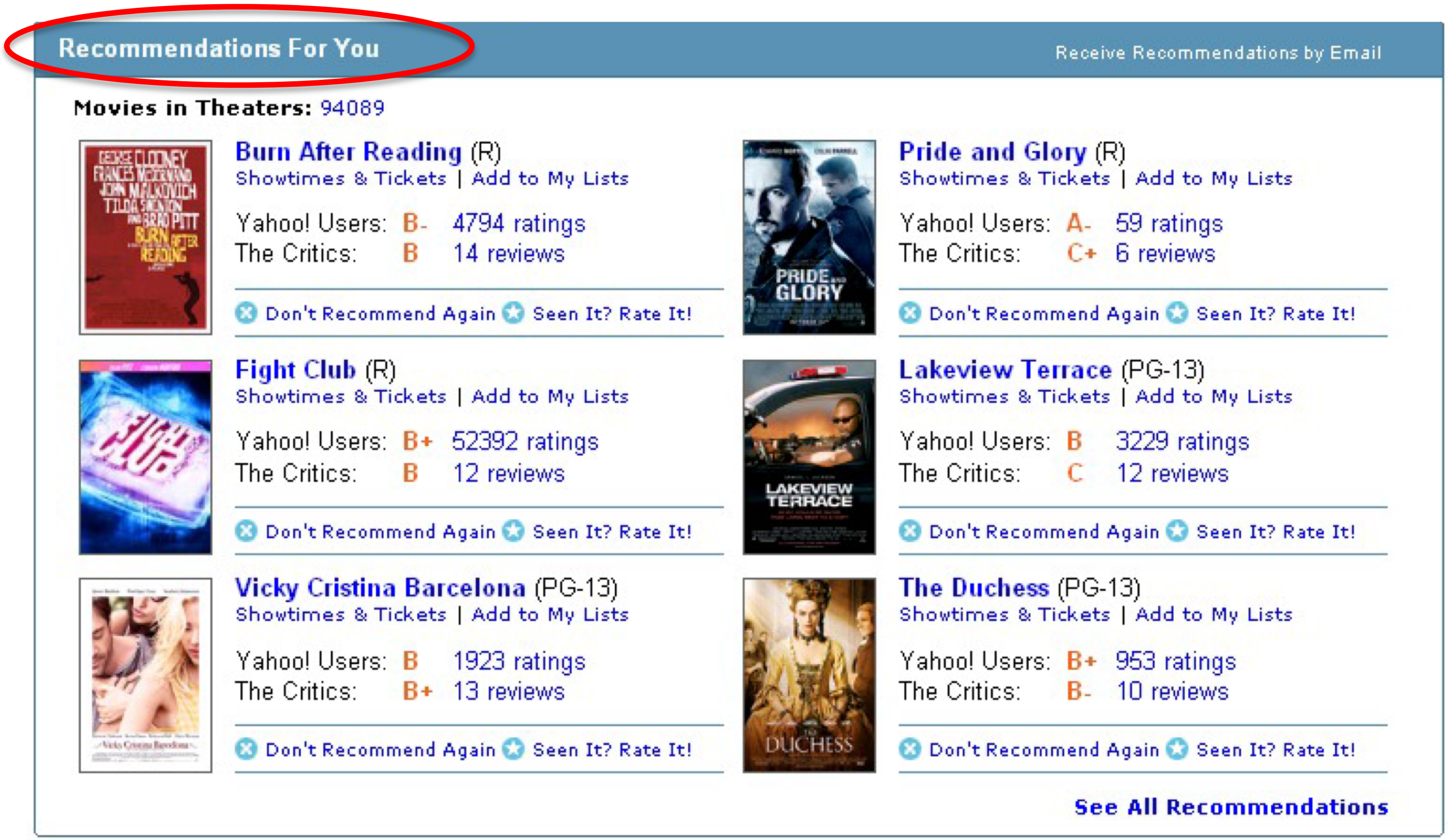
Real Life Examples

Real Life Examples



Real Life Examples

YAHOO! MOVIES



Real Life Examples

Real Life Examples

iLike™

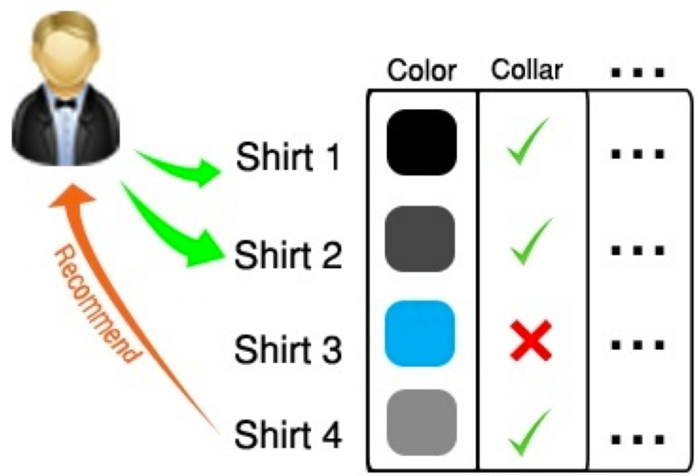
Recommendation Systems

- **Content-based Filtering**
  - Recommend items based on key-words
  - More appropriate for information retrieval

- **Collaborative Filtering (CF)**
  - Look at users with similar rating styles
  - Look at similar items for each item

Content-based Filtering

- **Matching user preference with content**
  - Requirement
    - Some information describing items (e.g., genre of movies)
    - Some sort of user profile describing the user's preference
  - "Similarity" is computed based on the item attributes
  - The task: Match a user with the content that may interest the user.



Collaborative Filtering: The Tasks

- **Which item should be recommended?**
  - Predict the unknown rating

Items

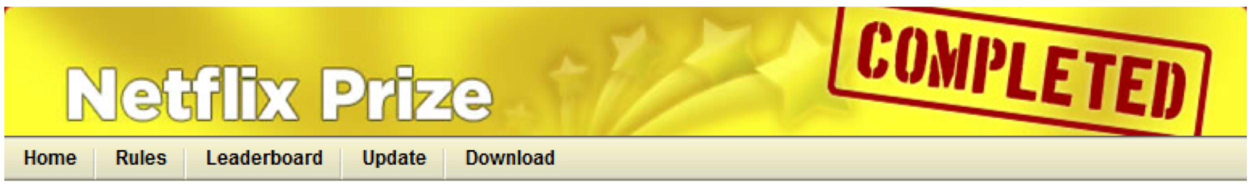
Users

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	4		4	3	3	5	
u_2		3		2			
u_3		2		5			1
u_4		5		1		3	
u_5	5		4		4		1
u_6	?		3		4	2	1

Underling assumption: personal tastes are correlated -
Active user will prefer those items which the similar users prefer.

