# Introduction to Social Computing
# Social Recommendation

Irwin King

ATT Labs, Research
&
Department of Computer Science and Engineering
The Chinese University of Hong Kong

king@cse.cuhk.edu.hk
http://www.cse.cuhk.edu.hk/~king

# Outline

- Social Search Engine
- Social Recommender Systems
- Social Media Analysis

# Macro Definition

- Search in
  - Shared bookmarks
  - Collaborative directories
  - Collaborative news/opinions
  - Social Q&A sites
  - etc...

# Micro Definition

## Leveraging Your Social Networks for Searching

# Leveraging All Kinds of Web Accounts

# Google's Social Search

Results from people in your social circle for **google bus**

Google Maps Ad on Chicago **Bus** - Googlified
haochi - connected via Tom on digg.com
**google** transit chicago **bus** ad. **Google** Transit recently became available to Chicago users and the Chicago team has been very active in **...**
googlified.com/**google**-maps-ad-on-chicago-**bus**/
More results from haochi »

**Google** Student Blog: The **Google** Apps **Bus** stops at the beginning
Google Students - connected via twitter.com
Almost two years later, the **Google** App to School **bus** pulled into Arizona State University and met with over a thousand students, faculty, and staff using **...**
**google**forstudents.blogspot.com/2008/09/**google**-apps-**bus**-stops-at-beginning.html
More results from Google Students »

Searches related to: **google bus**

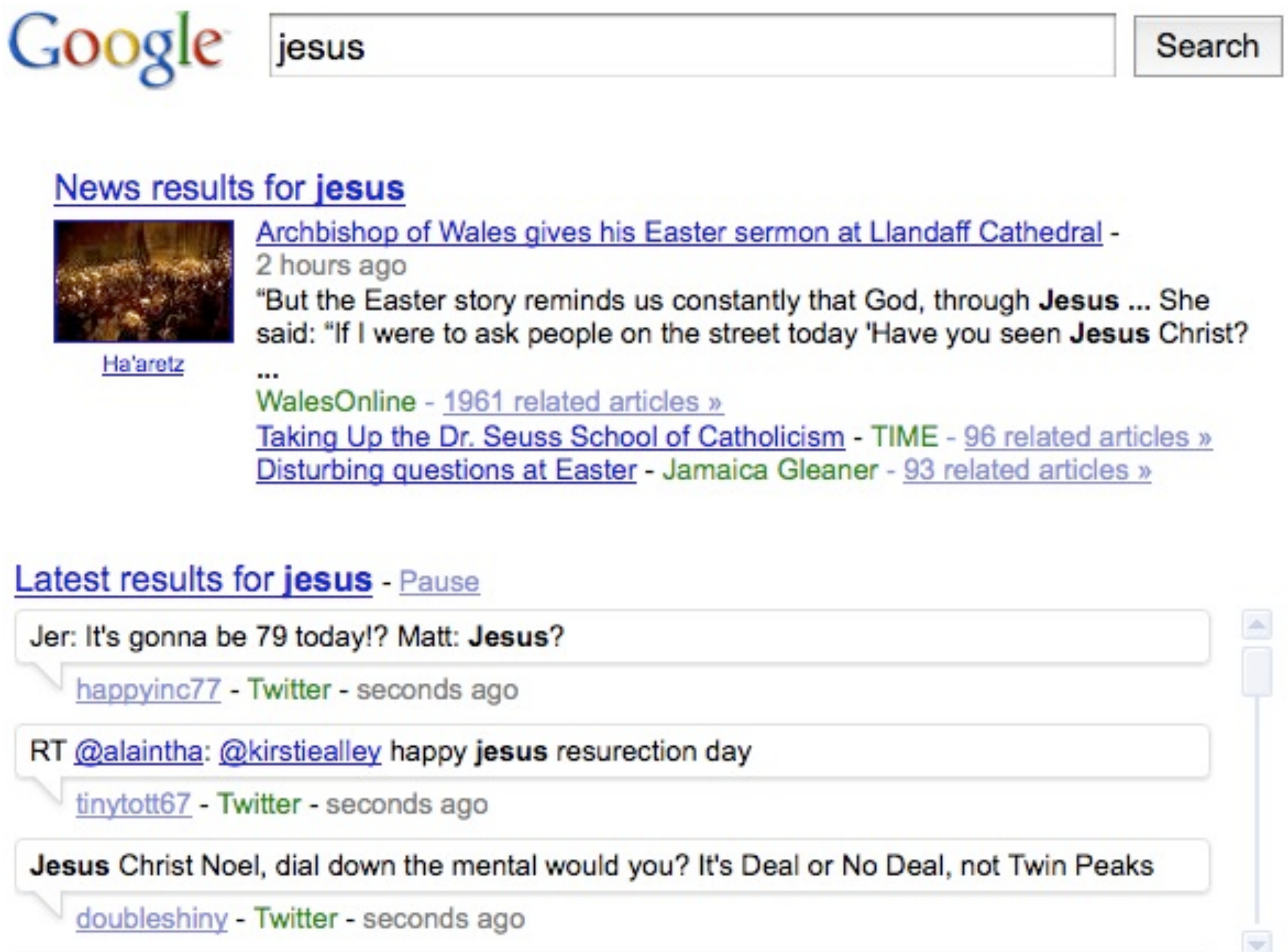
**tamil nadu** bus | google **apps** bus | google bus **routes** | google bus **transit**

Results from your social circle for **seattle** - BETA - My social circle - My social content
1078 photos - 17 contacts - Last photo 3 months ago

Results from people in your social circle for **san francisco international airport hotel** - BETA - My social circle - My social content

**San Francisco Airport Hotel** Burlingame California
Crowne Plaza SFO - connected via twitter.com
Our Burlingame **hotel** is only 1.5 miles south of **San Francisco International Airport** on the San Francisco Bay close to an array of exciting attractions. **...**
www.sfocp.com/
More results from Crowne Plaza SFO »

# Google's Social Search



# Aardvark

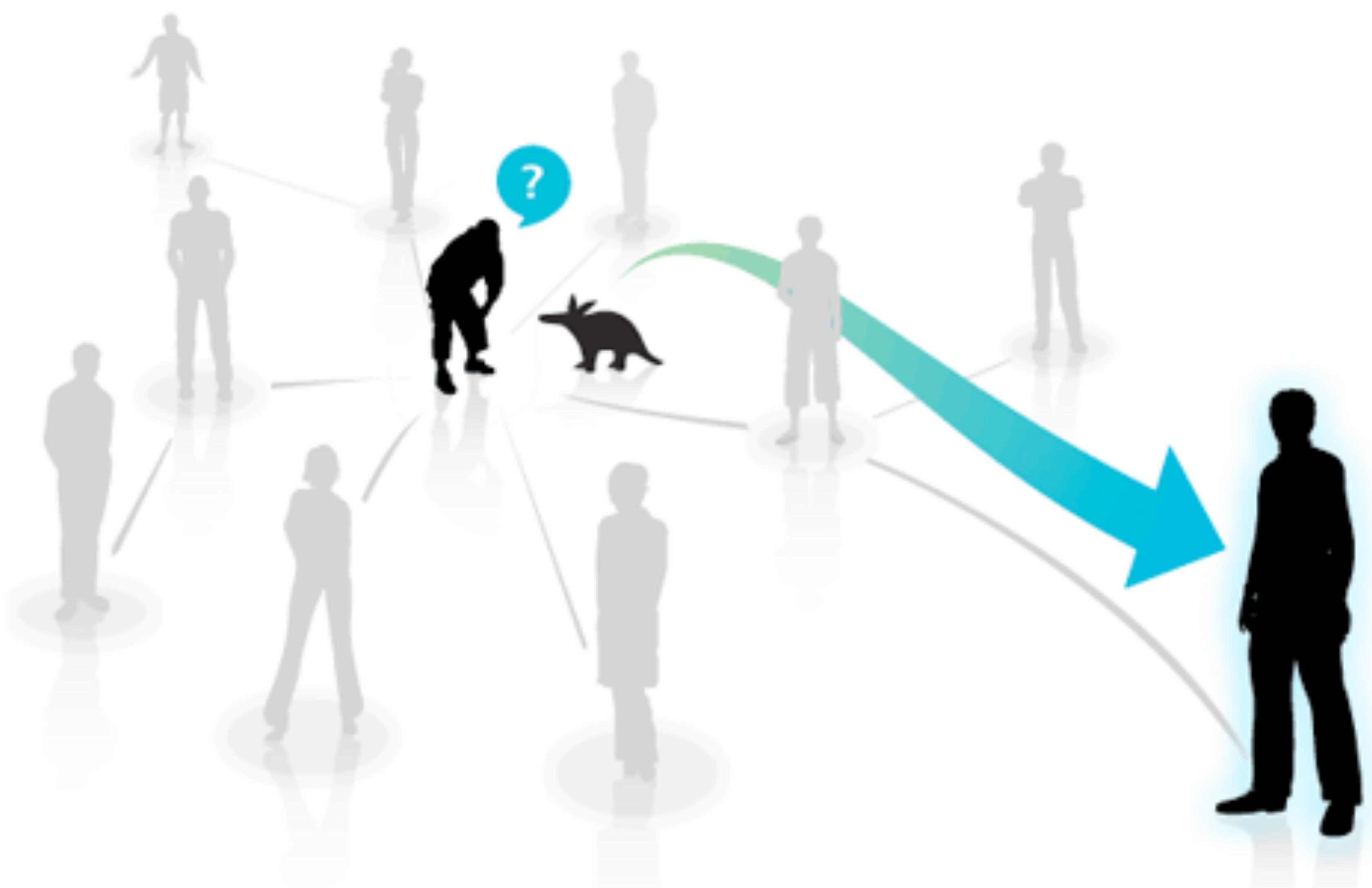

Ask a question and I'll find someone to answer
What's the best...
Example questions
Ask someone
1. Send Aardvark a question
2. Aardvark finds the perfect person to answer
3. Get their response in a few minutes
What's a great biking path around Golden Gate Park?
Sent 11:16 AM PDT
My favorite is a secret trail that takes you to the beach...
Sent 11:20 AM PDT

# Evolution of Search

- Question
  - Contents
  - Machine Intelligence (Dialog systems)
  - People
  - Friends
  - Hybrid

# The Anatomy of A Large-Scale Social Search Engine

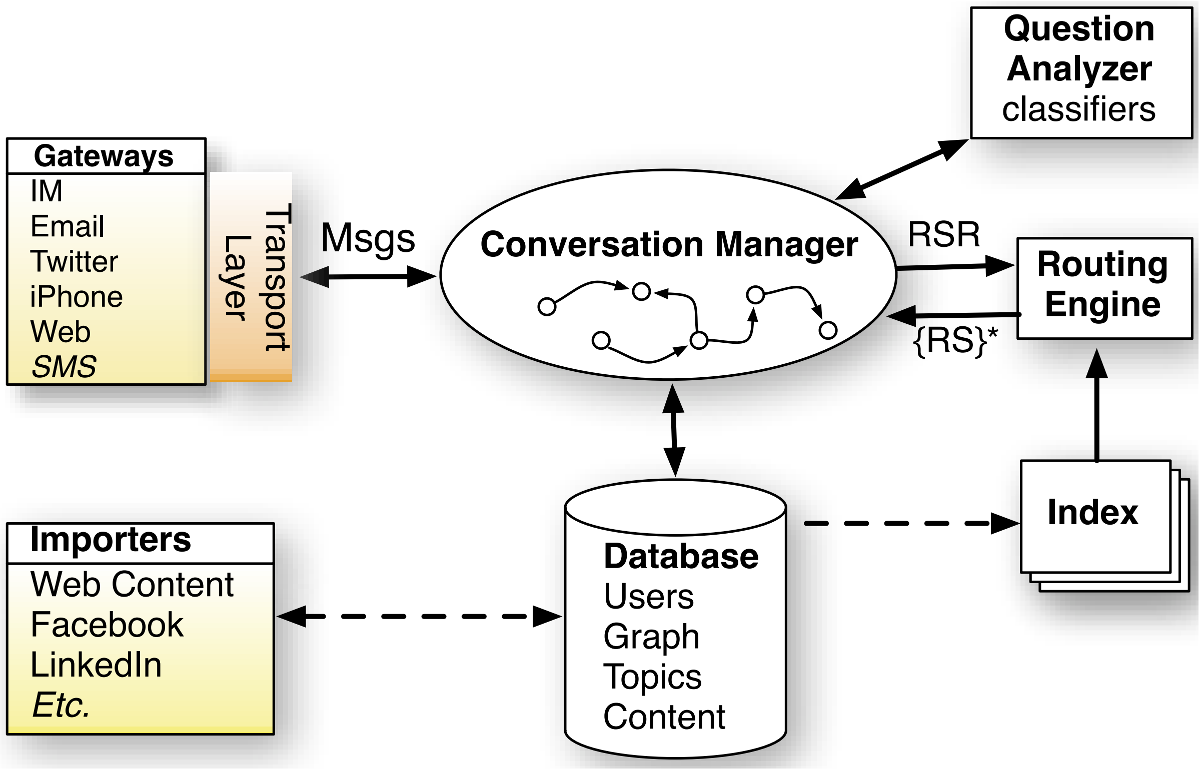
[D. Horowitz et al., WWW2010]



# The Anatomy of A Large-Scale Social Search Engine

[D. Horowitz et al., WWW2010]

- Main components
  - Crawler and Indexer
  - Query Analyzer
  - Ranking Function
  - UI



# The Anatomy of A Large-Scale Social Search Engine

[D. Horowitz et al., WWW2010]

- The model
  - With the topics *T*, the probability that user *i* will successfully answer question *q* is defined as

$$p(u_i|q) = \sum_{t \in T} p(u_i|t)p(t|q)$$

  - Given a question *q* from user *j*, return a ranked list of user *i* that maximizes $s(u_i, u_j, q)$

$$s(u_i, u_j, q) = p(u_i|u_j) \cdot p(u_i|q) = p(u_i|u_j) \sum_{t \in T} p(u_i|t)p(t|q)$$

# The Anatomy of A Large-Scale Social Search Engine

[D. Horowitz et al., WWW2010]

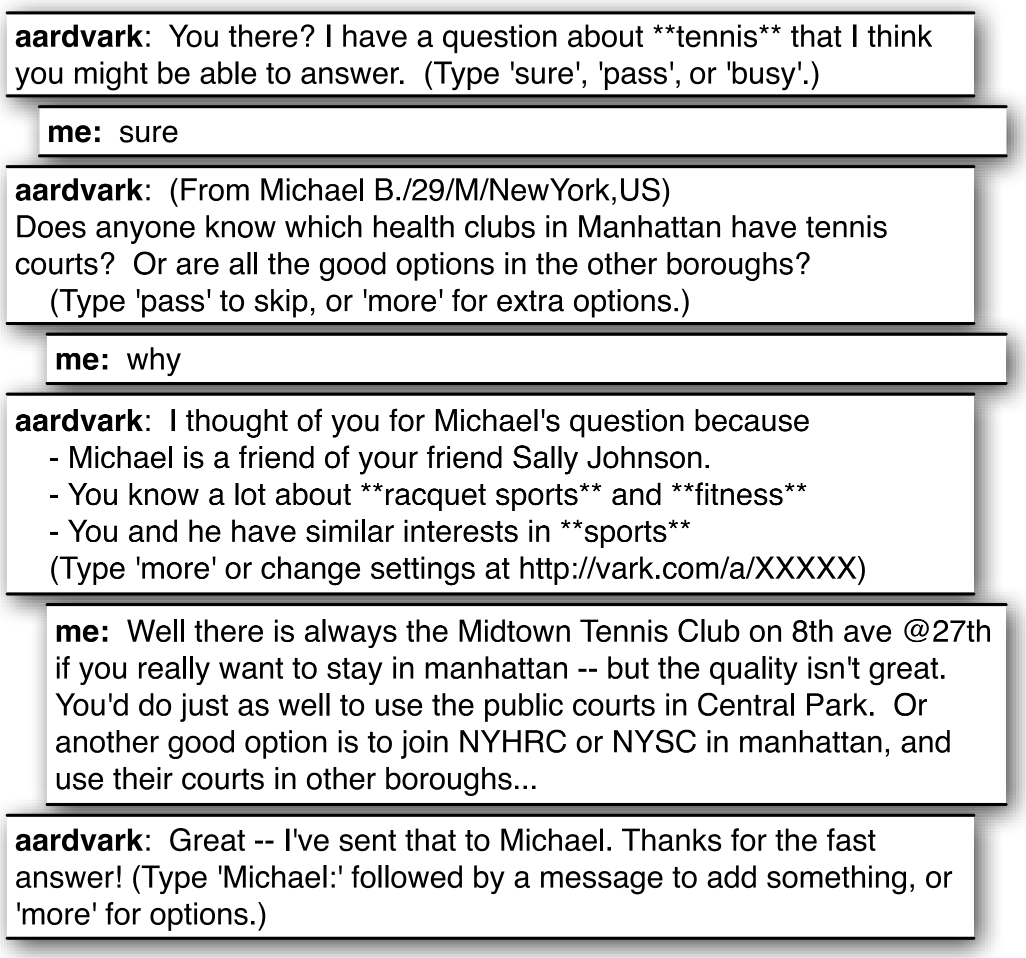



Figure 3: Example of Aardvark interacting with an answerer

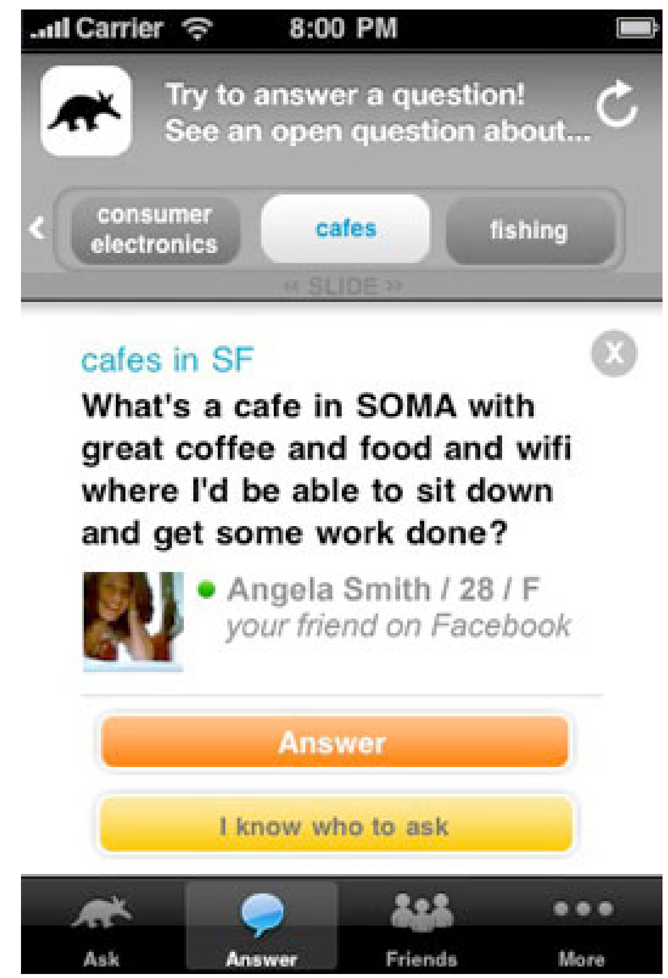



Figure 4: Screenshot of Aardvark Answering Tab on iPhone

# The Anatomy of A Large-Scale Social Search Engine

[D. Horowitz et al., WWW2010]