Netflix Prize

- **Netflix Prize : a movie recommendation challenge**
  - 1 Million USD for the best performing recommendation algorithm



Leaderboard

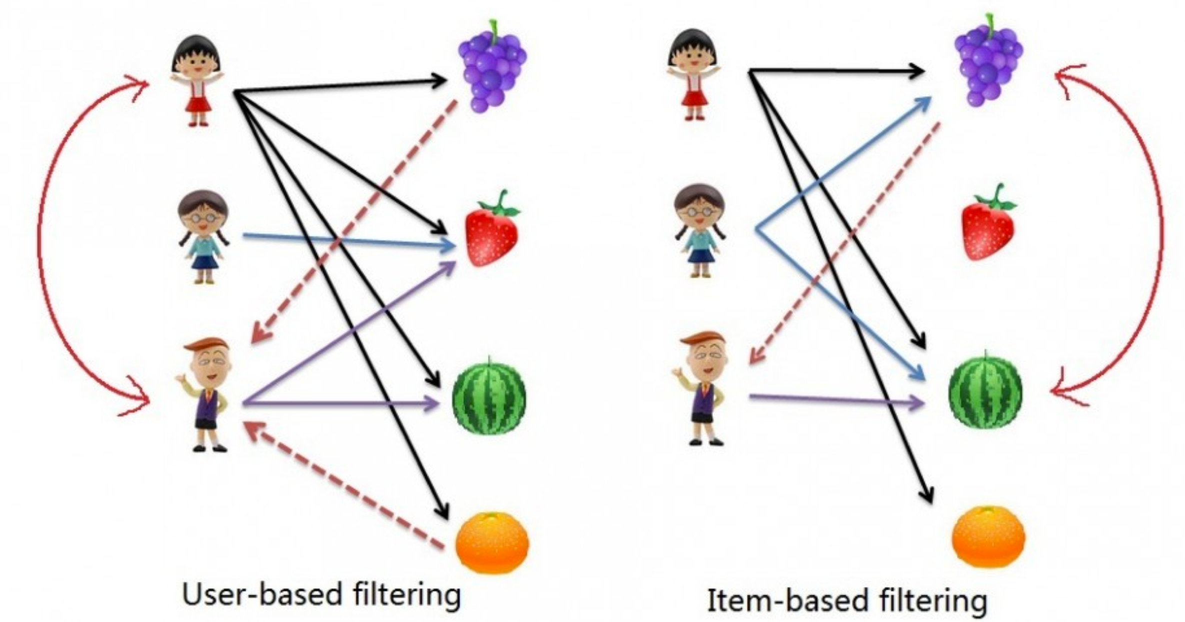
Showing Test Score. Click here to show quiz score

Display top 20 **leaders.**

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

Collaborative Filtering

- **A method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating).**

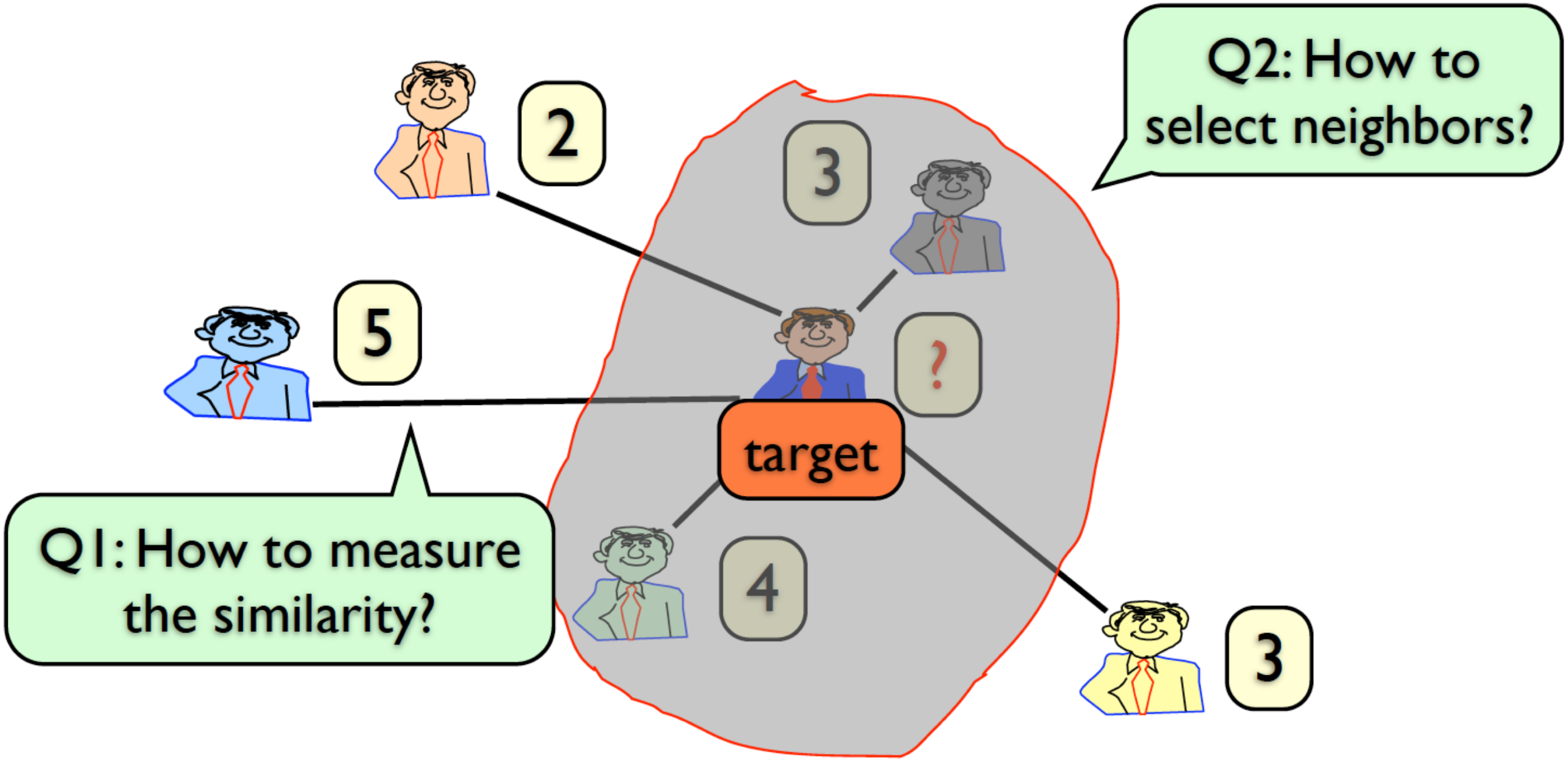


Collaborative Filtering

- **Memory-based (Neighborhood-based)**
  - User-based
  - Item-based

- **Model-based**
  - Matrix Factorization
  - Clustering Methods
  - Bayesian Methods
  - etc.

User-User Similarity



User-based Collaborative Filtering

Items

Users

4		3	3	3	1	
	3		2			
	2		5			1
	5		1		3	
5		4		4		1
?		3		4	2	1

User-based Collaborative Filtering

- **Predict the ratings of active users based on the ratings of similar users found in the user-item matrix**
- **Similarity between two users**
  - Pearson correlation coefficient

$$\mathbf{S}_{ij} = \frac{\sum_{k \in I} (R_{ik} - \bar{R}_i) \cdot (R_{jk} - \bar{R}_j)}{\sqrt{\sum_{k \in I} (R_{ik} - \bar{R}_i)^2} \sqrt{\sum_{k \in I} (R_{jk} - \bar{R}_j)^2}}$$

  - Cosine similarity

$$\mathbf{S}_{ij} = \frac{\sum_{k \in I} R_{ik} \cdot R_{jk}}{\sqrt{\sum_{k \in I} R_{ik}^2} \sqrt{\sum_{k} R_{jk}^2}}$$

  - I: items rated by both users

Items

Users

4		3	3	3	1	
	3		2			
	2		5			1
	5		1		3	
5		4		4		1
?		3		4	2	1

User-based Collaborative Filtering

- **Pearson correlation coefficient**

$$\mathbf{S}_{ij} = \frac{\sum_{k \in I}(R_{ik} - \bar{R}_i) \cdot (R_{jk} - \bar{R}_j)}{\sqrt{\sum_{k \in I}(R_{ik} - \bar{R}_i)^2}\sqrt{\sum_{k \in I}(R_{jk} - \bar{R}_j)^2}}$$

  - I: items rated by both users

Items

Users

4		3	3	3	1	
	3		2			
	2		5			1
	5		1		3	
5		4		4		1
?		3		4	2	1

$$\overline{R}_i = 3, \overline{R}_j = 2.67$$

$$S_{ij} = \frac{1 * 0.33 + 1 * 1.33 + (-2) * (-1.67)}{\sqrt{1^2 + 1^2 + (-2)^2}\sqrt{0.33^2 + 1.33^2 + (-1.67)^2}}$$

$$S_{ij} = 0.9449$$

User-based Collaborative Filtering

- **Weighted average rating:** $\hat{R_{i,k}} = \overline{R}_i + \frac{\sum_{u_j \in N_i} S_{ij}(R_{jk} - \overline{R}_j)}{\sum_{u_j \in N_i} S_{ij}}$
  - where N_i is the set of users (neighbors) who have rated the k-th item

Items

Users

4		3	3	3	1	
	3		2			
	2		5			1
	5		1		3	
5		4		4		1
?		3		4	2	1

0.8660

Pearson correlation coefficient

0.9449

$$\hat{R_{i,k}} = 2.5 + \frac{0.8660 * (4 - 2.8) + 0.9449 * (5 - 3.5)}{0.8860 + 0.9449} = 3.8565$$

Item-Item Similarity

- **Search for similarities among items**

- **Item-Item similarity is more stable than user-user similarity**
  - People change their preference over time, but items rarely change their features.

Items

Users

4		3	3	3	1	
	3		2			
	2		5			1
	5		1		3	
5		4		4		1
?		3		4	2	1

Item-based Collaborative Filtering

- **Similar as in user-user similarity but on item vectors**
  - Pearson correlation coefficient
  - Cosine similarity
  - Look for users who rated both items

Items

Users

4		3	3	3	1	
	3		2			
	2		5			1
	5		1		3	
5		4		4		1
?		3		4	2	1

$$\mathbf{S}_{ij} = \frac{\sum_{k\in I}(R_{ik}-\bar{R}_i)\cdot(R_{jk}-\bar{R}_j)}{\sqrt{\sum_{k\in I}(R_{ik}-\bar{R}_i)^2}\sqrt{\sum_{k\in I}(R_{jk}-\bar{R}_j)^2}}$$

$$\hat{R_{i,k}} = \overline{R}_i + \frac{\sum_{u_j\in N_i} S_{ij}(R_{jk}-\overline{R}_j)}{\sum_{u_j\in N_i} S_{ij}}$$

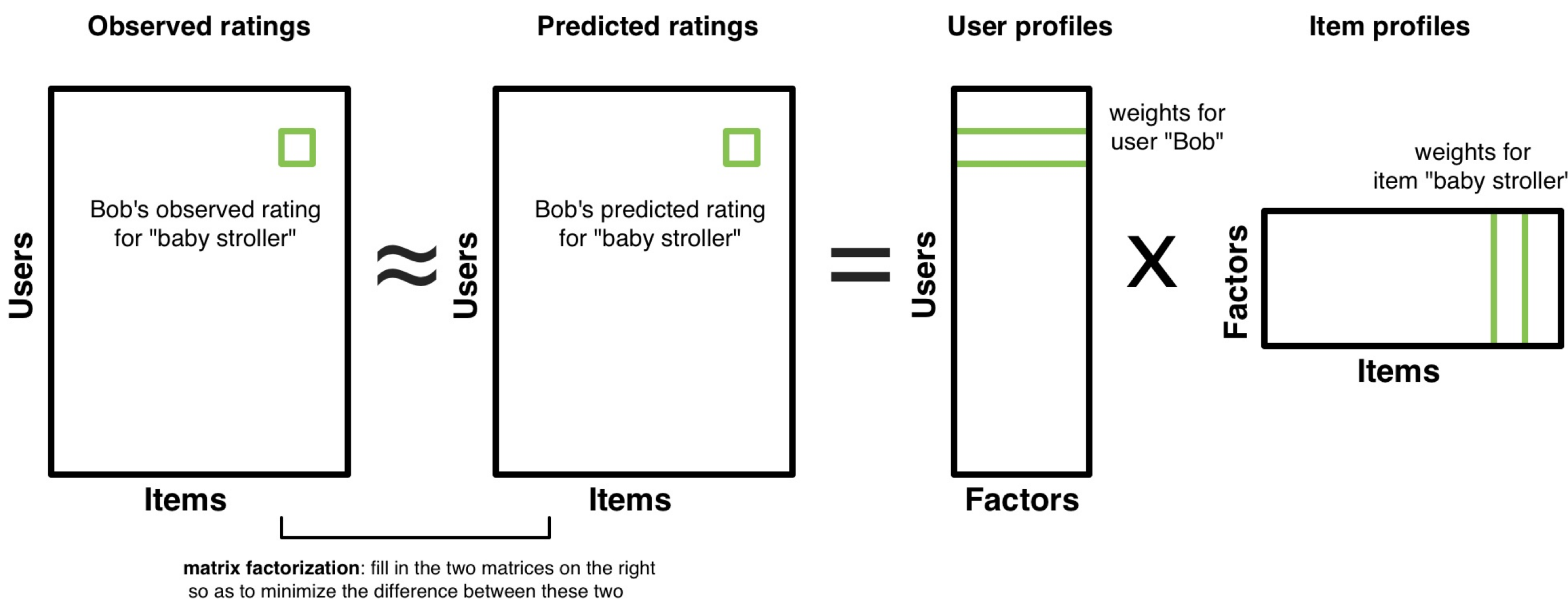
Collaborative Filtering

- **Memory-based (Neighborhood-based)**
  - User-based
  - Item-based

- **Model-based**
  - Matrix Factorization
  - Clustering Methods
  - Bayesian Methods
  - etc.