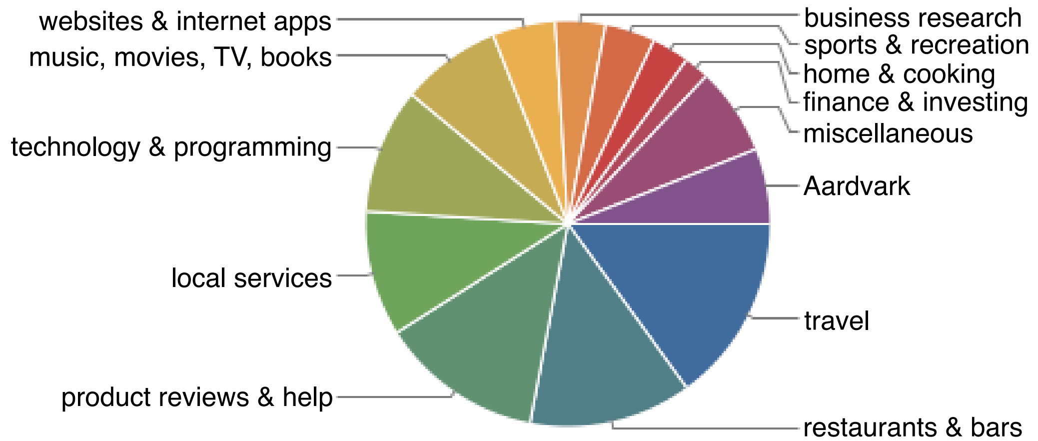



Figure 8: Categories of questions sent to Aardvark

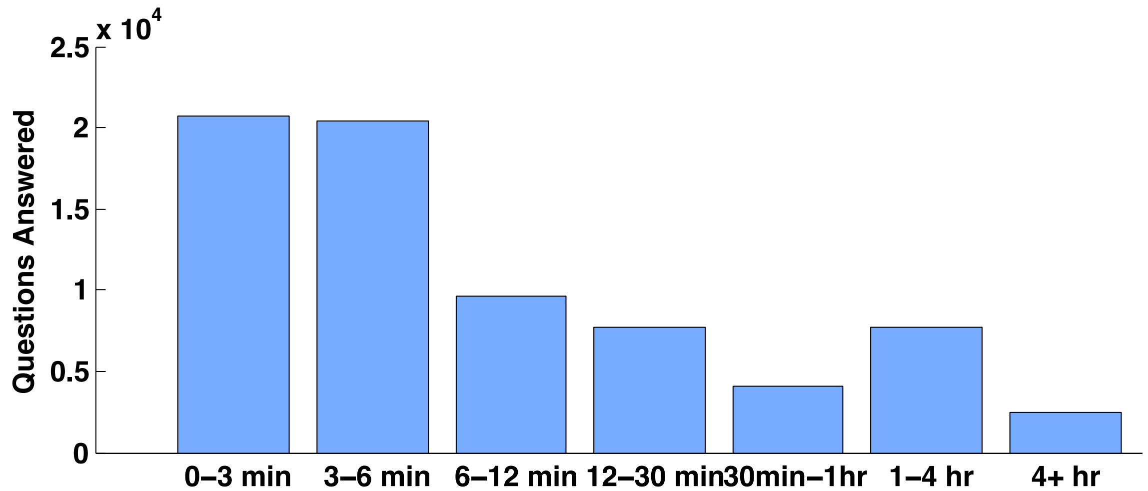



Figure 9: Distribution of questions and answering times.

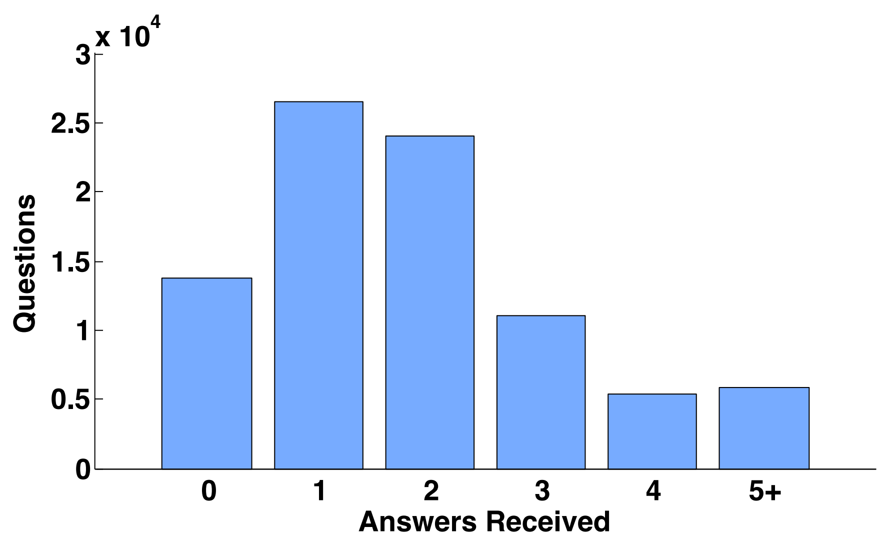



Figure 10: Distribution of questions and number of answers received.

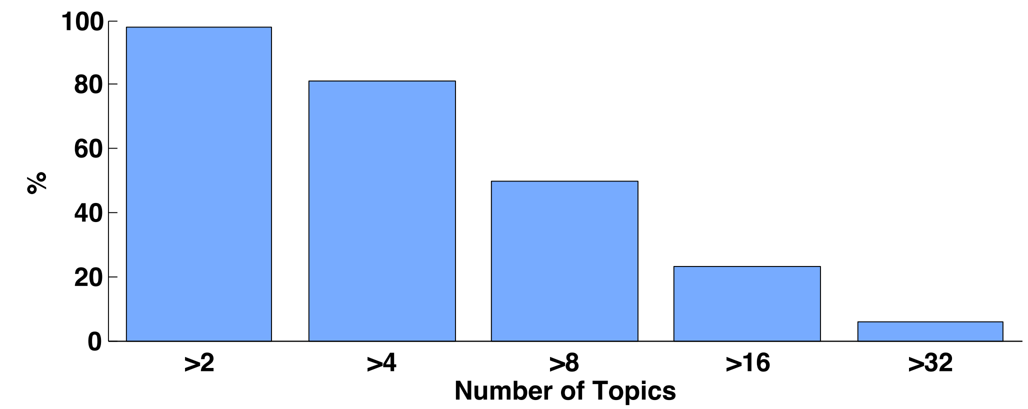



Figure 11: Distribution of percentage of users and number of topics

# References


- Introducing Google Social Search: I finally found my friend's New York blog! http://googleblog.blogspot.com/2009/10/introducing-google-social-search-i.html
- Search Is Getting More Social. http://googleblog.blogspot.com/2010/01/search-is-getting-more-social.html
- D. Horowitz, S. D. Kamvar. The Anatomy of a Large Scale Social Search Engine. WWW, 2010

# Outline

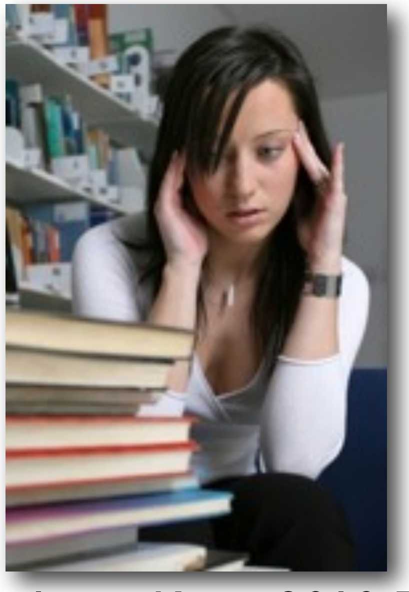
- Social Search Engine
- Social Recommender Systems
- Social Media Analysis

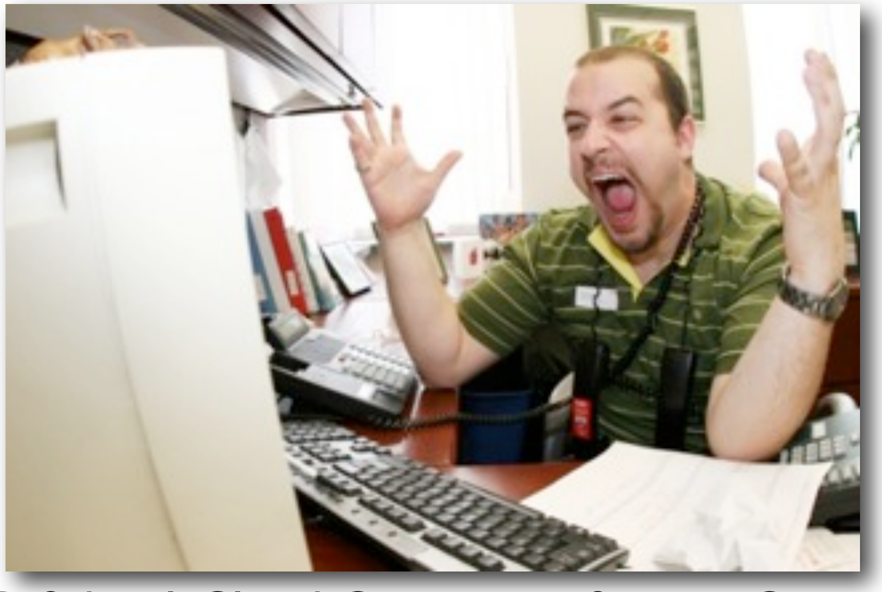
# Social Recommender Systems

- Introduction
- Collaborative Filtering
- Trust-aware Recommender Systems
- Social-based Recommender Systems

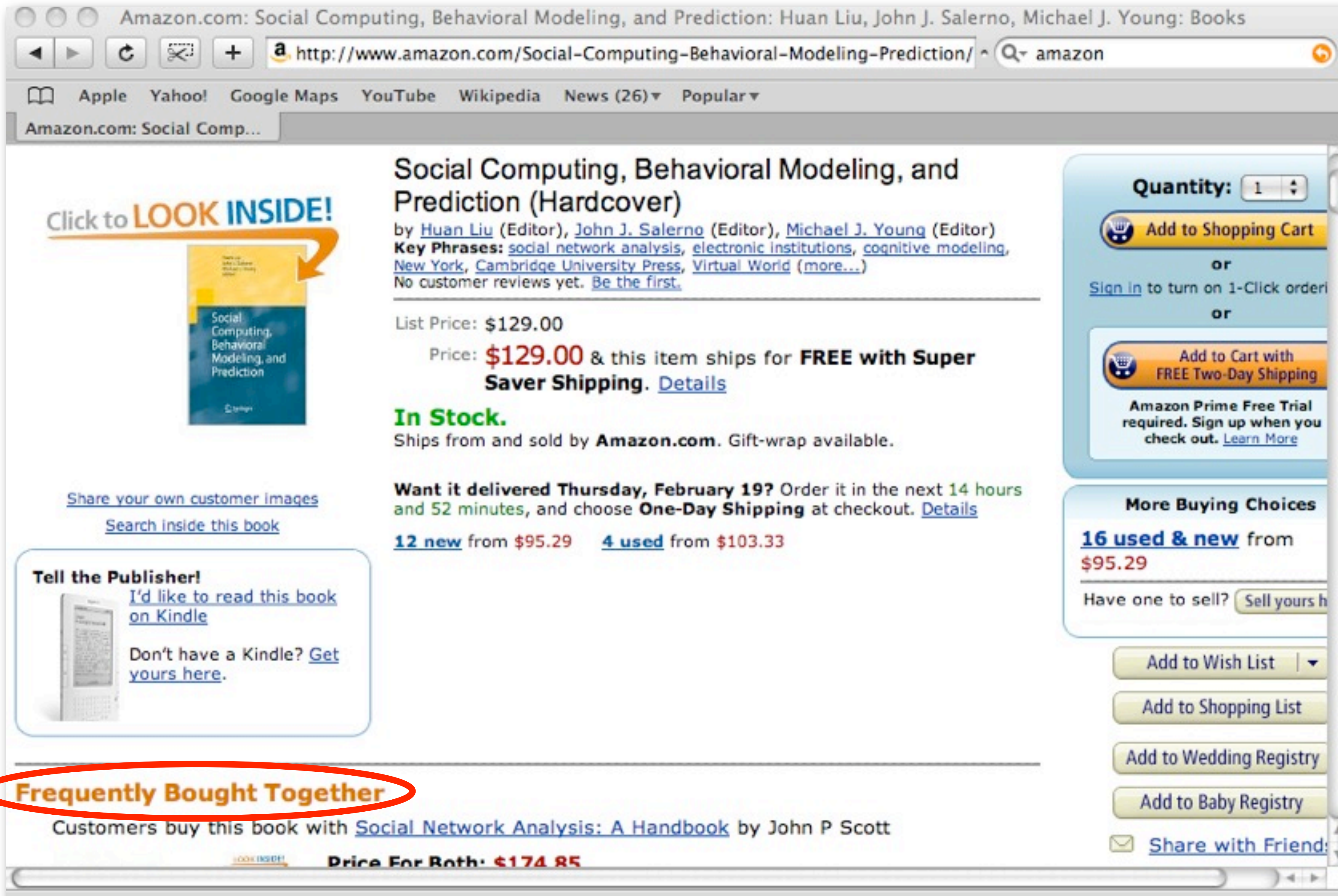
# How Much Information Is on the Web?