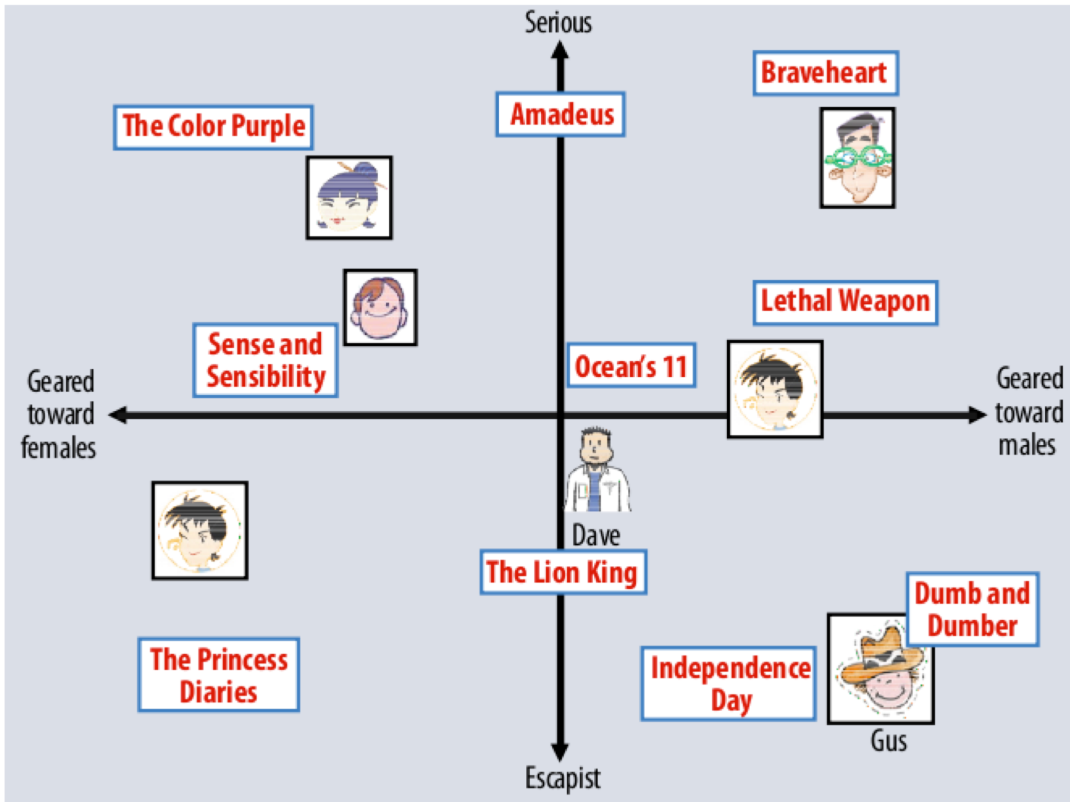
Matrix Factorization

- **Matrix factorization: characterizes both items and users by vectors of factors inferred from item rating patterns.**



Matrix Factorization

- **Interpretation of latent factors**
  - However, latent factors are practically often implicit and inexplicable.



Matrix Factorization

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	5	2		3		4		
u_2	4	3			5			
u_3	4		2				2	4
u_4								
u_5	5	1	2		4	3		
u_6	4	3		2	4		3	5

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	5	2	**2.5**	3	**4.8**	4	**2.2**	**4.8**
u_2	4	3	**2.4**	**2.9**	5	**4.1**	**2.6**	**4.7**
u_3	4	**1.7**	2	**3.2**	**3.9**	**3.0**	2	4
u_4	**4.8**	**2.1**	**2.7**	**2.6**	**4.7**	**3.8**	**2.4**	**4.9**
u_5	5	1	2	**3.4**	4	3	**1.5**	**4.6**
u_6	4	3	**2.9**	2	4	**3.4**	3	5

$$U = \begin{bmatrix} 1.55 & 1.22 & 0.37 & 0.81 & 0.62 & -0.01 \\ 0.36 & 0.91 & 1.21 & 0.39 & 1.10 & 0.25 \\ 0.59 & 0.20 & 0.14 & 0.83 & 0.27 & 1.51 \\ 0.39 & 1.33 & -0.43 & 0.70 & -0.90 & 0.68 \\ 1.05 & 0.11 & 0.17 & 1.18 & 1.81 & 0.40 \end{bmatrix} \quad V = \begin{bmatrix} 1.00 & -0.05 & -0.24 & 0.26 & 1.28 & 0.54 & -0.31 & 0.52 \\ 0.19 & -0.86 & -0.72 & 0.05 & 0.68 & 0.02 & -0.61 & 0.70 \\ 0.49 & 0.09 & -0.05 & -0.62 & 0.12 & 0.08 & 0.02 & 1.60 \\ -0.40 & 0.70 & 0.27 & -0.27 & 0.99 & 0.44 & 0.39 & 0.74 \\ 1.49 & -1.00 & 0.06 & 0.05 & 0.23 & 0.01 & -0.36 & 0.80 \end{bmatrix}$$

Matrix Factorization

- **Matrix Factorization in Collaborative Filtering**
  - To fit the product of two (low rank) matrices to the observed rating matrix.
  - To find two latent user and item feature matrices.
  - To use the fitted matrix to predict the unobserved ratings.

$$\begin{pmatrix} \boxed{u_{11} \quad \cdots \quad u_{1k}} \\ \vdots \quad \ddots \quad \vdots \\ u_{m1} \quad \cdots \quad u_{mk} \end{pmatrix} \begin{pmatrix} \boxed{\begin{matrix} v_{11} \\ \vdots \\ v_{k1} \end{matrix}} & \begin{matrix} \cdots \\ \ddots \\ \cdots \end{matrix} & \begin{matrix} v_{1n} \\ \vdots \\ v_{kn} \end{matrix} \end{pmatrix}$$

User-specific latent feature vector

Item-specific latent feature column vector

Matrix Factorization

- Optimization Problem

Given a *m x n* rating matrix *R*, to find two matrices $U \in \mathbb{R}^{l \times m}$ and $V \in \mathbb{R}^{l \times n}$,

$$R \approx U^T V,$$

where $l < \min(m, n)$, is the number of factors

Matrix Factorization

- **Models**
  - SVD-like Algorithm
  - Regularized Matrix Factorization (RMF)
  - Probabilistic Matrix Factorization (PMF)
  - Non-negative Matrix Factorization (NMF)

SVD-like Algorithm

- **Minimizing**

$$\frac{1}{2}||R - U^T V||_F^2$$

- **For collaborative filtering**

$$\min_{U,V} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T V_j)^2$$

where I_{ij} is the indicator function that is equal to 1 if user u_i rated item v_j and equal to 0 otherwise.

Regularized Matrix Factorization

- Minimize the loss based on the observed ratings with regularization terms to avoid over-fitting problem

$$\min_{U,V} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T V_j)^2 + \boxed{\frac{\lambda_1}{2} ||U||_F^2 + \frac{\lambda_2}{2} ||V||_F^2}$$

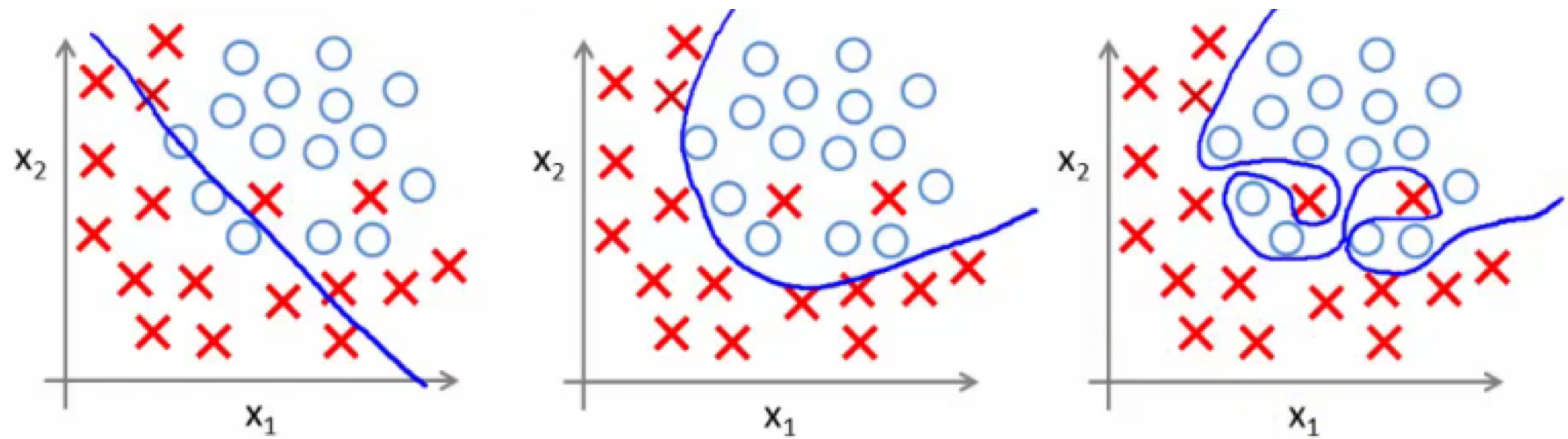
Regularization terms

where $\lambda_1, \lambda_2 > 0$

- The problem can be solved by simple stochastic gradient descent algorithm.

What is the Over-Fitting Problem?

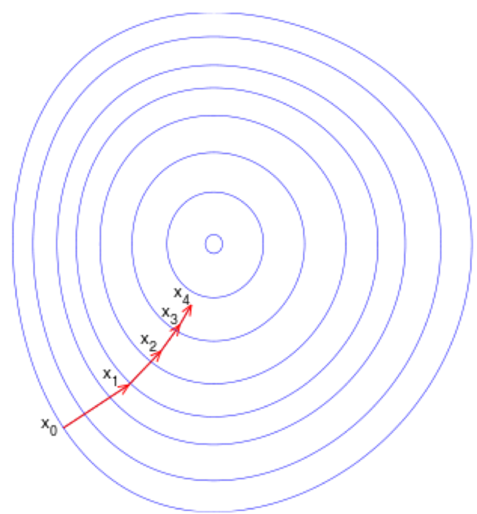
- **Over-fitting: a statistical model describes random error or noise instead of the underlying relationship.**



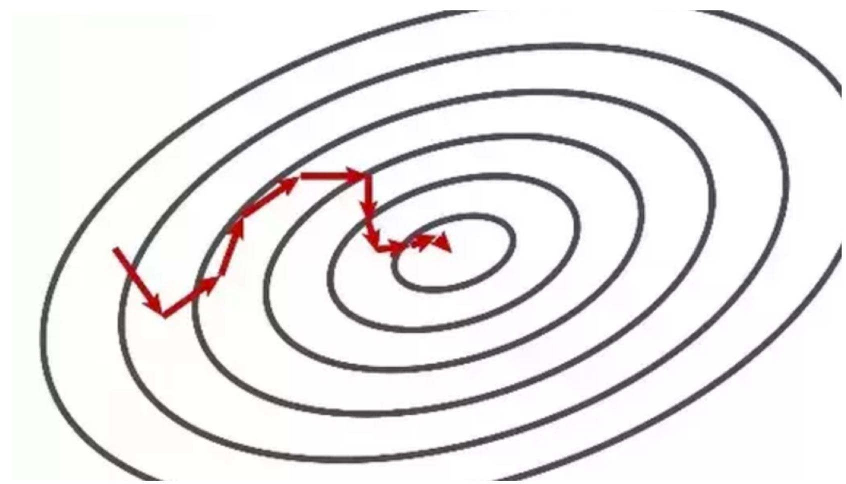
Stochastic Gradient Descent (SGD)

- **A stochastic approximation of the gradient descent optimization method.**
- **An objective function written as a sum of differentiable functions**

$$\min_{U,V} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T V_j)^2$$



Gradient descent



Stochastic gradient descent

SGD for Matrix Factorization

- **A close look at objective loss function**

$$e_{i,j} = \frac{1}{2}(R_{i,j} - \sum_k U_{k,i}V_{k,j})^2 + \frac{\lambda_1}{2}||U||^2 + \frac{\lambda_2}{2}||V||^2$$

- **Computing the partial derivative w.r.t. U and V**

$$\frac{\partial e_{i,j}}{\partial U_{k,i}} = -(R_{i,j} - \sum_k U_{k,i}V_{k,j})V_{k,j} + \lambda_2 U_{k,i}$$

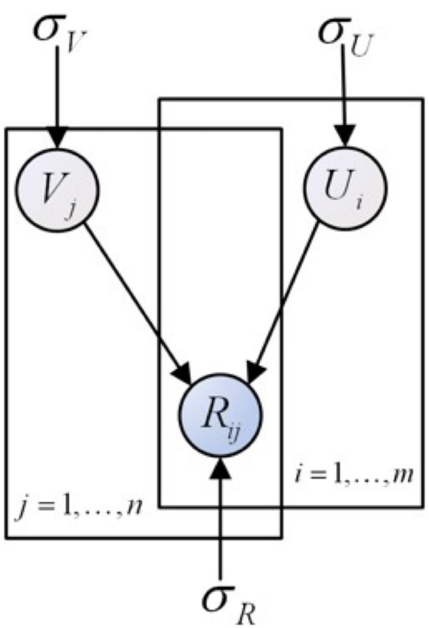
$$\frac{\partial e_{i,j}}{\partial V_{k,j}} = -(R_{i,j} - \sum_k U_{k,i}V_{k,j})U_{k,i} + \lambda_2 V_{k,j}$$

- **Update U and V accordingly**

$$U_{k,i} = U_{k,i} - \alpha \frac{\partial e_{i,j}}{\partial U_{k,i}} \qquad V_{k,j} = V_{k,j} - \alpha \frac{\partial e_{i,j}}{\partial V_{k,j}}$$

Probabilistic Matrix Factorization

- Assume zero-mean Gaussian priors on user and item feature:
- Define a conditional distribution over the observed ratings as:



$$p(U|\sigma_U^2) = \prod_{i=1}^{m} \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I})$$

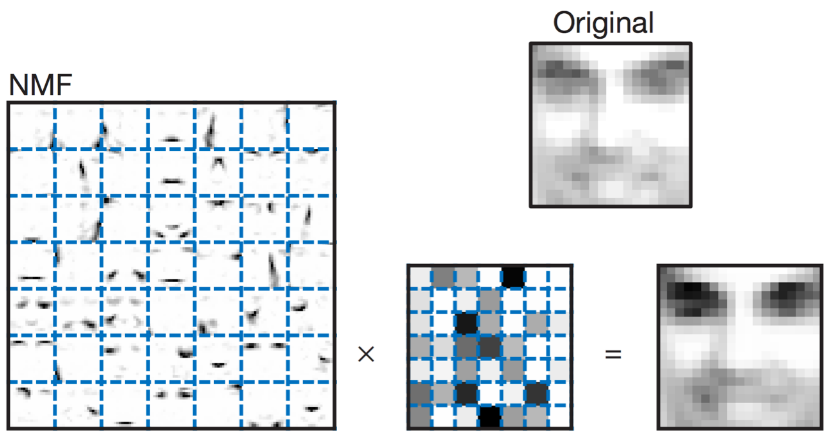
$$p(V|\sigma_V^2) = \prod_{j=1}^{n} \mathcal{N}(V_j|0, \sigma_V^2 \mathbf{I})$$