# Information Overload

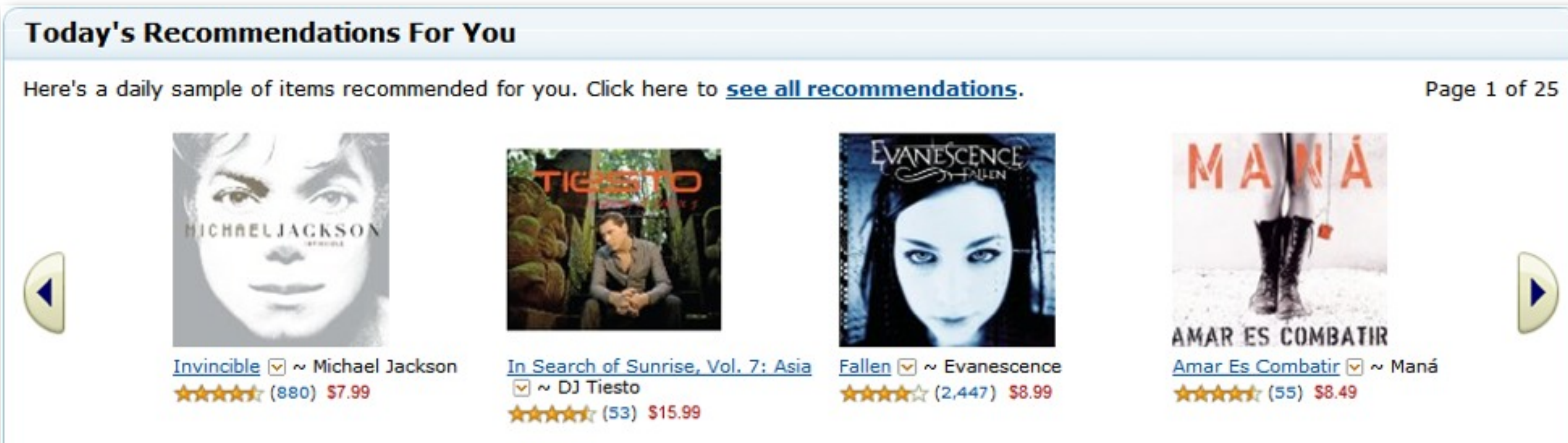


# Real Life Examples



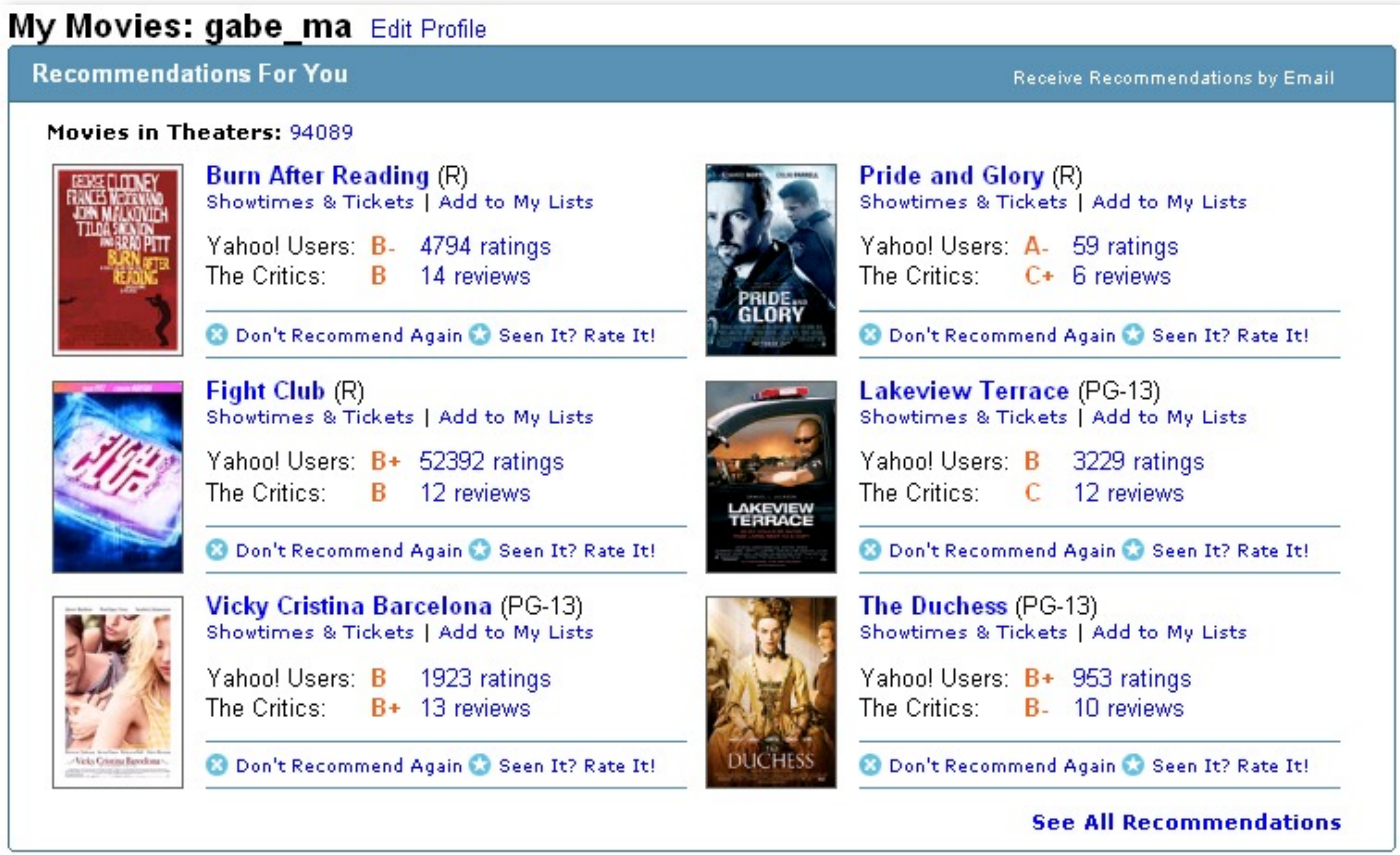
# Real Life Examples

# Real Life Examples

# Real Life Examples

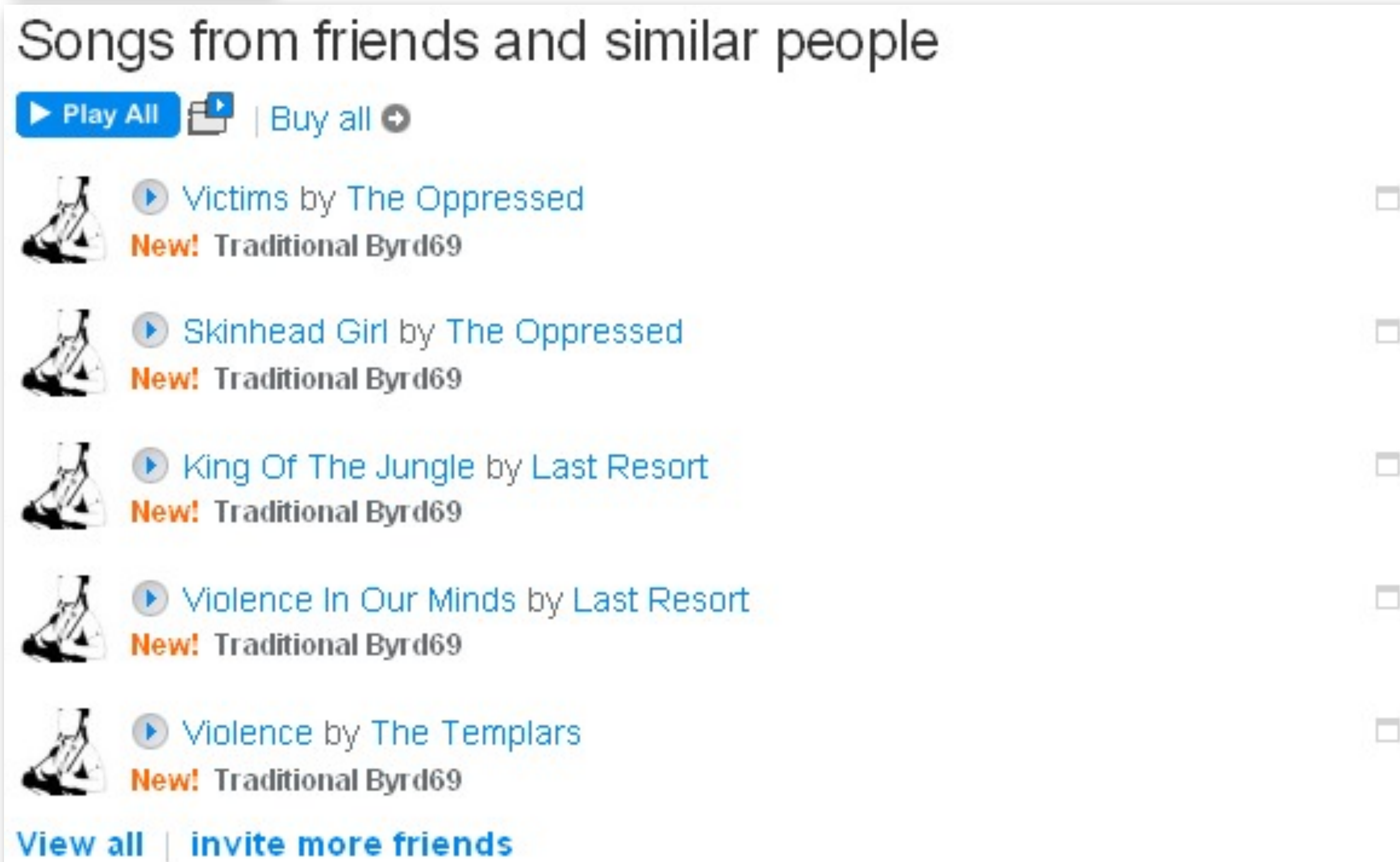
YAHOO! MOVIES



iTunes 8

# Real Life Examples

# Real Life Examples



# Basic Approaches

- 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

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

# Framework

Items

Users

| | $i_1$ | $i_2$ | | | | $i_j$ | | | | | | | $i_m$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $u_1$ | | | | | | | | | | | | | |
| $u_2$ | 1 | 3 | | 4 | | 2 | | 5 | | | 3 | 4 | |
| | | | | | | | | | | | | | |
| $u_i$ | | 3 | | 4 | | $r_{ij}$ | 3 | 4 | | 3 | 4 | | 4 |
| | | | | | | | | | | | | | |
| $u_n$ | 1 | | | 3 | 5 | 2 | | 4 | 1 | | | 3 | |

- The tasks
  - Find the unknown rating?
  - Which item should be recommended?

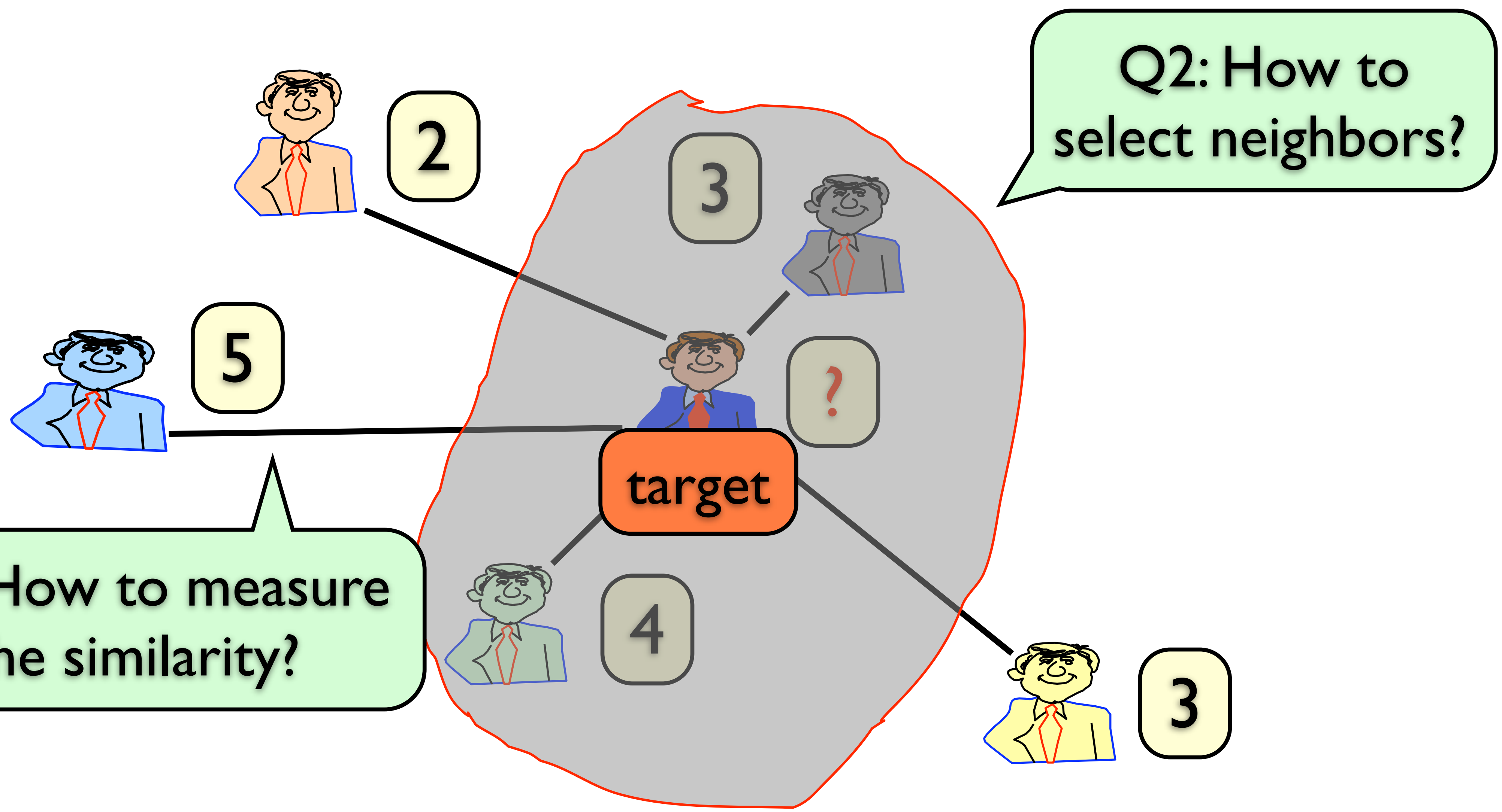
# Social Recommender Systems

- Introduction
- Collaborative Filtering
- Trust-aware Recommender Systems
- Social-based Recommender Systems

# Collaborative Filtering

- Memory-based (Neighborhood-based)
  - User-based
  - Item-based
- Model-based
  - Clustering Methods
  - Bayesian Methods
  - Matrix Factorization
  - etc.

# User-User Similarity



# User-based Collaborative Filtering

Items

Users

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $u_1$ | | | | | | | | | | | | | |
| $u_2$ | 1 | 3 | | 4 | | 2 | | 5 | | | 3 | 4 | |
| $u_3$ | | | | | | | | | | | | | |
| $u_4$ | | 3 | | 4 | | | 3 | 4 | | 3 | 4 | | 4 |
| $u_5$ | | | | | | | | | | | | | |
| $u_6$ | 1 | | | 3 | 5 | 2 | | 4 | 1 | | | 3 | |

# User-based Collaborative Filtering

Items

Users

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $u_1$ | | | | | | | | | | | | | |
| $u_2$ | 1 | 3 | | 4 | | 2 | | 5 | | | 3 | 4 | |
| $u_3$ | | | | | | | | | | | | | |
| $u_4$ | | 3 | | 4 | | | 3 | 4 | | 3 | 4 | | 4 |
| $u_5$ | | | | | | | | | | | | | |
| $u_6$ | 1 | | | 3 | 5 | 2 | | 4 | 1 | | | 3 | |

# User-based Collaborative Filtering

Items

| Users | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $u_1$ | | | | | | | | | | | | | |
| $u_2$ | 1 | 3 | | 4 | | 2 | | 5 | | | 3 | 4 | |
| $u_3$ | | | | | | | | | | | | | |
| $u_4$ | | 3 | | 4 | | | 3 | 4 | | 3 | 4 | | 4 |
| $u_5$ | | | | | | | | | | | | | |
| $u_6$ | 1 | | | 3 | 5 | 2 | | 4 | 1 | | | 3 | |

# User-based Collaborative Filtering

Items

Users

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $u_1$ | | | | | | | | | | | | | |
| $u_2$ | 1 | 3 | | 4 | | 2 | | 5 | | | 3 | 4 | |
| $u_3$ | | | | | | | | | | | | | |
| $u_4$ | | 3 | | 4 | | | 3 | 4 | | 3 | 4 | | 4 |
| $u_5$ | | | | | | | | | | | | | |
| $u_6$ | 1 | | | 3 | 5 | 2 | | 4 | 1 | | | 3 | |

# User-based Collaborative Filtering

Items

Users

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $u_1$ | | | | | | | | | | | | | |
| $u_2$ | 1 | 3 | | 4 | | 2 | | 5 | | | 3 | 4 | |
| $u_3$ | | | | | | | | | | | | | |
| $u_4$ | | 3 | | 4 | | | 3 | 4 | | 3 | 4 | | 4 |
| $u_5$ | | | | | | | | | | | | | |
| $u_6$ | 1 | | | 3 | 5 | 2 | | 4 | 1 | | | 3 | |

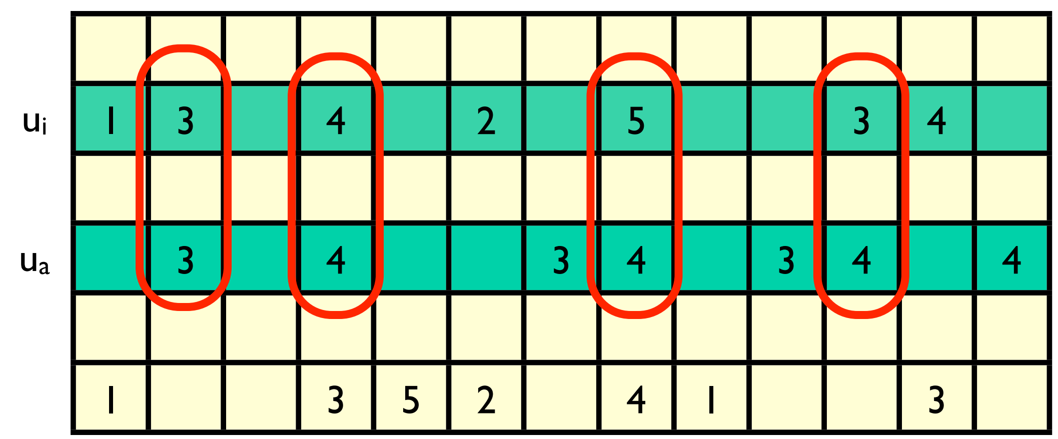
# User-based Collaborative Filtering

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

$$w(a,i) = \frac{\sum_j (r_{aj} - \bar{r}_a)(r_{ij} - \bar{r}_i)}{\sqrt{\sum_j (r_{aj} - \bar{r}_a)^2 \sum_j (r_{ij} - \bar{r}_i)^2}} \quad j \in I(a) \cap I(i)$$

  - Cosine measure

$$c(a,i) = \frac{r_a \cdot r_i}{||r_a||_2 * ||r_i||_2}$$



# Collaborative Filtering

- Memory-based (Neighborhood-based)
  - User-based
  - Item-based
- Model-based
  - Clustering Methods
  - Bayesian Methods
  - Matrix Factorization
  - etc.

# Item-Item Similarity

- Search for similarities among items
- Item-Item similarity is more stable than user-user similarity

# Correlation-based Methods

[Sarwar, 2001]

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

$$s_{ij} = \frac{\sum_u (r_{uj} - \bar{r}_j)(r_{ui} - \bar{r}_i)}{\sqrt{\sum_u (r_{uj} - \bar{r}_j)^2 \sum_u (r_{ui} - \bar{r}_i)^2}}$$

  - u: users rated both items

| | $i_1$ | $i_2$ | | | | $i_i$ | | $i_j$ | | | | | $i_m$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $u_1$ | | | | | | | | | | | | | |
| $u_2$ | 1 | 3 | | 4 | | 2 | | 5 | | | 3 | 4 | |
| | | | | | | | | | | | | | |
| $u_i$ | | 3 | | 4 | | | 3 | 4 | | 3 | 4 | | 4 |
| | | | | | | | | | | | | | |
| $u_n$ | 1 | | | 3 | 5 | 2 | | 4 | 1 | | | 3 | |

# Collaborative Filtering

- Memory-based (Neighborhood-based)
  - User-based
  - Item-based
- Model-based
  - Clustering Methods
  - Bayesian Methods
  - Matrix Factorization
  - etc...

# 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.

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

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

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

Regularization terms

where $\lambda_1, \lambda_2 > 0$ .

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

# Regularized Matrix Factorization

- Algorithm for RMF
  - Not convex & local optimal
  - Gradient-decent algorithm
    - Gradient computation with randomly initialized *U* and *V*

$$\frac{\partial L}{\partial u_{il}} = \lambda u_{il} - \sum_{j|(i,j)\in S} (y_{ij} - \widehat{y_{ij}}) v_{jl}$$

$$\frac{\partial L}{\partial v_{il}} = \lambda v_{il} - \sum_{j|(i,j)\in S} (y_{ij} - \widehat{y_{ij}}) u_{jl}$$

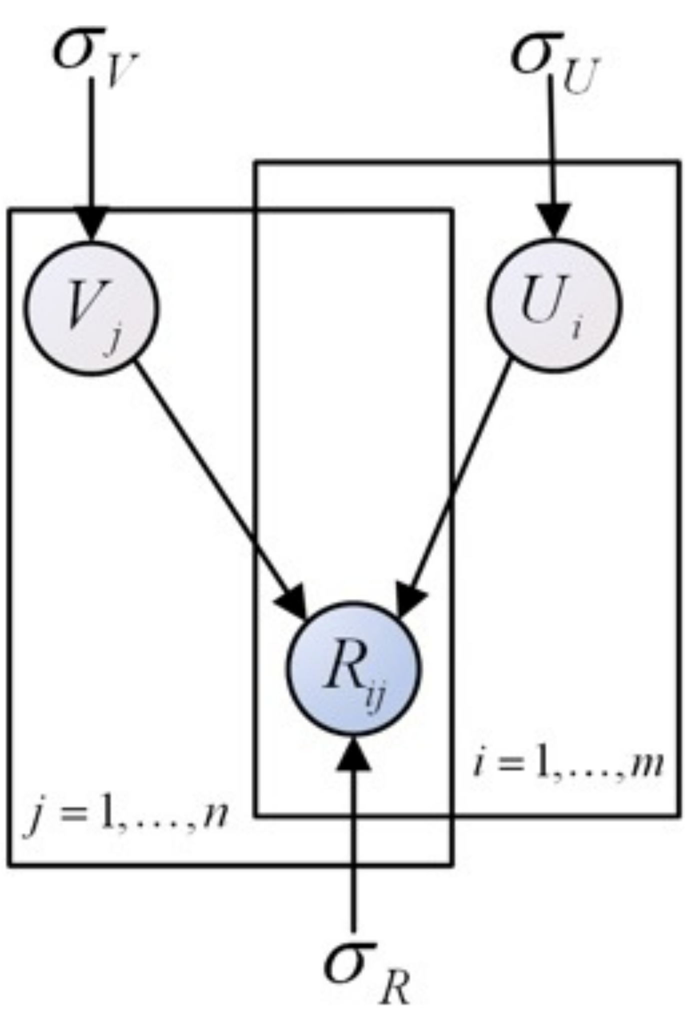
    - Update *U* and *V* alternatively

$$u_{il}^{(t+1)} = u_{il}^{(t)} - \tau \frac{\partial L}{\partial u_{il}^{(t)}}$$

$$v_{jl}^{(t+1)} = v_{jl}^{(t)} - \tau \frac{\partial L}{\partial v_{jl}^{(t)}}$$

# Probabilistic Matrix Factorization

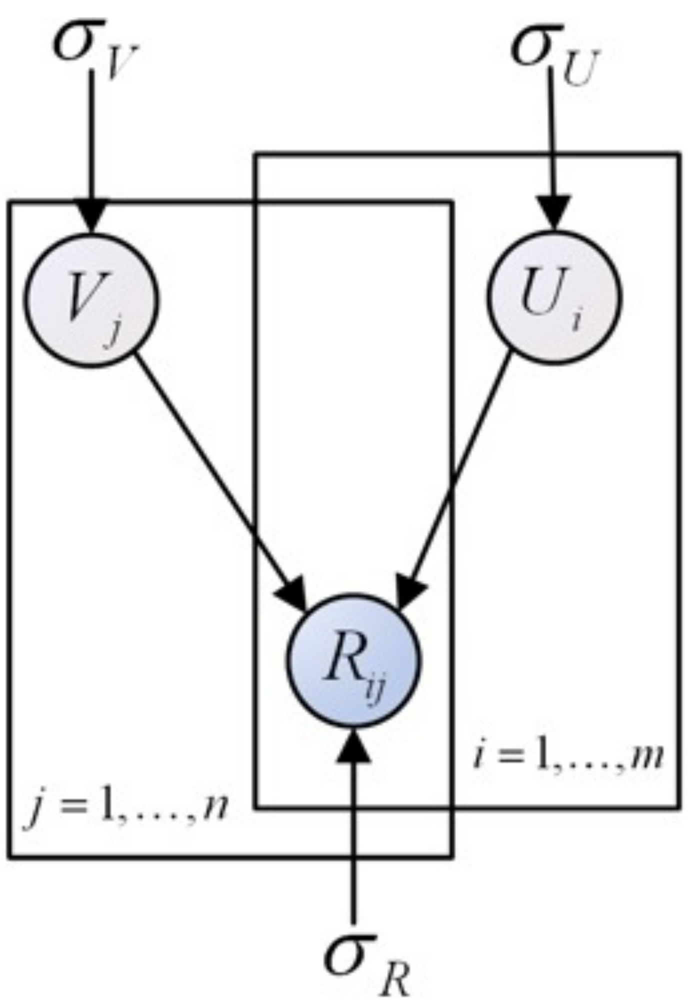
- PMF
  - Define a conditional distribution over the observed ratings as:




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

# Probabilistic Matrix Factorization

- PMF
  - Assume zero-mean spherical Gaussian priors on user and item feature:

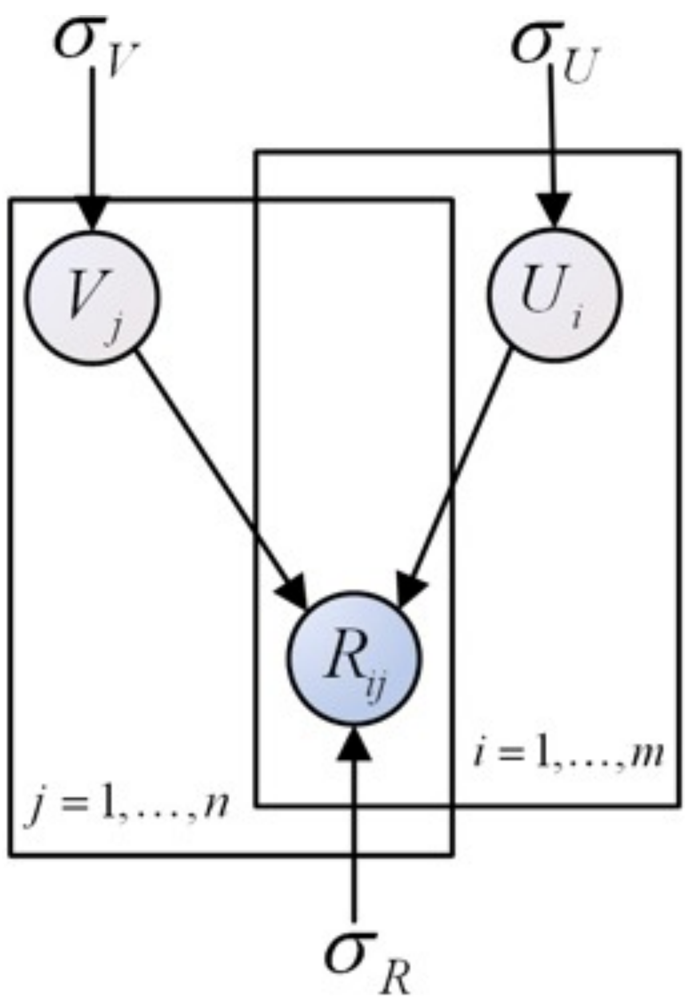



$$p(U|\sigma_U^2) = \prod_{i=1}^{m} \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I})$$

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

# Probabilistic Matrix Factorization

- PMF
  - Bayesian inference




$$p(U,V|R,\sigma_R^2,\sigma_U^2,\sigma_V^2) \propto p(R|U,V,\sigma_R^2)p(U|\sigma_U^2)p(V|\sigma_V^2)$$
$$= \prod_{i=1}^{m}\prod_{j=1}^{n}\left[\mathcal{N}\left(R_{ij}|g(U_i^T V_j),\sigma_R^2\right)\right]^{I_{ij}^R}$$
$$\times \prod_{i=1}^{m}\mathcal{N}(U_i|0,\sigma_U^2\mathbf{I}) \times \prod_{j=1}^{n}\mathcal{N}(V_j|0,\sigma_V^2\mathbf{I}).$$

# Non-negative Matrix Factorization

- NMF
  - Given an observed matrix Y, to find two non-negative matrices U and V
  - Two types of loss functions
    - Squared error function

$$\sum_{ij} (R_{ij} - U_i^T V_j)^2$$

    - Divergence

$$D(R||U^T V) = \sum_{ij} (R_{ij} \log \frac{R_{ij}}{U_i^T V_j} - R_{ij} + U_i^T V_j)$$

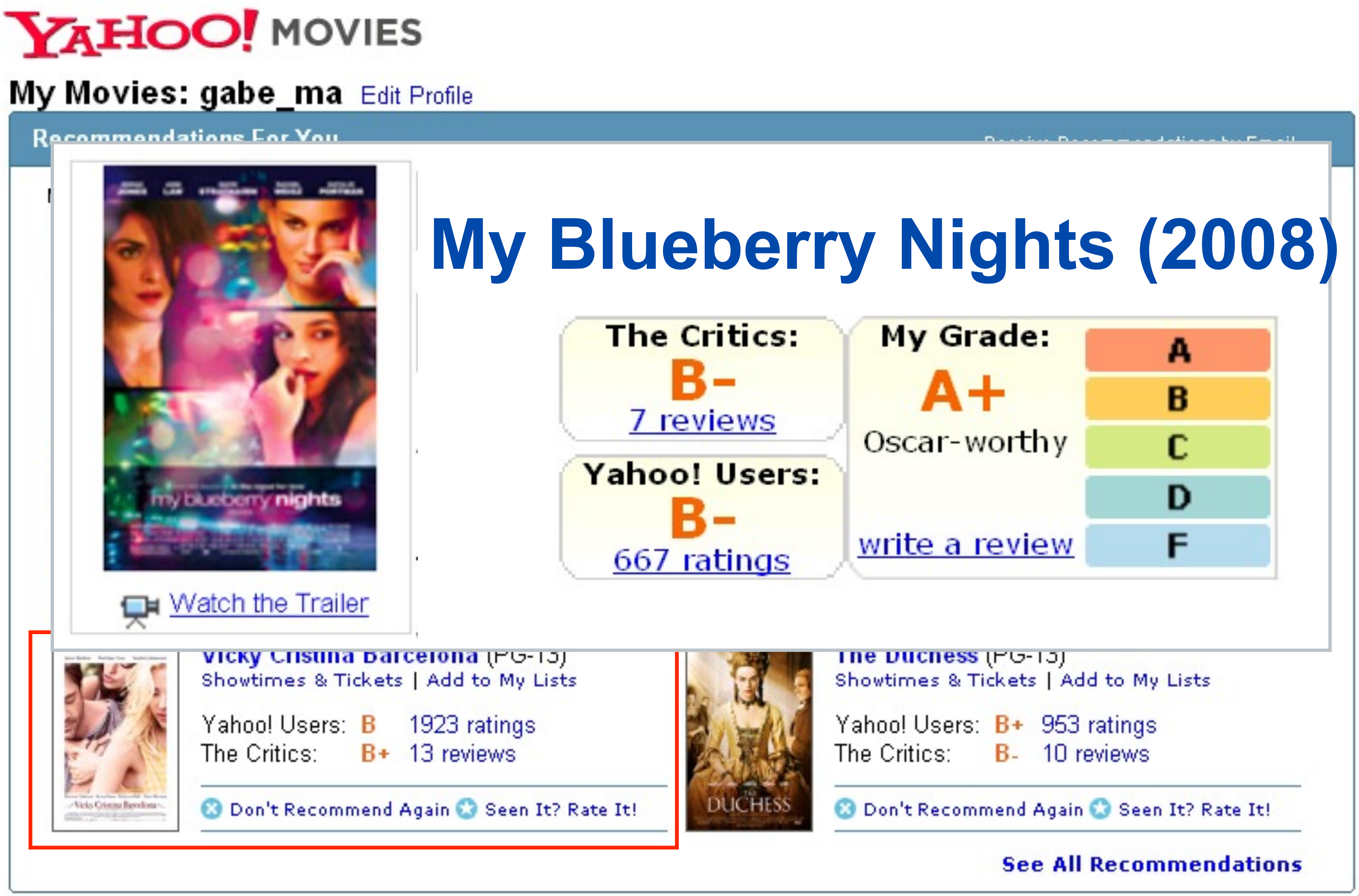
- Solving by multiplicative updating rules

# Social Recommender Systems

- Introduction
- Collaborative Filtering
- Trust-aware Recommender Systems
- Social-based Recommender Systems

# Challenges

- Data sparsity problem



# Challenges

## My Movie Ratings


**The Pursuit of Happyness** (PG-13, 1 hr. 57 min.)
Buy DVD | Add to My Lists

Yahoo! Users: B+ 38992 ratings
The Critics: B- 13 reviews

My Rating: A+


**Finding Nemo** (G, 1 hr. 40 min.)
Buy DVD | Add to My Lists

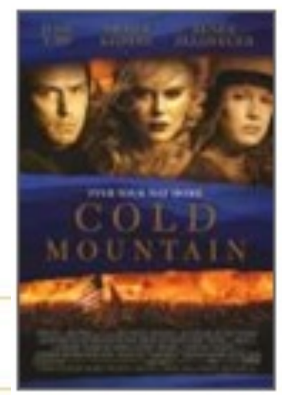
Yahoo! Users: B+ 137394 ratings
The Critics: A- 14 reviews

My Rating: A


**My Blueberry Nights** (PG-13, 1 hr. 30 min.)
Buy DVD | Add to My Lists

Yahoo! Users: B- 756 ratings
The Critics: B- 7 reviews

My Rating: A+




**Cold Mountain** (R, 2 hrs. 35 min.)
Buy DVD | Add to My Lists

Yahoo! Users: B 38986 ratings
The Critics: B+ 10 reviews

My Rating: B+


**The Lord of the Rings: The Fellowship of the Ring**
Buy DVD | Add to My Lists

Yahoo! Users: A- 110957 ratings
The Critics: A 15 reviews

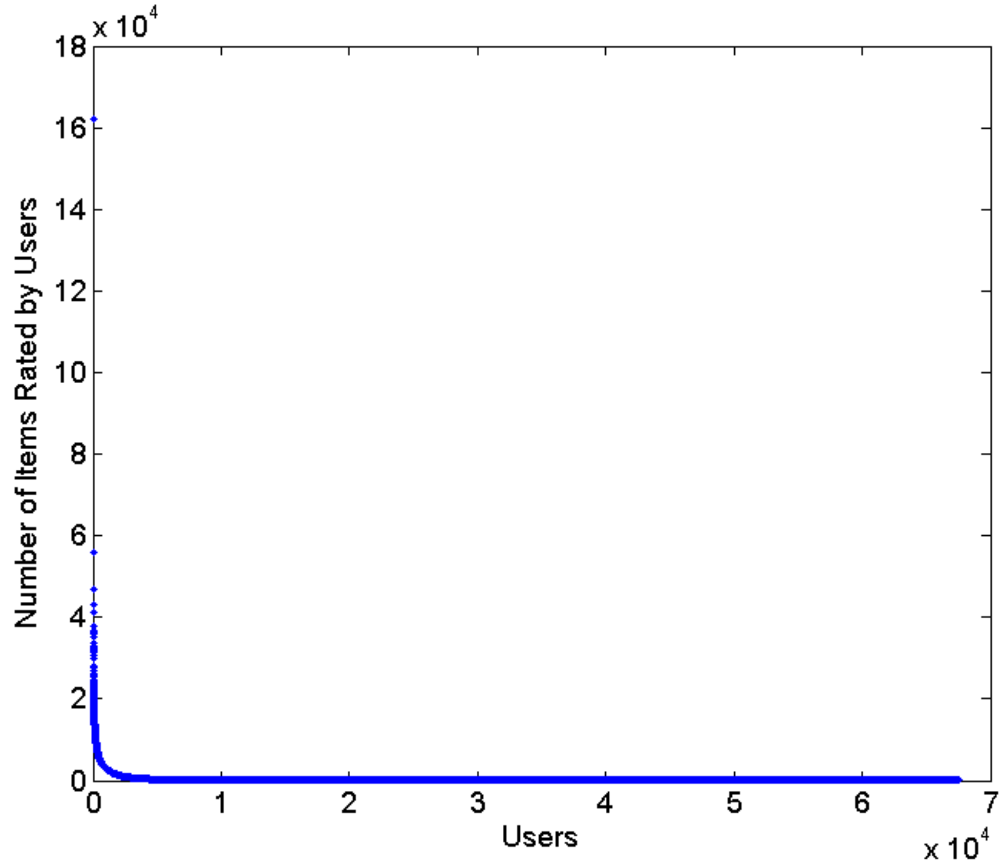
My Rating: A

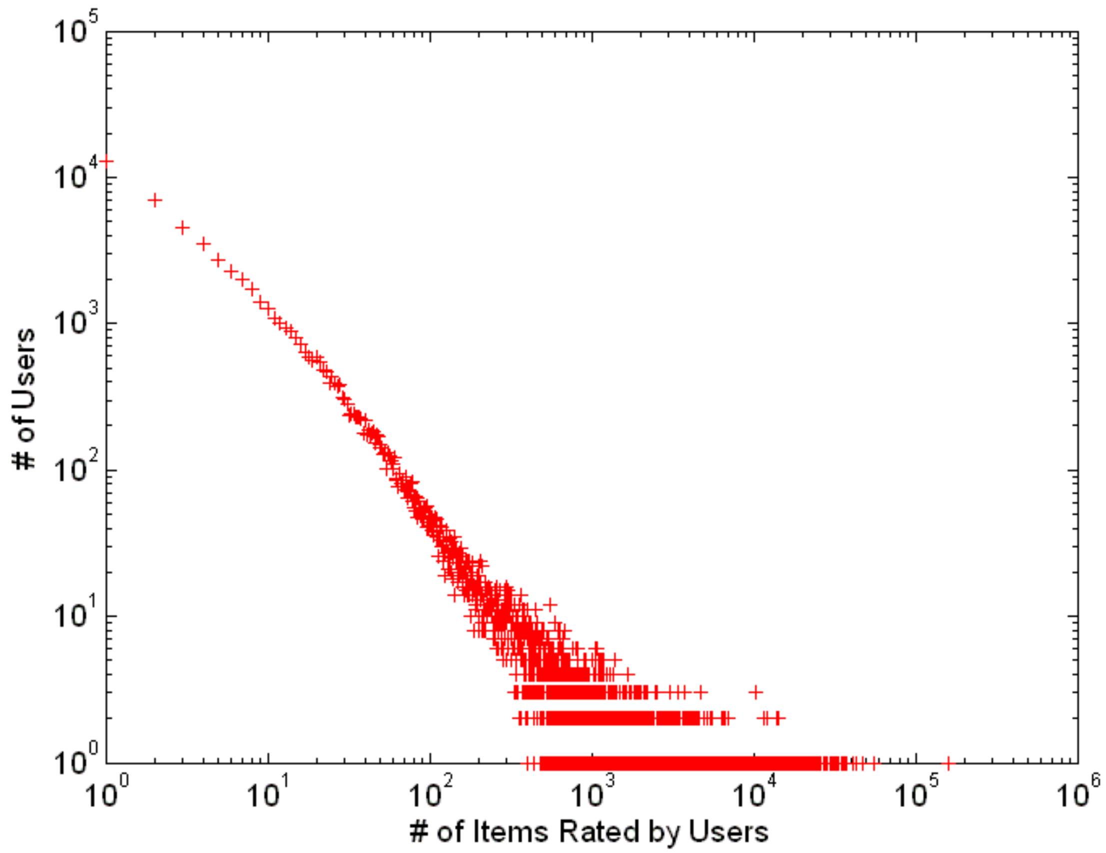

**Shrek 2** (PG, 1 hr. 32 min.)
Buy DVD | Add to My Lists

Yahoo! Users: B+ 150368 ratings
The Critics: B 15 reviews

My Rating: B

# Number of Ratings per User




Extracted From Epinions.com

114,222 users, 754,987 items and 13,385,713 ratings

# Challenges

- Traditional recommender systems ignore the social connections between users

Recommendations from friends

# Social Recommendation Using Probabilistic Matrix Factorization

[Hao Ma, et al., CIKM2008]

# Motivations

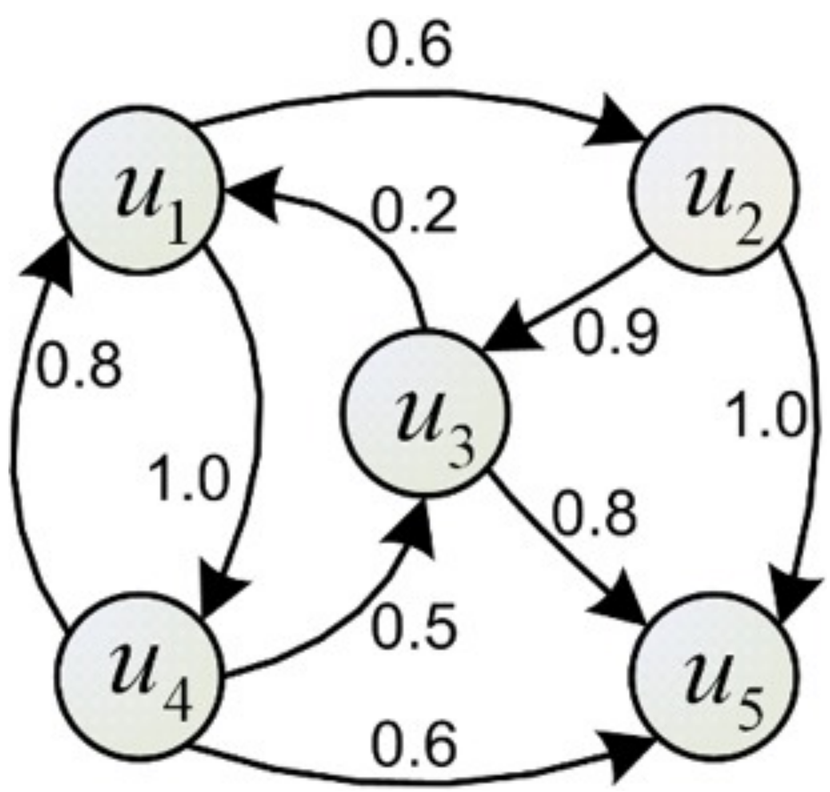
- "Yes, there is a correlation - from social networks to personal behavior on the web"

  Parag Singla and Matthew Richardson (WWW'08)

  - Analyze the who talks to whom social network over 10 million people with their related search results
  - People who chat with each other are more likely to share the same or similar interests

- To improve the recommendation accuracy and solve the data sparsity problem, users' social network should be taken into consideration

# Problem Definition



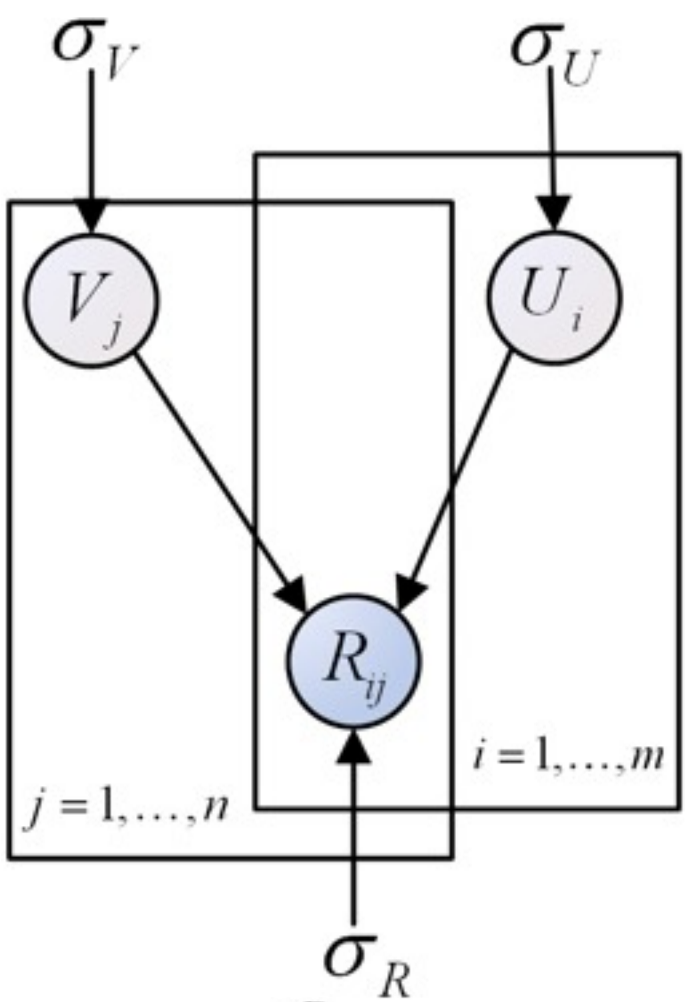

Social Trust Graph

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ |
|---|---|---|---|---|---|---|
| $u_1$ | | 5 | 2 | | 3 | |
| $u_2$ | 4 | | | 3 | | 4 |
| $u_3$ | | | 2 | | | 2 |
| $u_4$ | 5 | | | 3 | | |
| $u_5$ | | 5 | 5 | | | 3 |

User-Item Rating Matrix

# User-Item Matrix Factorization

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ |
|---|---|---|---|---|---|---|
| $u_1$ | | 5 | 2 | | 3 | |
| $u_2$ | 4 | | | 3 | | 4 |
| $u_3$ | | | 2 | | | 2 |
| $u_4$ | 5 | | | 3 | | |
| $u_5$ | | 5 | 5 | | | 3 |



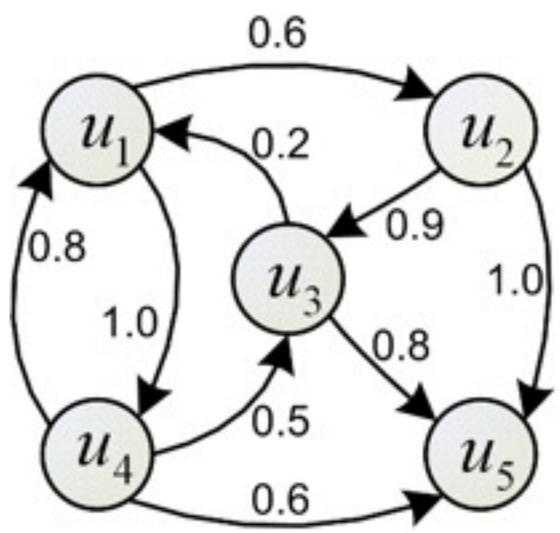

$$p(R|U, V, \sigma_R^2) = \prod_{i=1}^{m} \prod_{j=1}^{n} \left[ \mathcal{N}\left( R_{ij} | g(U_i^T V_j), \sigma_R^2 \right) \right]^{I_{ij}^R}$$

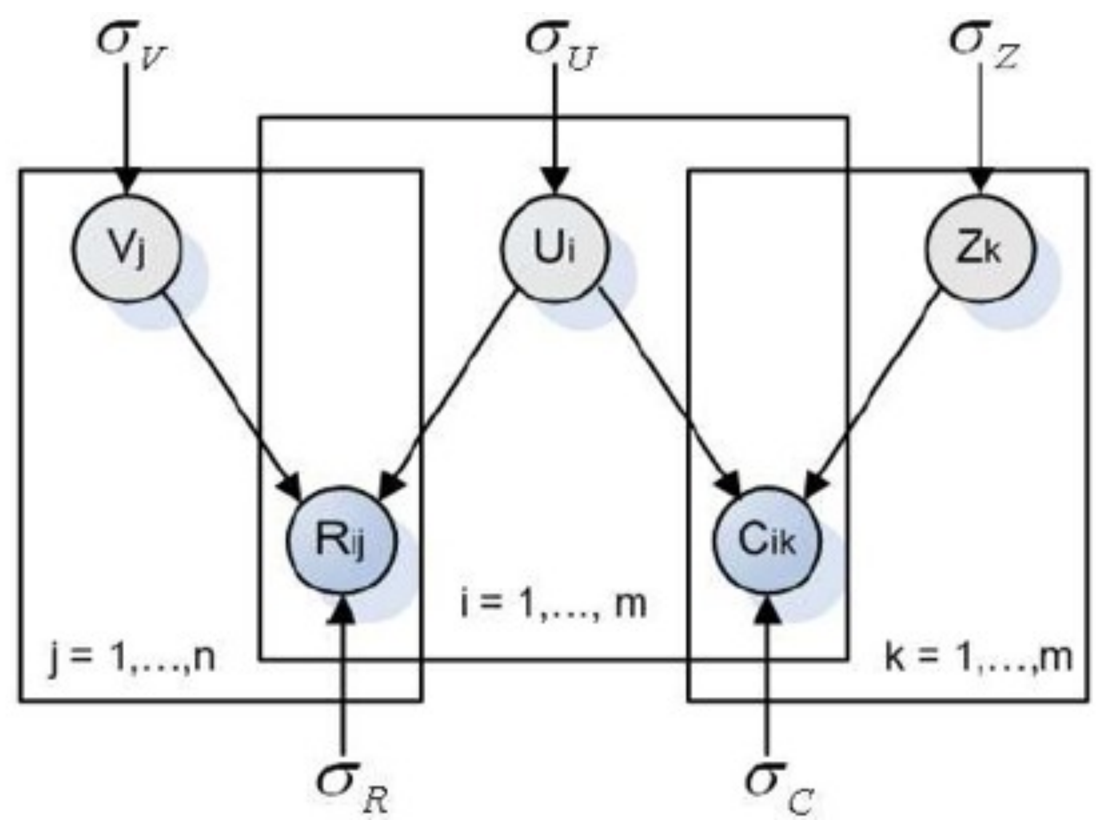
$$p(U|\sigma_U^2) = \prod_{i=1}^{m} \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I}) \qquad p(V|\sigma_V^2) = \prod_{j=1}^{n} \mathcal{N}(V_j|0, \sigma_V^2 \mathbf{I})$$

R. Salakhutdinov and A. Mnih (NIPS'08)

# SoRec

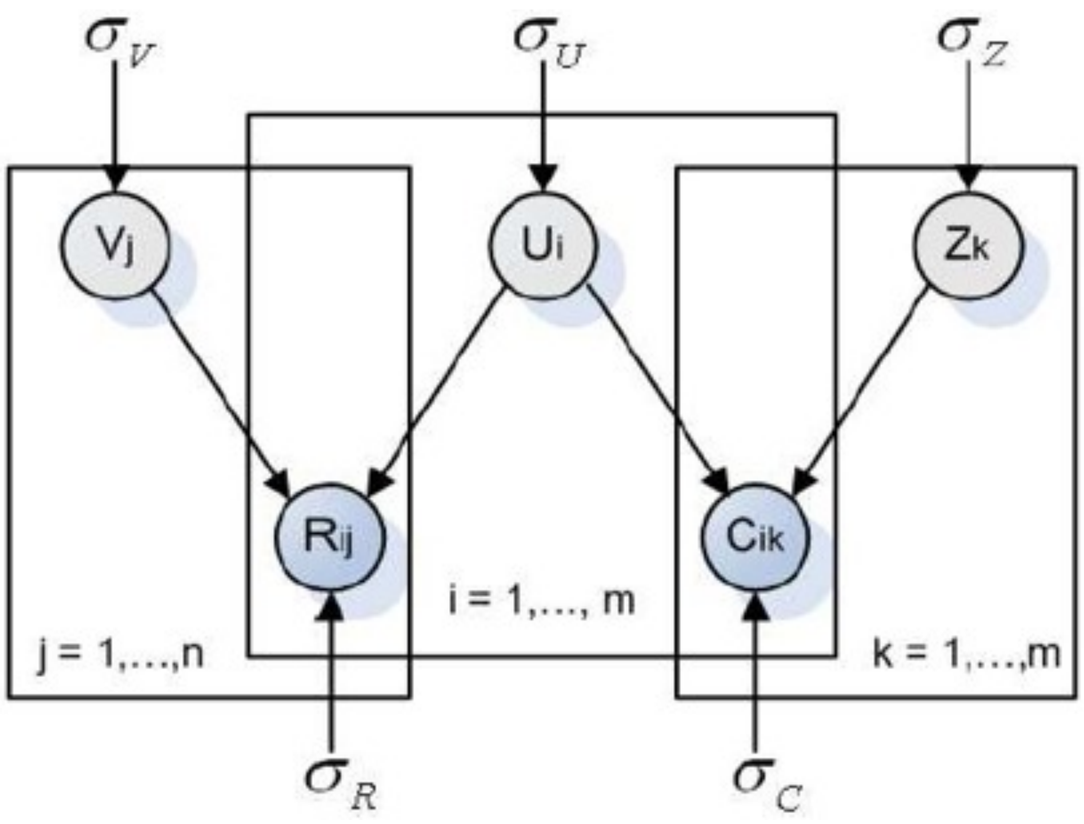
| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ |
|---|---|---|---|---|---|---|
| $u_1$ | | 5 | 2 | | 3 | |
| $u_2$ | 4 | | | 3 | | 4 |
| $u_3$ | | | 2 | | | 2 |
| $u_4$ | 5 | | | 3 | | |
| $u_5$ | | 5 | 5 | | | 3 |




SoRec

# SoRec




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

$$p(C|U,Z,\sigma_C^2) = \prod_{i=1}^{m}\prod_{k=1}^{m} \mathcal{N}\left[\left(c_{ik}|g(U_i^T Z_k),\sigma_C^2\right)\right]^{I_{ik}^C}$$

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

$$p(Z|\sigma_Z^2) = \prod_{k=1}^{m} \mathcal{N}(Z_k|0,\sigma_Z^2\mathbf{I})$$

$$\mathcal{L}(R,C,U,V,Z) =$$
$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R(r_{ij}-g(U_i^T V_j))^2+\frac{\lambda_C}{2}\sum_{i=1}^{m}\sum_{k=1}^{m} I_{ik}^C(c_{ik}^*-g(U_i^T Z_k))^2$$
$$+\frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2 + \frac{\lambda_Z}{2}\|Z\|_F^2,$$

# SoRec

$$\frac{\partial \mathcal{L}}{\partial U_i} = \sum_{j=1}^{n} I_{ij}^R g'(U_i^T V_j)(g(U_i^T V_j) - r_{ij})V_j$$

$$+ \lambda_C \sum_{j=1}^{m} I_{ik}^C g'(U_i^T Z_k)(g(U_i^T Z_k) - c_{ik}^*)Z_k + \lambda_U U_i,$$

$$\frac{\partial \mathcal{L}}{\partial V_j} = \sum_{i=1}^{m} I_{ij}^R g'(U_i^T V_j)(g(U_i^T V_j) - r_{ij})U_i + \lambda_V V_j,$$

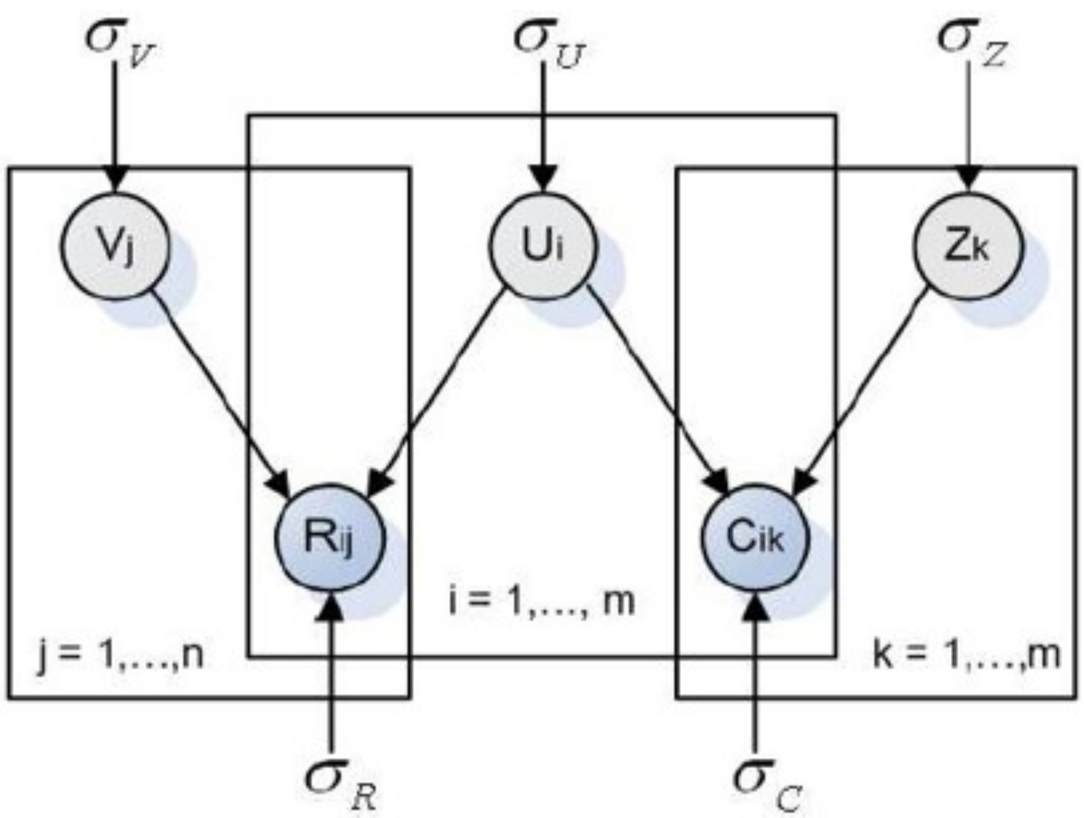
$$\frac{\partial \mathcal{L}}{\partial Z_k} = \lambda_C \sum_{i=1}^{m} I_{ik}^C g'(U_i^T Z_k)(g(U_i^T Z_k) - c_{ik}^*)U_i + \lambda_Z Z_k,$$

# Complexity Analysis

- For the Objective Function $O(\rho_R l + \rho_C l)$
- For $\frac{\partial \mathcal{L}}{\partial U}$ the complexity is $O(\rho_R l + \rho_C l)$
- For $\frac{\partial \mathcal{L}}{\partial V}$ the complexity is $O(\rho_R l)$
- For $\frac{\partial \mathcal{L}}{\partial Z}$ the complexity is $O(\rho_C l)$
- In general, the complexity of our method is linear with the observations in these two matrices

# Disadvantages of SoRec

- Lack of interpretability
- Does not reflect the real-world recommendation process

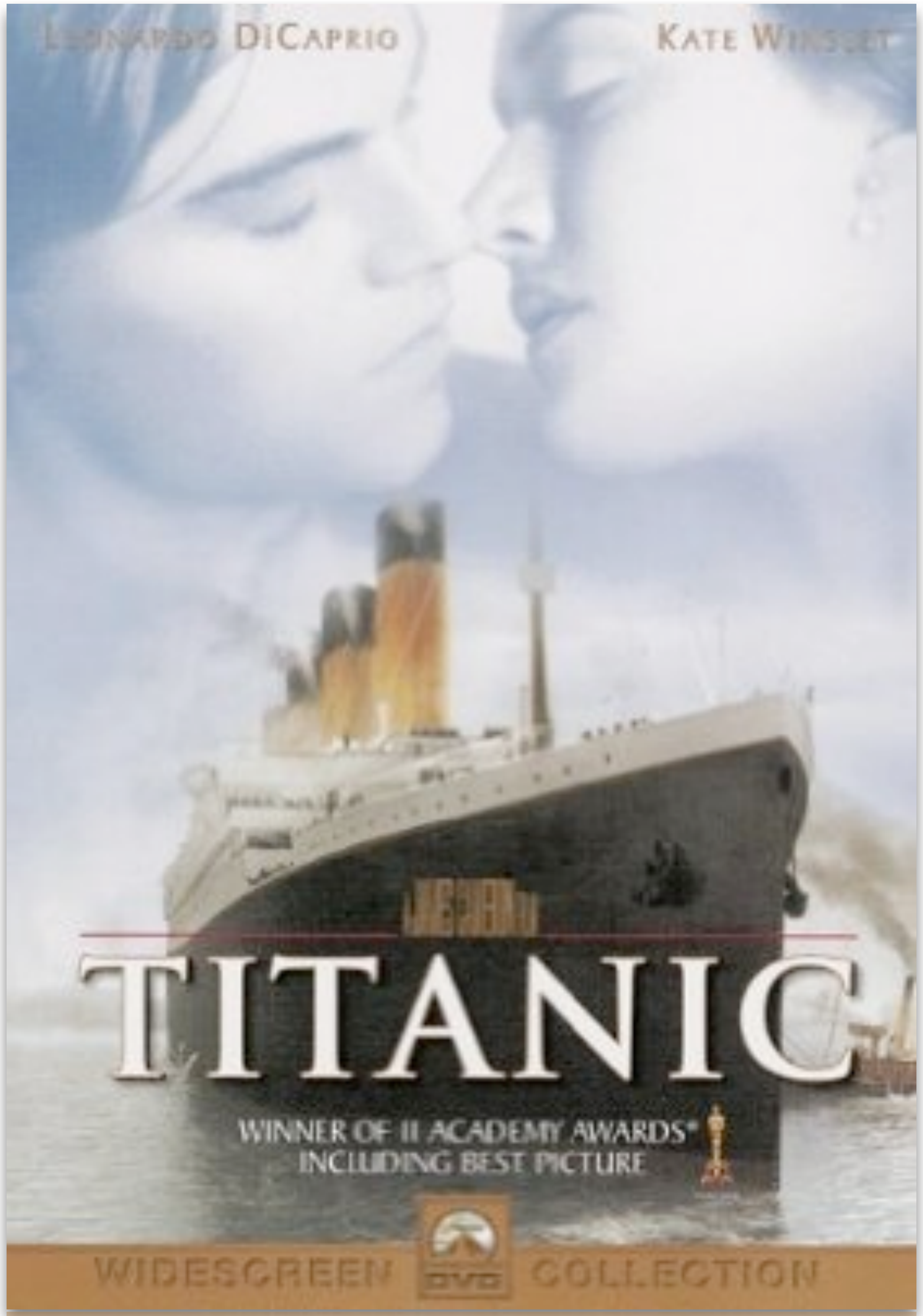



SoRec

# Learning to Recommend with Social Trust Ensemble

[Hao Ma, et al., SIGIR2009]