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

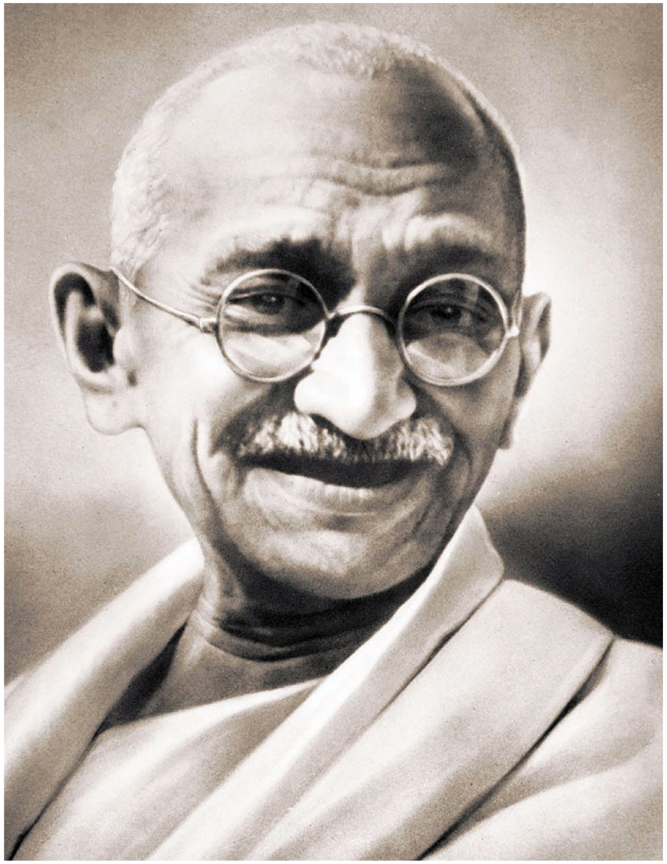
Non-negative Matrix Factorization

- **Given an observed matrix Y, to find two non-negative matrices U and V**

- **What is the advantage of the non-negative constraint?**
  - Intuition and interpretation (particularly for images)
  - No obvious advantages for recommendation



Social Recommendations



"Interdependence is and ought to be as much the ideal of man as self-sufficiency.
Man is a social being."

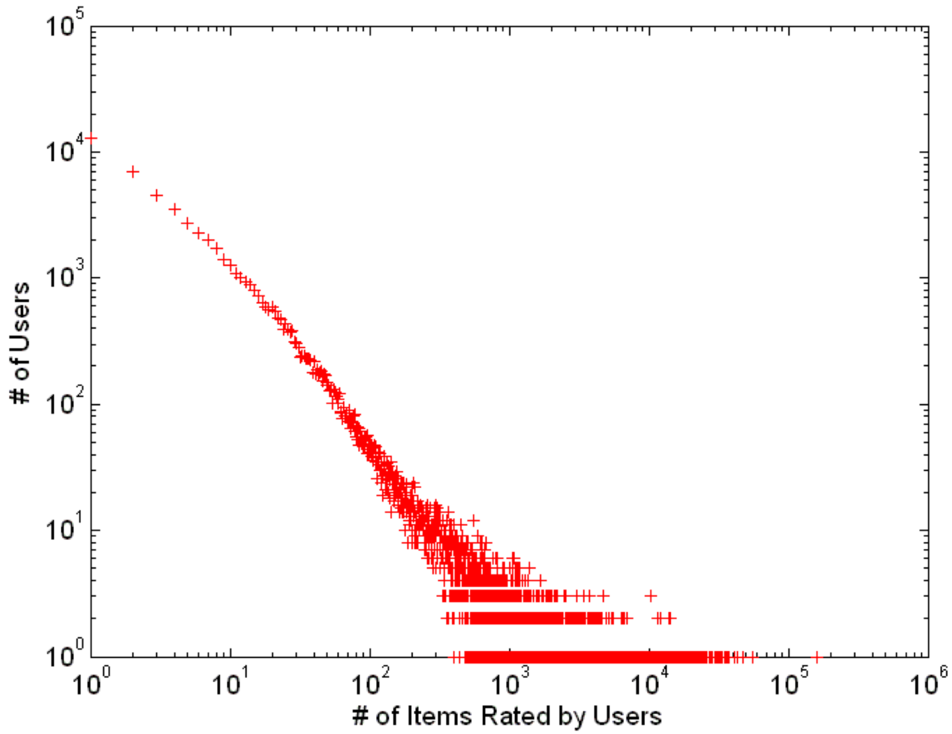
Challenges in Recommendation Systems

- **Data sparsity problems**

0	0	0	0	0	0	0	4	0	0
0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	5	0	0	0	0
0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	6	0	0	0	0
0	0	0	0	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3
0	0	8	0	0	0	0	0	0	0

Data Sparsity Problem

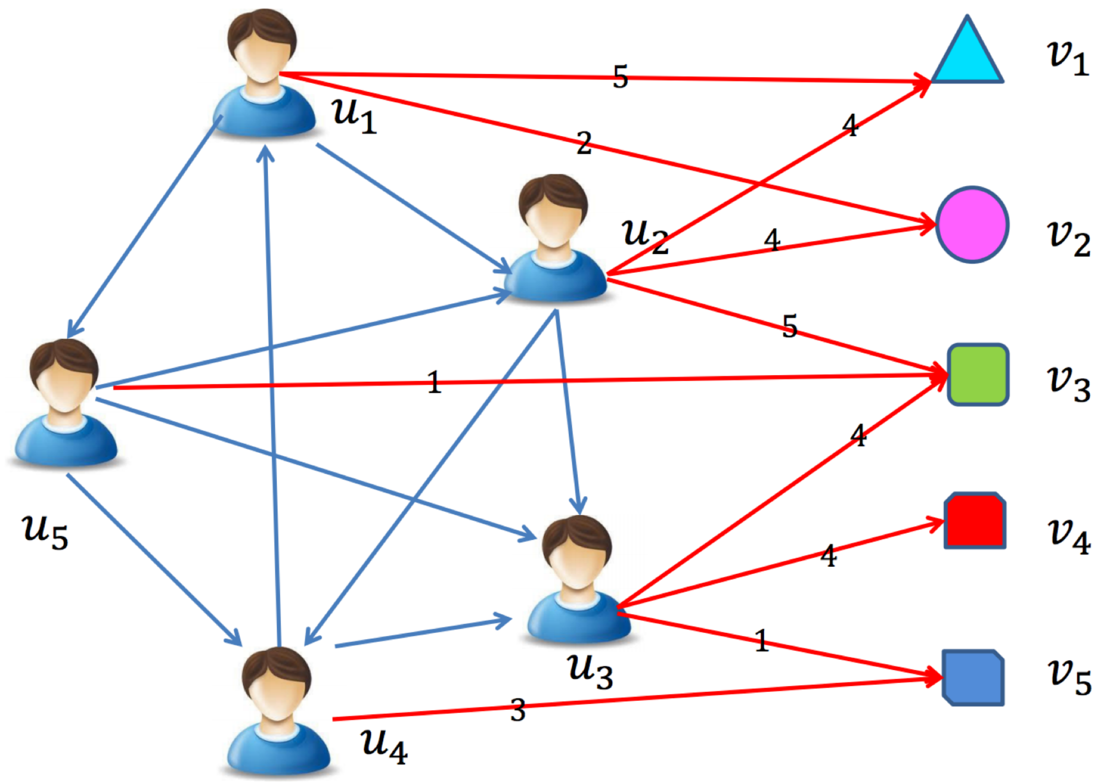
- **Epinions Dataset**
  - 114,222 users/ 754,987 items/ 13,385,713 ratings
  - Density: 0.0155%



eXascale Infolab

Social Recommendations

- **If links between users and items are sparse, can we use user social links for recommendation?**