# 1st Motivation

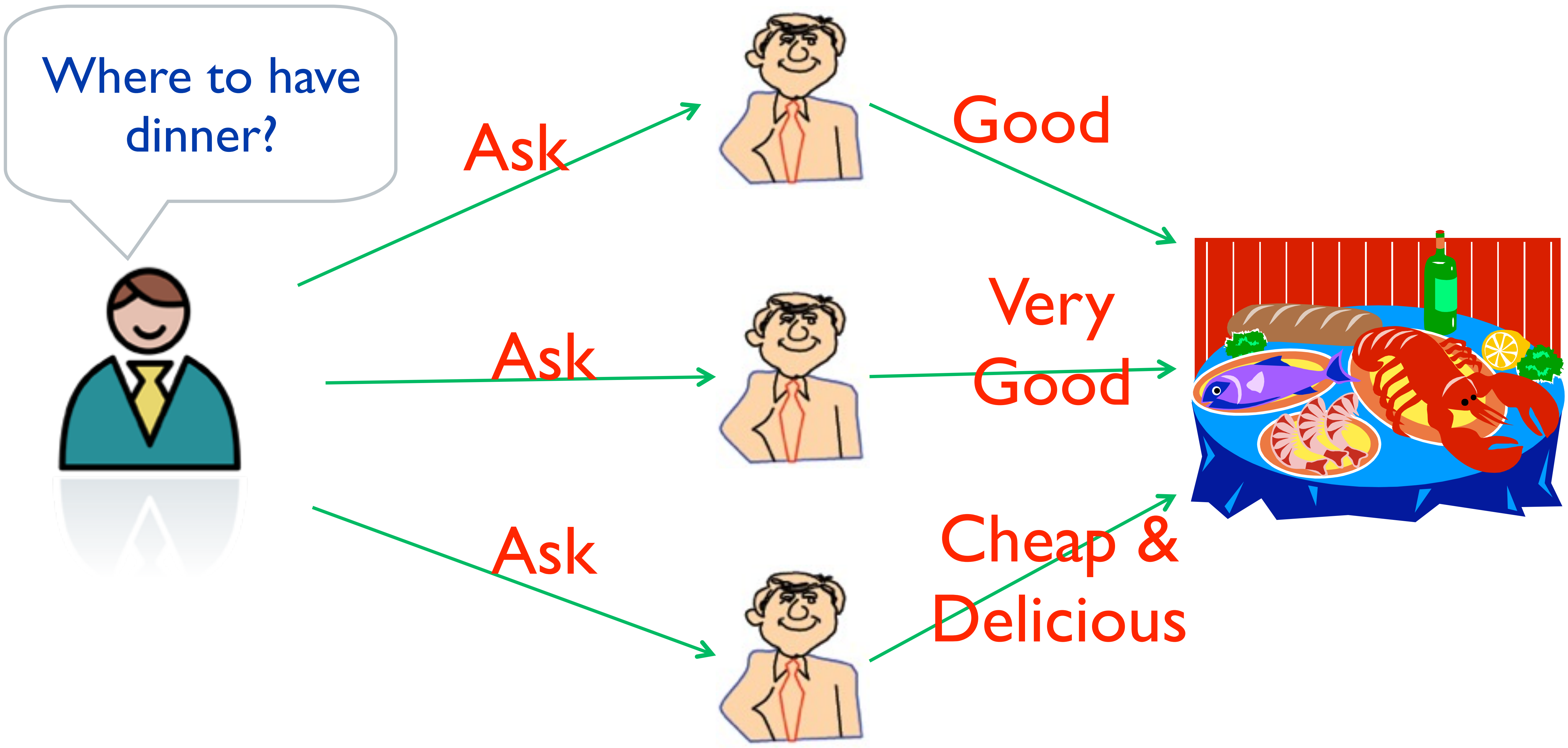
# 1st Motivation


# 1$^{st}$ Motivation

- Users have their own characteristics, and they have different tastes on different items, such as movies, books, music, articles, food, etc.

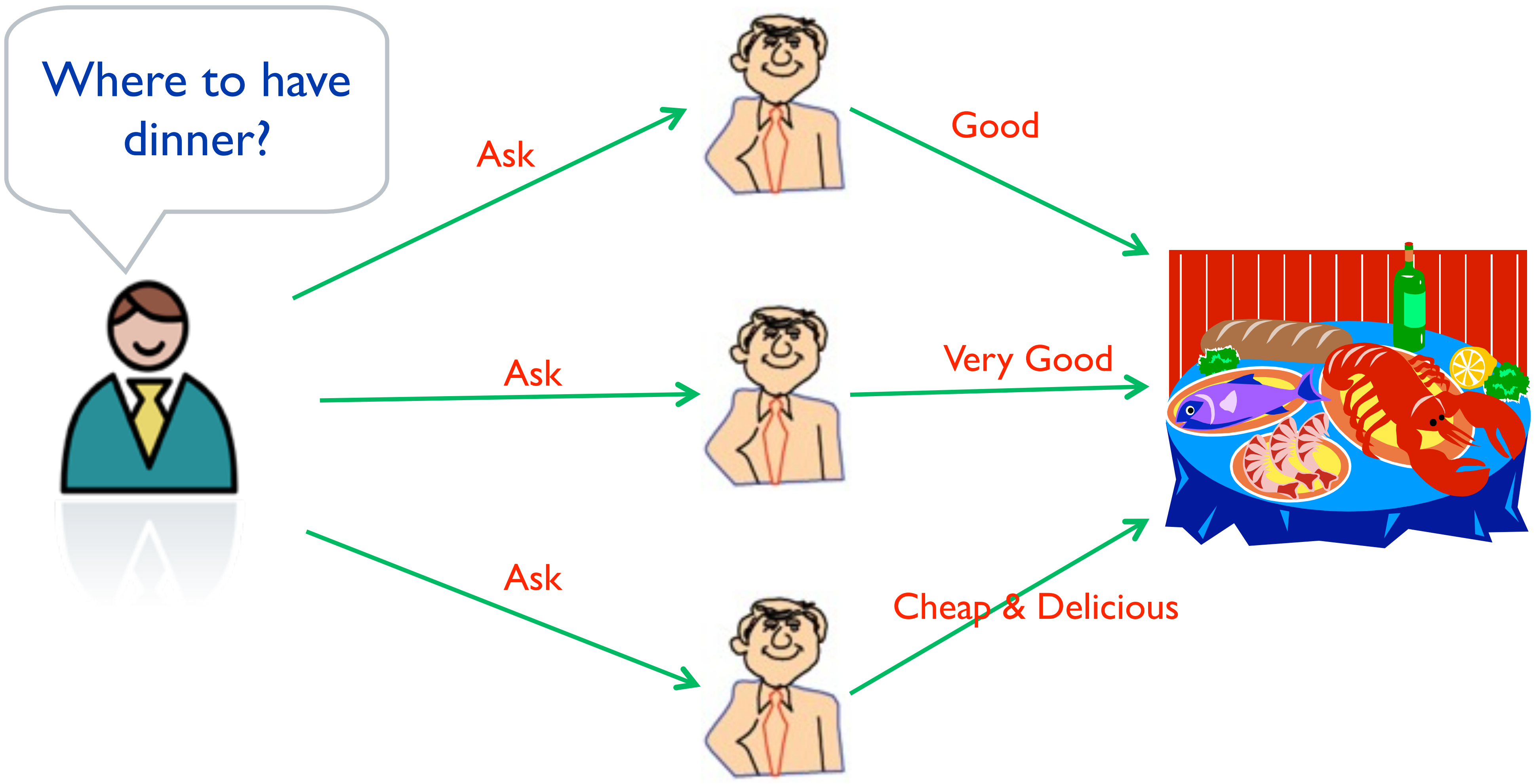



# 2nd Motivation



# 2nd Motivation

- Users can be easily influenced by the friends they trust, and prefer their friends' recommendations.



# Motivations

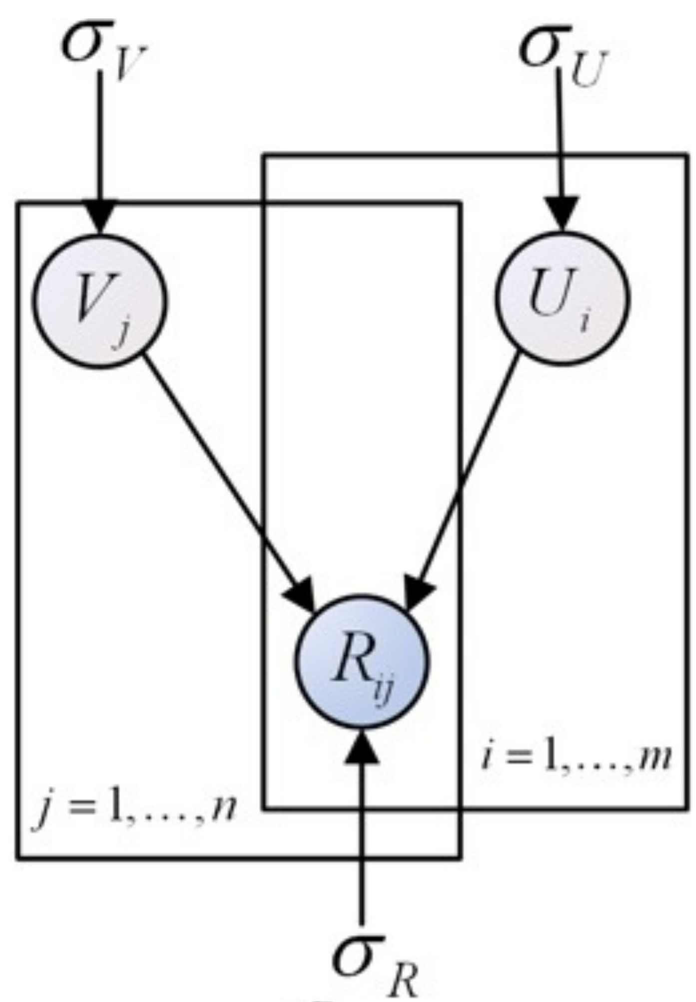
- Users have their own characteristics, and they have different tastes on different items, such as movies, books, music, articles, food, etc.

- Users can be easily influenced by the friends they trust, and prefer their friends' recommendations.

- One user's final decision is the balance between his/her own taste and his/her trusted friends' favors.

# User-Item Matrix Factorization

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ |
|---|---|---|---|---|---|---|
| $u_1$ | | 5 | 2 | | 3 | |
| $u_2$ | 4 | | | 3 | | 4 |
| $u_3$ | | | 2 | | | 2 |
| $u_4$ | 5 | | | 3 | | |
| $u_5$ | | 5 | 5 | | | 3 |




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

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

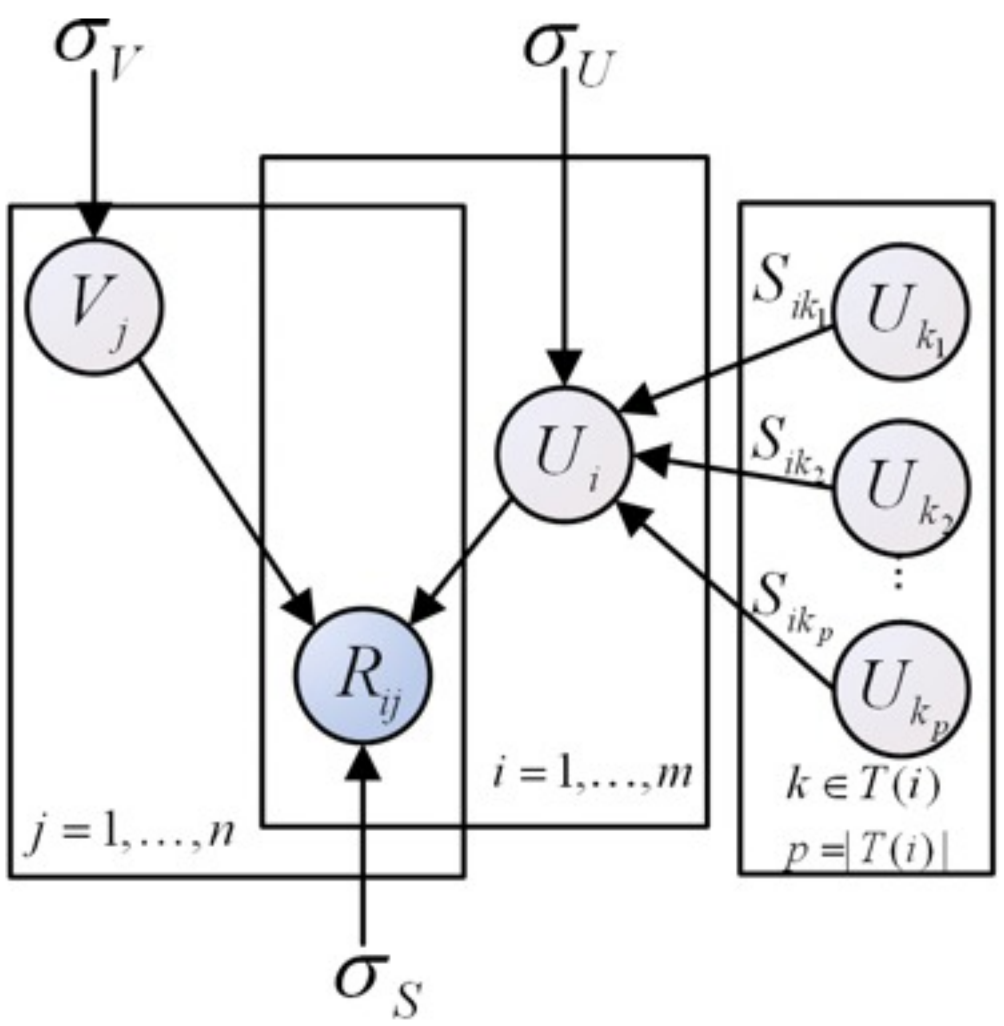
[R. Salakhutdinov, et al., NIPS2008]

# Recommendations by Trusted Friends

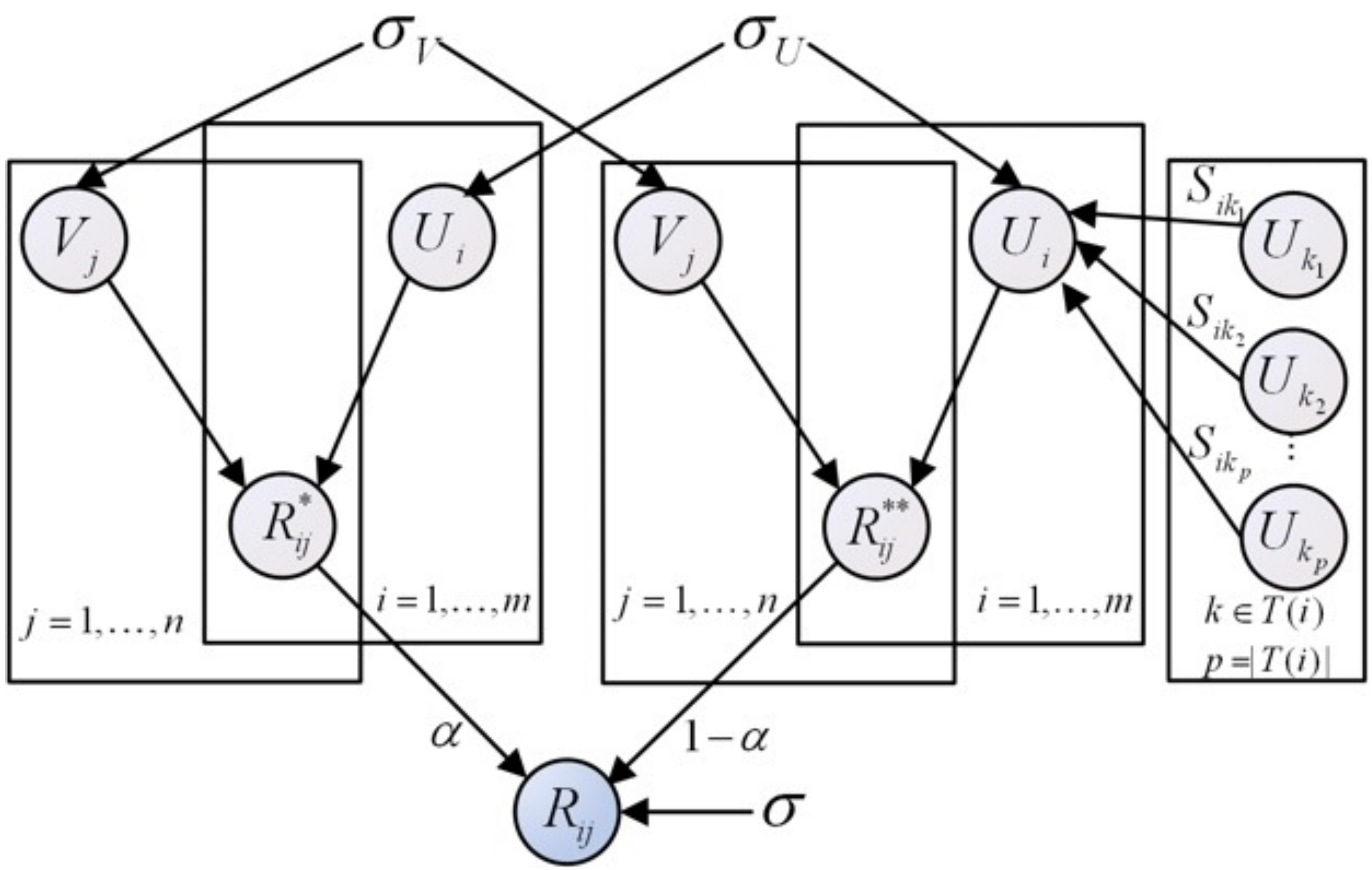
$$\widehat{R}_{ik} = \frac{\sum_{j \in \mathcal{T}(i)} R_{jk} S_{ij}}{|\mathcal{T}(i)|}$$

$$\widehat{R}_{ik} = \sum_{j \in \mathcal{T}(i)} R_{jk} S_{ij}$$

$$p(R|S,U,V,\sigma_R^2) = \prod_{i=1}^{m} \prod_{j=1}^{n} \left[ \mathcal{N}\left( R_{ij} | g\left( \sum_{k \in \mathcal{T}(i)} S_{ik} U_k^T V_j \right), \sigma_S^2 \right) \right]^{I_{ij}^R}$$



# Recommendation with Social Trust Ensemble




$$\prod_{i=1}^{m} \prod_{j=1}^{n} \left[ \mathcal{N}\left( R_{ij} | g(\alpha U_i^T V_j + (1-\alpha) \sum_{k \in \mathcal{T}(i)} S_{ik} U_k^T V_j), \sigma^2 \right) \right]^{I_{ij}^R}$$

# Recommendation with Social Trust Ensemble

$$\mathcal{L}(R,S,U,V) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}\Big(R_{ij} - g\big(\alpha U_i^T V_j + (1-\alpha)\sum_{k\in\mathcal{T}(i)} S_{ik}U_k^T V_j\big)\Big)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2, \tag{15}$$

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial U_i} =& \alpha\sum_{j=1}^{n} I_{ij}^{R} g'\big(\alpha U_i^T V_j + (1-\alpha)\sum_{k\in\mathcal{T}(i)} S_{ik}U_k^T V_j\big)V_j \\ &\times\big(g(\alpha U_i^T V_j + (1-\alpha)\sum_{k\in\mathcal{T}(i)} S_{ik}U_k^T V_j) - R_{ij}\big) \\ &+(1-\alpha)\sum_{p\in\mathcal{B}(i)}\sum_{j=1}^{n} I_{pj}^{R} g'\big(\alpha U_p^T V_j + (1-\alpha)\sum_{k\in\mathcal{T}(p)} S_{pk}U_k^T V_j\big) \\ &\times\big(g(\alpha U_p^T V_j + (1-\alpha)\sum_{k\in\mathcal{T}(p)} S_{pk}U_k^T V_j) - R_{pj}\big)S_{pi}V_j \\ &+\lambda_U U_i,\end{aligned}$$

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial V_j} =& \sum_{i=1}^{m} I_{ij}^{R} g'\big(\alpha U_i^T V_j + (1-\alpha)\sum_{k\in\mathcal{T}(i)} S_{ik}U_k^T V_j\big) \\ &\times\big(g(\alpha U_i^T V_j + (1-\alpha)\sum_{k\in\mathcal{T}(i)} S_{ik}U_k^T V_j) - R_{ij}\big) \\ &\times\big(\alpha U_i + (1-\alpha)\sum_{k\in\mathcal{T}(i)} S_{ik}U_k^T\big) + \lambda_V V_j,\end{aligned}$$