Social Recommendations

Traditional Recommendation

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	4		3	3	3	1	
u_2		3		2			
u_3		2		5			1
u_4		5		1		3	
u_5	5		4		4		1
u_6	?		3		4	2	1

\+

Social Relationship

	u_1	u_2	u_3	u_4	u_5	u_6
u_1						1
u_2			1	1	1	
u_3		1				1
u_4		1			1	
u_5		1		1		
u_6	1		1			

Social Recommendations

- **Intuition: Users usually resort to their friends for recommendation. Thus, their preference might be similar to each other.**

a social recommendation CF model = a basic CF model + a social information model

Memory based Social Recommendation

- **Considering social information when computing user-user similarity**
  - Friend-based social relationship
    - Consider friends (bi-directional relationship) as correlated users

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	4		3	3	3	1	
u_2		3		2			
u_3		2		5			1
u_4		5		1		3	
u_5	5		4		4		1
u_6	?		3		4	2	1

Friendship

	u_1	u_2	u_3	u_4	u_5	u_6
u_1						1
u_2			1	1	1	
u_3		1				1
u_4		1			1	
u_5		1		1		
u_6	1		1			

Symmetric adjacency matrix

Memory based Social Recommendation

- **Considering social information when computing user-user similarity**
  - Trust-based social relationship
    - Consider trust (uni-directional relationship) between users
      - E.g. Twitter followers/followings

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	4		3	3	3	1	
u_2		3		2			
u_3		2		5			1
u_4		5		1		3	
u_5	5		4		4		1
u_6	?		3		4	2	1

Trust-based Relationship

	u_1	u_2	u_3	u_4	u_5	u_6
u_1		1			1	
u_2			1		1	
u_3						1
u_4		1			1	
u_5				1		1
u_6	1		1			

Asymmetric adjacency matrix

Memory based Social Recommendation

- **Considering social information when computing user-user similarity**
  - Trust-based social relationship
    - More complex trust modeling

Celebrity

Domain expert

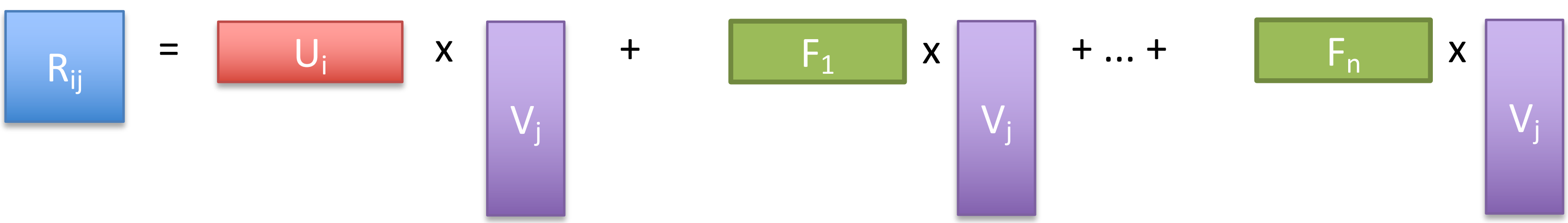
Authorities

Model-based Social Recommendation

- **Matrix factorization + Social relationship**
- **Three categories (depending on how social information is used)**
  - Ensemble methods
  - Regularization methods
  - Co-factorization methods

Ensemble Methods

- **Assumption:**
  - Users and their social networks should have similar ratings on items, and a missing rating for a given user is predicted as a linear combination of ratings from the user and her social network.

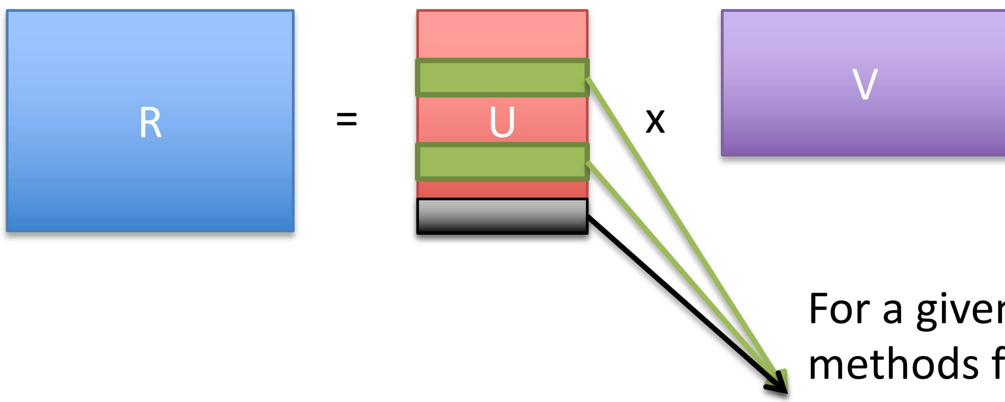


  - Same objective function with different formulation of R_{ij}

$$\hat{\mathbf{R}}_{ij} = \mathbf{u}_i^\top \mathbf{v}_j + \beta \sum_{u_k \in \mathcal{N}_i} \mathbf{S}_{ik} \mathbf{u}_k^\top \mathbf{v}_j$$

Regularization Methods

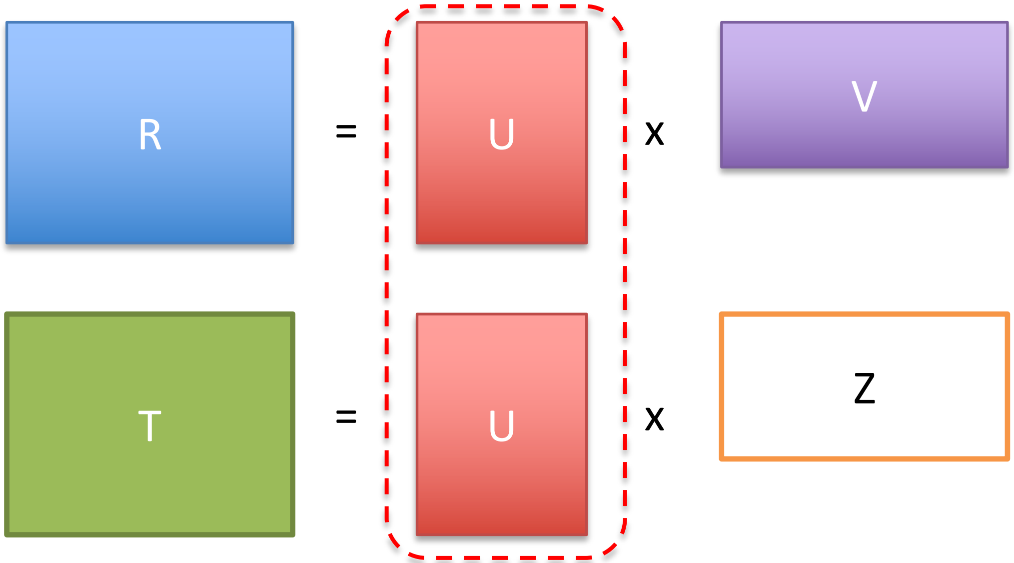
- **Assumption:**
  - A user's preference should be similar to that of her social network



For a given user u_i, regularization methods force her preference u_i to be closer to that of users in u_i's social network.