# Complexity

- In general, the complexity of this method is linear with the observations the user-item matrix

# Epinions Dataset

- 51,670 users who rated 83,509 items with totally 631,064 ratings
- Rating Density 0.015%
- The total number of issued trust statements is 511,799

# Metrics

- Mean Absolute Error and Root Mean Square Error

$$MAE = \frac{\sum_{i,j} |r_{i,j} - \widehat{r}_{i,j}|}{N}$$

$$RMSE = \sqrt{\frac{\sum_{i,j} (r_{i,j} - \widehat{r}_{i,j})^2}{N}}$$

# Comparisons

**Table III: Performance Comparisons (A Smaller MAE or RMSE Value Means a Better Performance)**

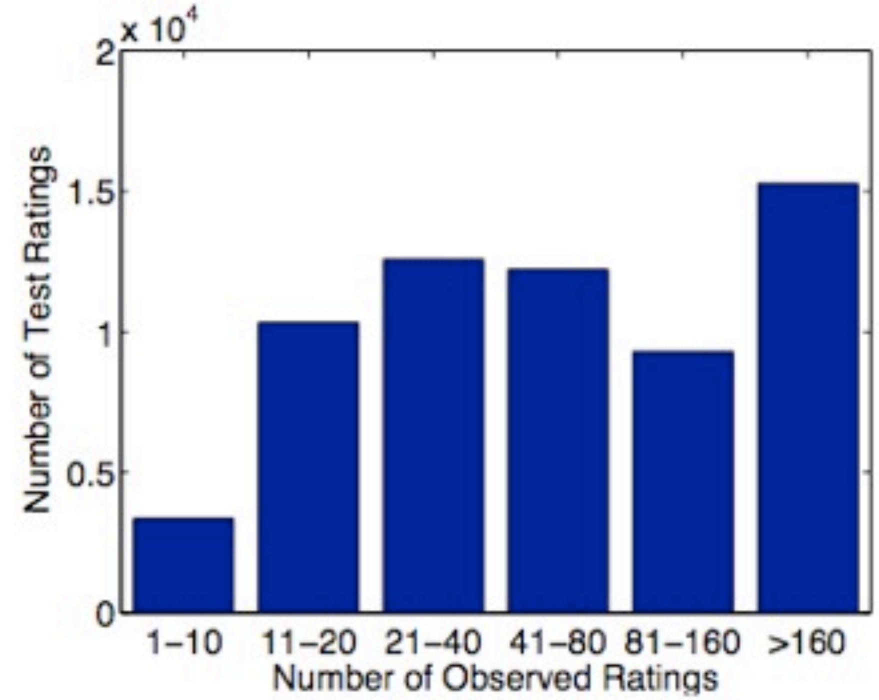
| Training Data | Metrics | Dimensionality = 5 | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | UserMean | ItemMean | NMF | PMF | Trust | SoRec | RSTE |
| 90% | MAE | 0.9134 | 0.9768 | 0.8738 | 0.8676 | 0.9054 | 0.8442 | **0.8377** |
| | RMSE | 1.1688 | 1.2375 | 1.1649 | 1.1575 | 1.1959 | 1.1333 | **1.1109** |
| 80% | MAE | 0.9285 | 0.9913 | 0.8975 | 0.8951 | 0.9221 | 0.8638 | **0.8594** |
| | RMSE | 1.1817 | 1.2584 | 1.1861 | 1.1826 | 1.2140 | 1.1530 | **1.1346** |
| **Training Data** | **Metrics** | **Dimensionality = 10** | | | | | | |
| | | UserMean | ItemMean | NMF | PMF | Trust | SoRec | RSTE |
| 90% | MAE | 0.9134 | 0.9768 | 0.8712 | 0.8651 | 0.9039 | 0.8404 | **0.8367** |
| | RMSE | 1.1688 | 1.2375 | 1.1621 | 1.1544 | 1.1917 | 1.1293 | **1.1094** |
| 80% | MAE | 0.9285 | 0.9913 | 0.8951 | 0.8886 | 0.9215 | 0.8580 | **0.8537** |
| | RMSE | 1.1817 | 1.2584 | 1.1832 | 1.1760 | 1.2132 | 1.1492 | **1.1256** |

NMF --- D. D. Lee and H. S. Seung (Nature 1999)
PMF --- R. Salakhutdinov and A. Mnih (NIPS 2008)
SoRec --- H. Ma, H. Yang, M. R. Lyu and I. King (CIKM 2008)
Trust, RSTE --- H. Ma, I. King and M. R. Lyu (SIGIR 2009)

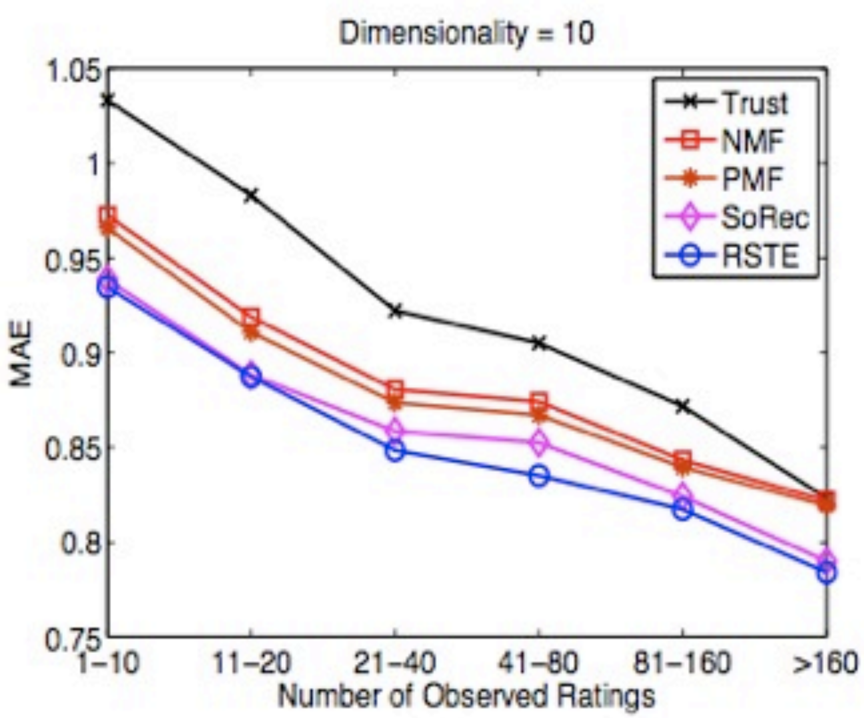
# Performance on Different Users

- Group all the users based on the number of observed ratings in the training data
- 6 classes: "1 − 10", "11 − 20", "21 − 40", "41 − 80", "81 − 160", "> 160",

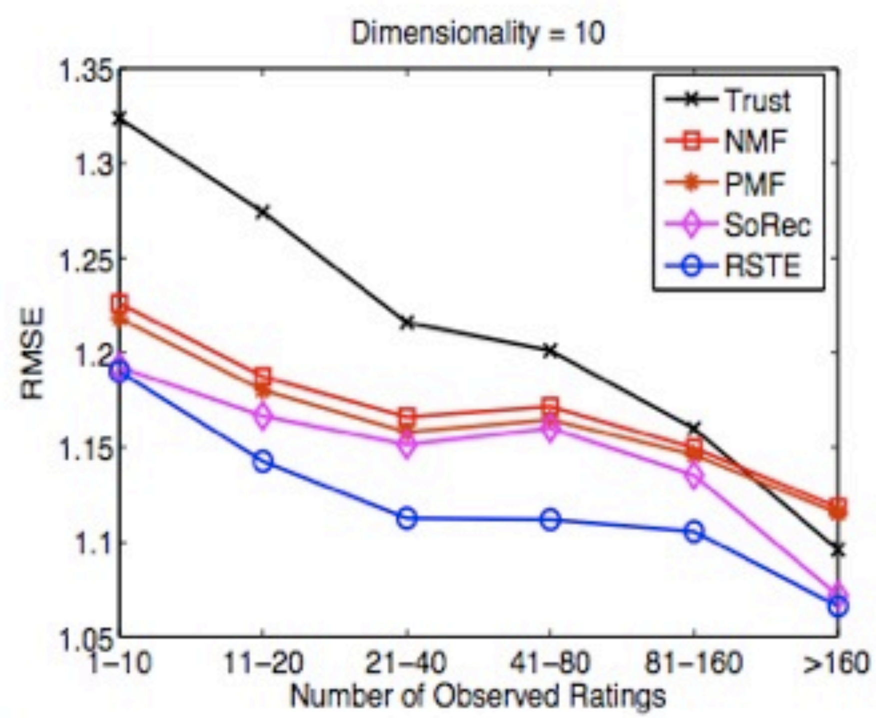
# Performance on Different Users




(a) Distribution of Testing Data (90% as Training Data)

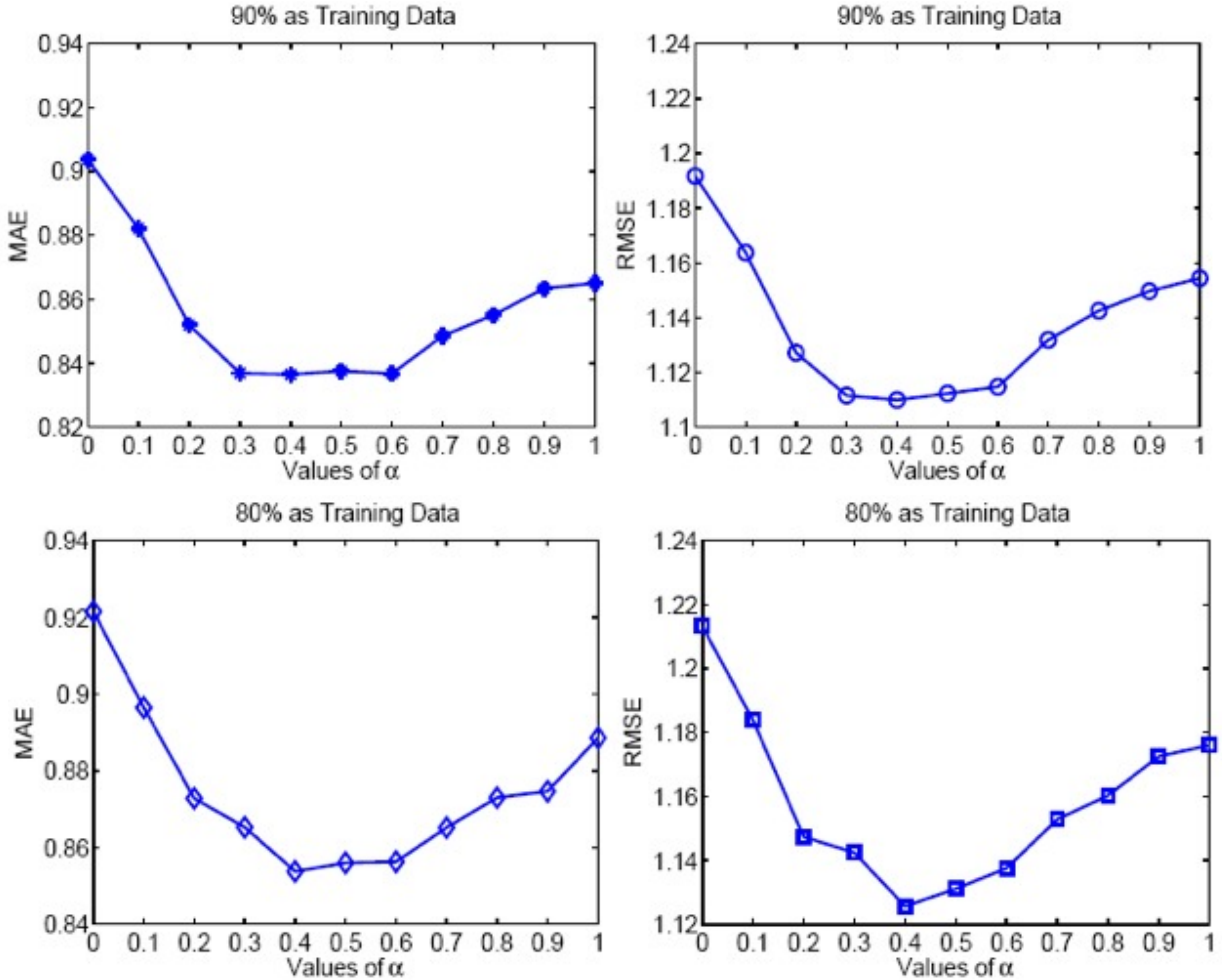



(b) MAE Comparison on Different User Rating Scales (90% as Training Data)



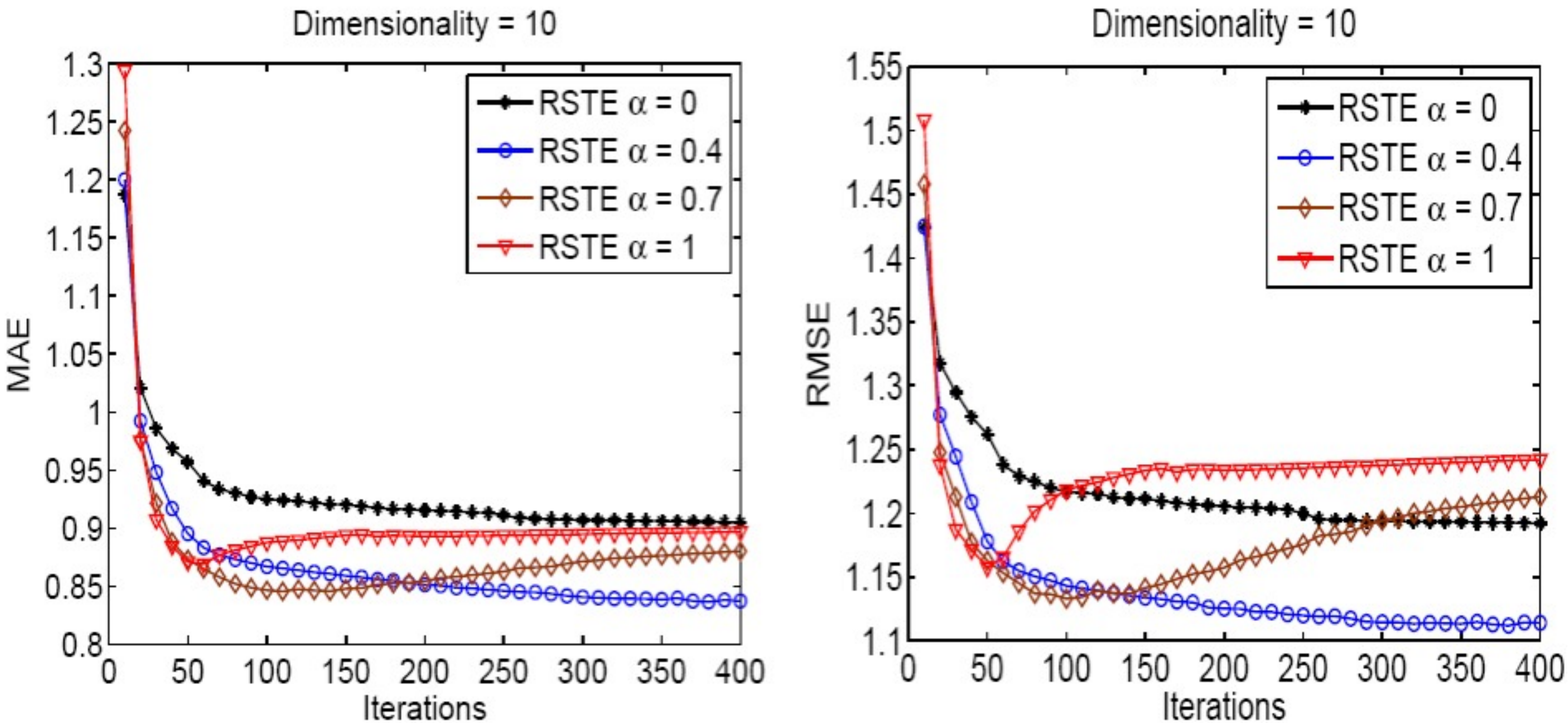

(c) RMSE Comparison on Different User Rating Scales (90% as Training Data)

# Impact of Parameter Alpha




Impact of Parameter $\alpha$ (Dimensionality = 10)

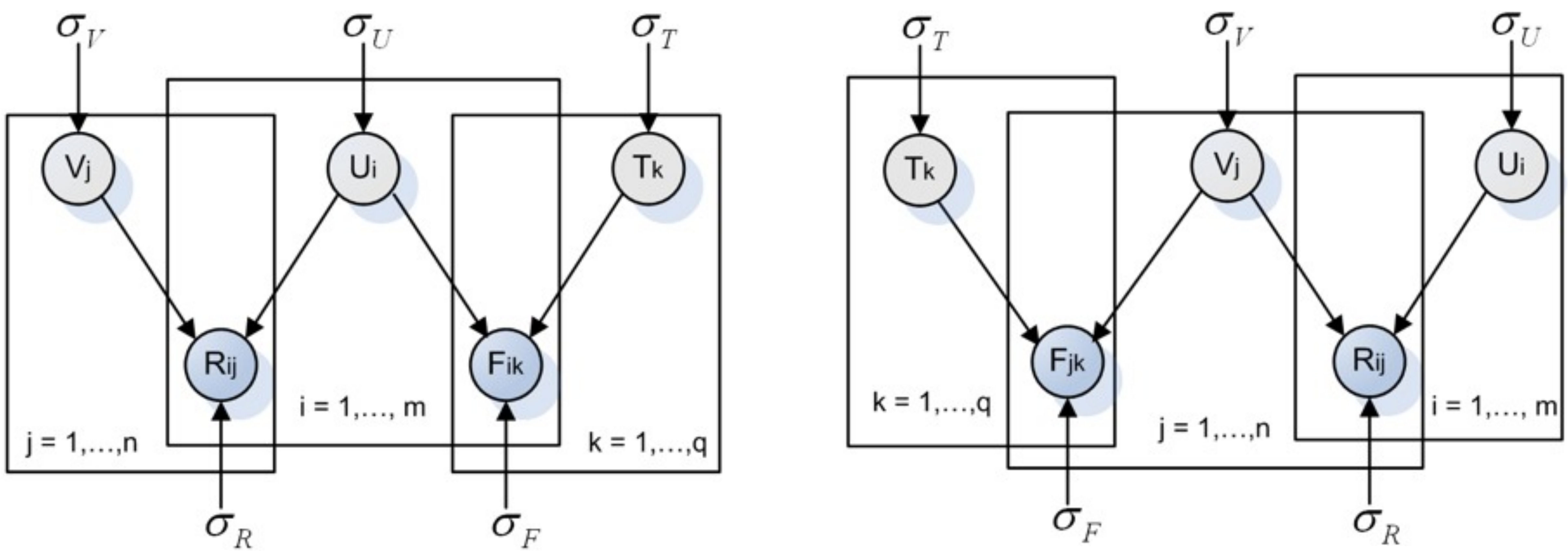
# MAE and RMSE Changes with Iterations




90% as Training Data

# Further Discussion of SoRec

- Improving Recommender Systems Using Social Tags




MovieLens Dataset

71,567 users, 10,681 movies,

10,000,054 ratings, 95,580 tags

# Further Discussion of SoRec

- MAE

Table V: MAE comparison with other approaches on MovieLens dataset (A smaller MAE value means a better performance)

| Methods | | 80% Training | 50% Training | 30% Training | 10% Training |
|---|---|---|---|---|---|
| User Mean | | 0.7686 | 0.7710 | 0.7742 | 0.8234 |
| Item Mean | | 0.7379 | 0.7389 | 0.7399 | 0.7484 |
| 5D | SVD | 0.6390 | 0.6547 | 0.6707 | 0.7448 |
| | PMF | 0.6325 | 0.6542 | 0.6698 | 0.7430 |
| | SoRecUser | 0.6209 | 0.6419 | 0.6607 | 0.7040 |
| | SoRecItem | **0.6199** | **0.6407** | **0.6395** | **0.7026** |
| 10D | SVD | 0.6386 | 0.6534 | 0.6693 | 0.7431 |
| | PMF | 0.6312 | 0.6530 | 0.6683 | 0.7417 |
| | SoRecUser | 0.6197 | 0.6408 | 0.6595 | 0.7028 |
| | SoRecItem | **0.6187** | **0.6395** | **0.6584** | **0.7016** |