Co-factorization Methods

- **Assumption:**
  - The i-th user u_i should share the same user preference vector u_i in the rating space (rating information) and the social space (social information).

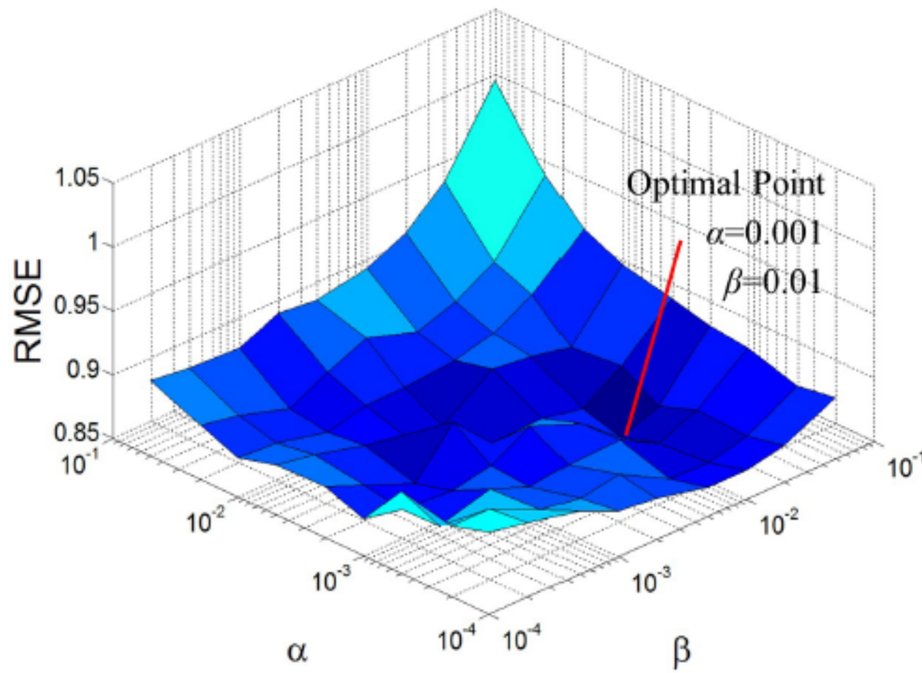


Shared factor

Co-factorization in the user-item matrix **R** and the user-user social relation matrix **T** by sharing the same user preference latent factor

Example of Performance

- **POI recommendation in Foursquare**
  - α and β control how much the social and item (POIs) influence the matrix factorization process, respectively.



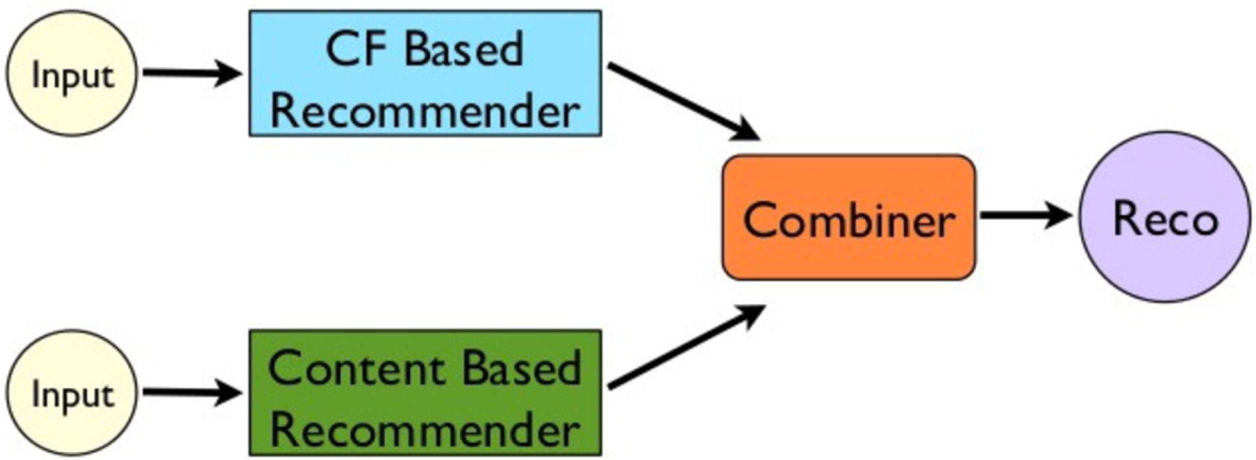
(a) New York restaurant dataset.

Advanced Topics on Recommendations

- Hybrid Recommendation
- Top-N (ranking-based) recommendation
- Context in recommendation
- Privacy

Hybrid Recommendation

- **Merging the results from content-based recommendation and collaborative filtering**



Top-N (ranking-based) Recommendation

- **Rating prediction:**
  - Predict individual ratings of users on items as accurate as possible
- **Ranking prediction:**
  - Predict ranking of items that a user may be interested in.

Real rating from u_1

	i_1	i_2	i_3
u_1	1	2	5

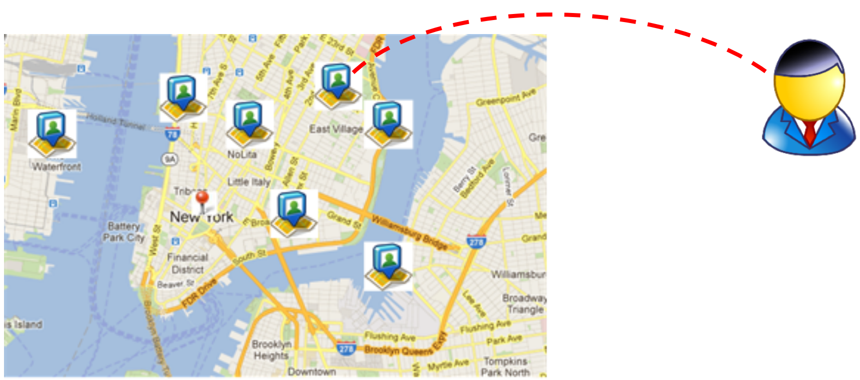
	i_1	i_2	i_3
Prediction 1	3	4	5

Same MAE, but different ranking loss

	i_1	i_2	i_3
Prediction 2	1	4	3

Context in recommendation

- **Spatiotemporal context**
  - More constraints
    - Geographical distance
    - Temporal suitability (e.g., opening hour)
  - Spatiotemporal user preference modeling



It's Friday 20:00 pm, Jim is at [40.71448, -74.00598], which places should we recommend to him?

Privacy issues

- **Releasing a user's rating information may cause serious privacy leakage.**
  - Inference attacks on identity, gender, income level, politic views, etc.
- **Privacy-preserving data publishing**
  - Add a certain level of noise to user rating such that the privacy is protected while the noise-added ratings can still be used for recommendation

Conclusion

- **Why do we need recommendation systems?**
  - Information overload -> Searching pertinent information is hard!
- **Recommendation Techniques**
  - Content-based recommendation
  - Collaborative-based recommendation
    - Memory-based methods
    - Model-based methods
- **Social Recommendation**
  - Social relationship in recommendation
- **Advanced Topics in Recommendation**