# Further Discussion of SoRec

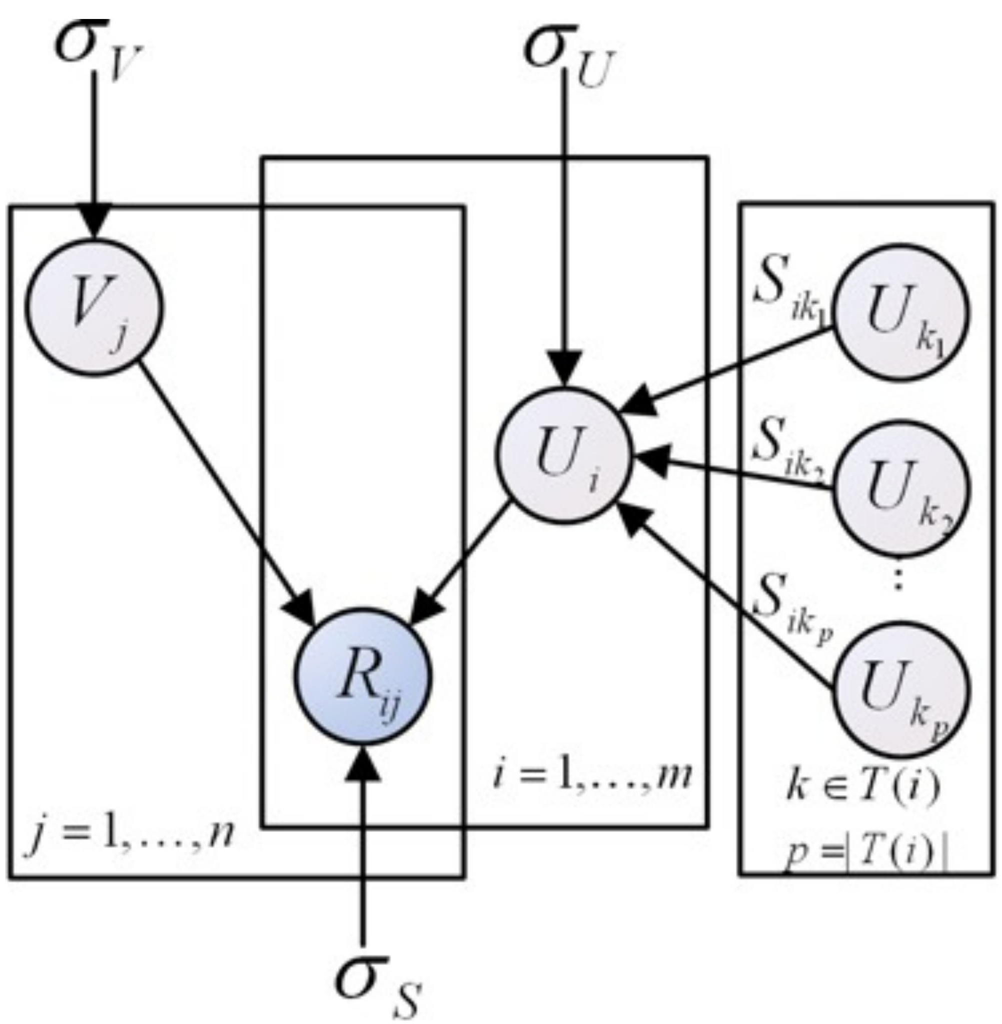
- RMSE

Table VI: RMSE comparison with other approaches on MovieLens dataset (A smaller RMSE value means a better performance)

| Methods | | 80% Training | 50% Training | 30% Training | 10% Training |
|---|---|---|---|---|---|
| User Mean | | 0.9779 | 0.9816 | 0.9869 | 1.1587 |
| Item Mean | | 0.9440 | 0.9463 | 0.9505 | 0.9851 |
| 5D | SVD | 0.8327 | 0.8524 | 0.8743 | 0.9892 |
| | PMF | 0.8310 | 0.8582 | 0.8758 | 0.9698 |
| | SoRecUser | 0.8121 | 0.8384 | 0.8604 | 0.9042 |
| | SoRecItem | **0.8112** | **0.8370** | **0.8591** | **0.9033** |
| 10D | SVD | 0.8312 | 0.8509 | 0.8728 | 0.9878 |
| | PMF | 0.8295 | 0.8569 | 0.8743 | 0.9681 |
| | SoRecUser | 0.8110 | 0.8372 | 0.8593 | 0.9034 |
| | SoRecItem | **0.8097** | **0.8359** | **0.8578** | **0.9019** |

# Further Discussion of RSTE

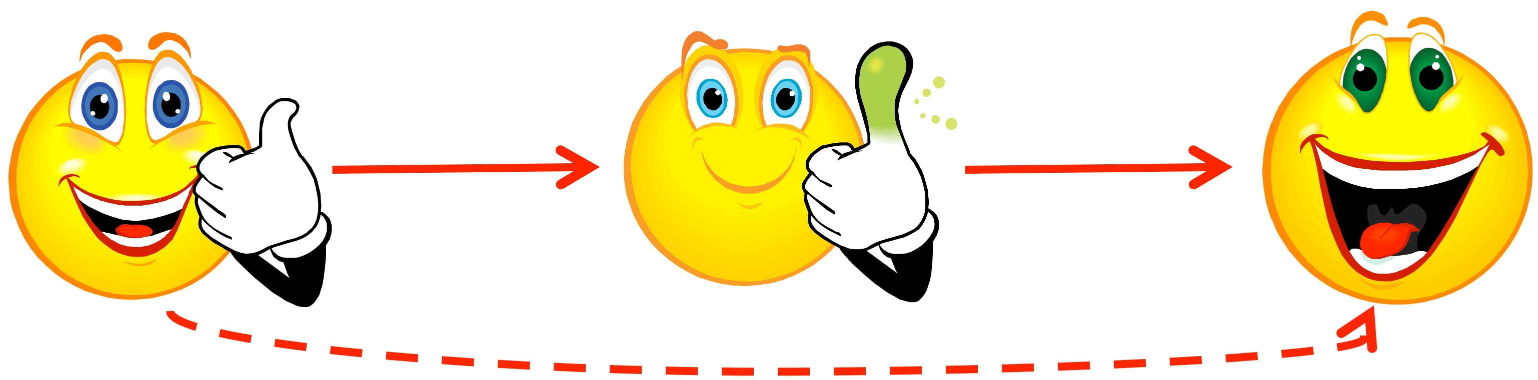
- Relationship with Neighborhood-based methods



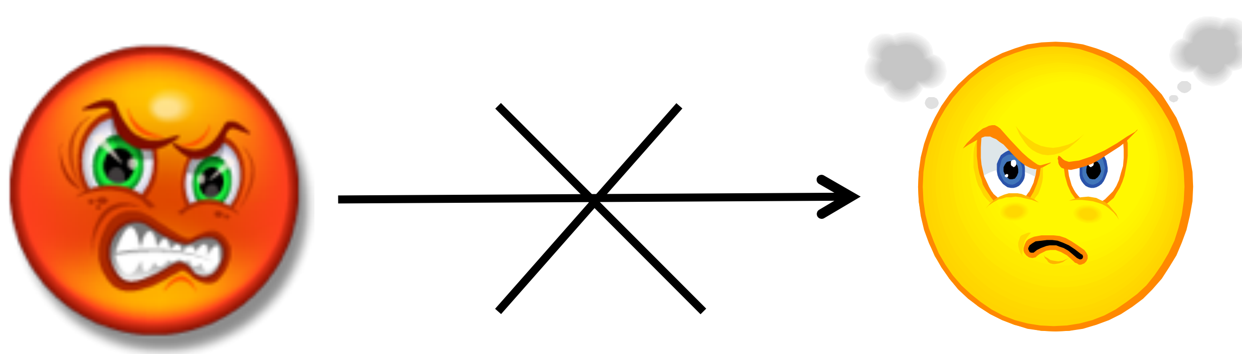

- The trusted friends are actually the explicit neighbors
- We can easily apply this method to include implicit neighbors
- Using PCC to calculate similar users for every user

# What We Cannot Model Using SoRec and RSTE?

- Propagation of trust



- Distrust



# Recommend with Social Distrust

[Hao Ma, et al., RecSys2009]

# Distrust

- Users' distrust relations can be interpreted as the "dissimilar" relations
  - On the web, user $U_i$ distrusts user $U_d$ indicates that user $U_i$ disagrees with most of the opinions issued by user $U_d$.

# Distrust

$$\max_{U} \frac{1}{2} \sum_{i=1}^{m} \sum_{d \in \mathcal{D}^{+}(i)} S_{id}^{\mathcal{D}} \|U_i - U_d\|_F^2$$

$$\begin{aligned}
\min_{U,V} \mathcal{L}_{\mathcal{D}}(R, S^{\mathcal{D}}, U, V) &= \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}^{R} (R_{ij} - g(U_i^T V_j))^2 \\
&+ \frac{\beta}{2} \sum_{i=1}^{m} \sum_{d \in \mathcal{D}^{+}(i)} (-S_{id}^{\mathcal{D}} \|U_i - U_d\|_F^2) \\
&+ \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2.
\end{aligned}$$

# Trust

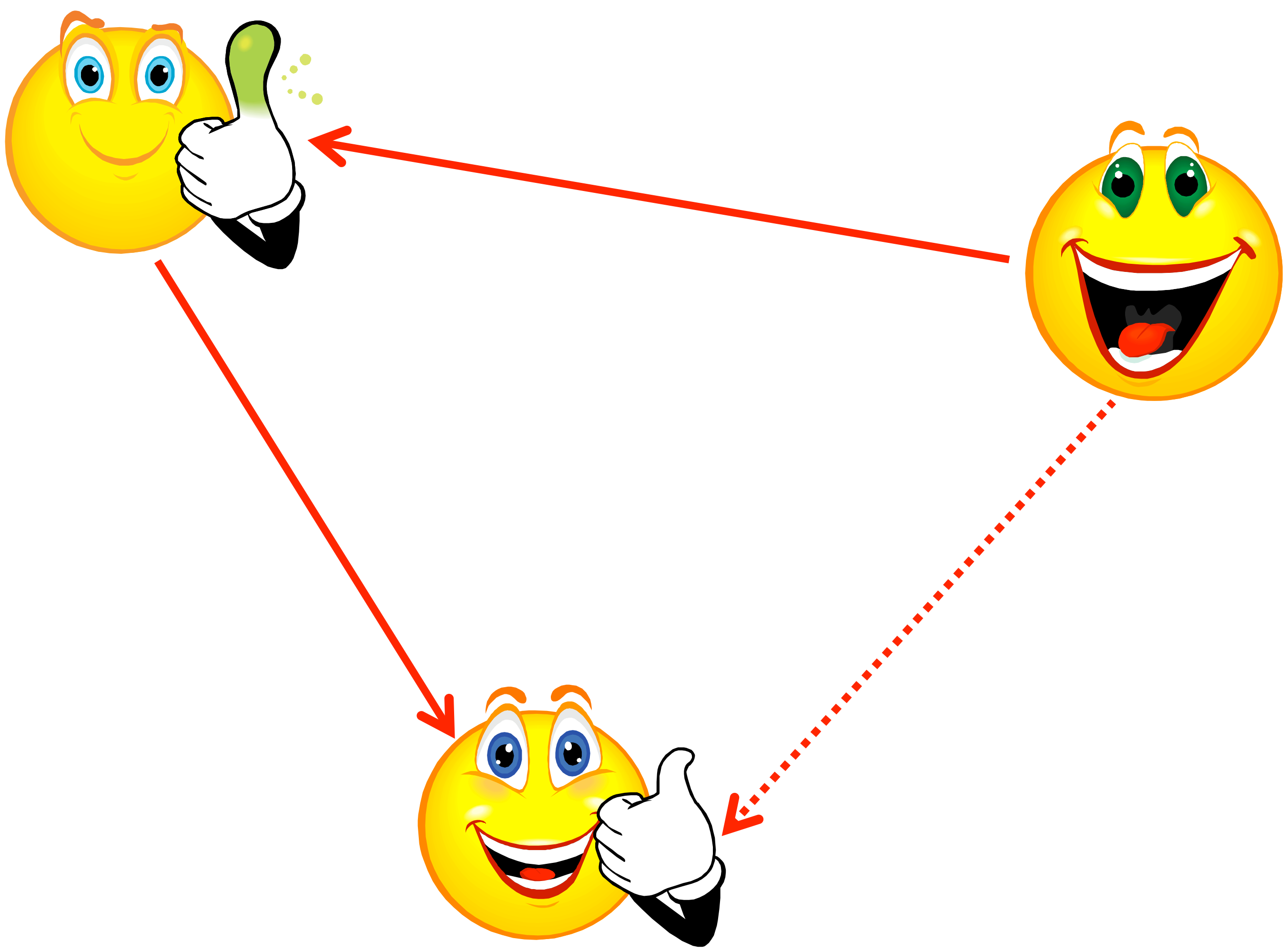
- Users' trust relations can be interpreted as the "similar" relations
    - On the web, user $U_i$ trusts user $U_t$ indicates that user $U_i$ agrees with most of the opinions issued by user $U_t$.

# Trust

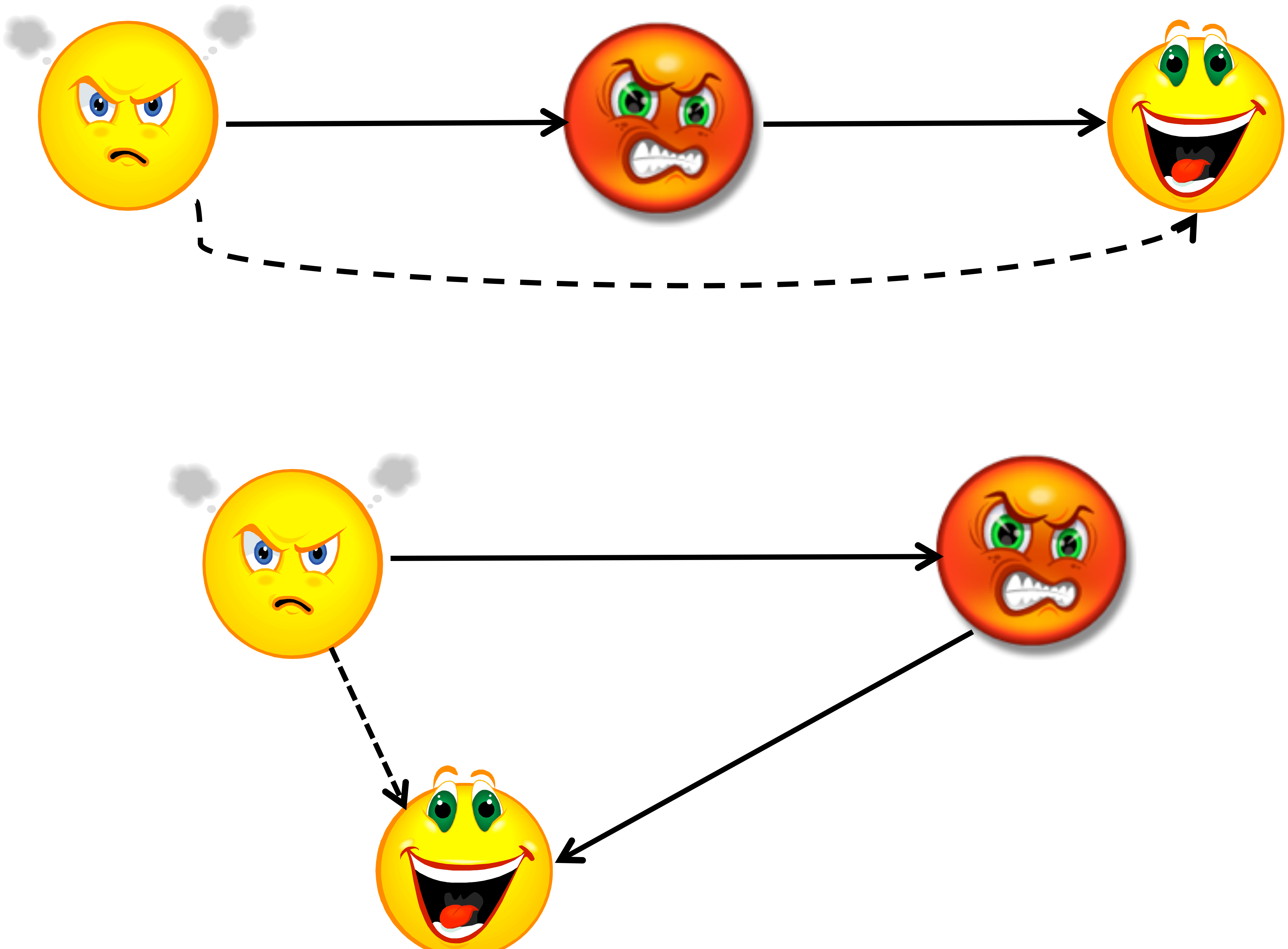
$$\min_{U} \frac{1}{2} \sum_{i=1}^{m} \sum_{t \in \mathcal{T}^+(i)} S_{it}^{\mathcal{T}} \|U_i - U_t\|_F^2$$

$$\begin{aligned} \min_{U,V} \mathcal{L}_{\mathcal{T}}(R, S^{\mathcal{T}}, U, V) &= \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}^{R} (R_{ij} - g(U_i^T V_j))^2 \\ &+ \frac{\alpha}{2} \sum_{i=1}^{m} \sum_{t \in \mathcal{T}^+(i)} (S_{it}^{\mathcal{T}} \|U_i - U_t\|_F^2) \\ &+ \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2. \end{aligned}$$

# Trust Propagation



# Distrust Propagation?



# Experiments

- Dataset - Epinions
- 131,580 users, 755,137 items, 13,430,209 ratings
- 717,129 trust relations, 123,670 distrust relations

# Data Statistics

**Table 1: Statistics of User-Item Rating Matrix of Epinions**

| Statistics | User | Item |
|---|---|---|
| Min. Num. of Ratings | 1 | 1 |
| Max. Num. of Ratings | 162169 | 1179 |
| Avg. Num. of Ratings | 102.07 | 17.79 |

**Table 2: Statistics of Trust Network of Epinions**

| Statistics | Trust per User | Be Trusted per User |
|---|---|---|
| Max. Num. | 2070 | 3338 |
| Avg. Num. | 5.45 | 5.45 |

**Table 3: Statistics of Distrust Network of Epinions**

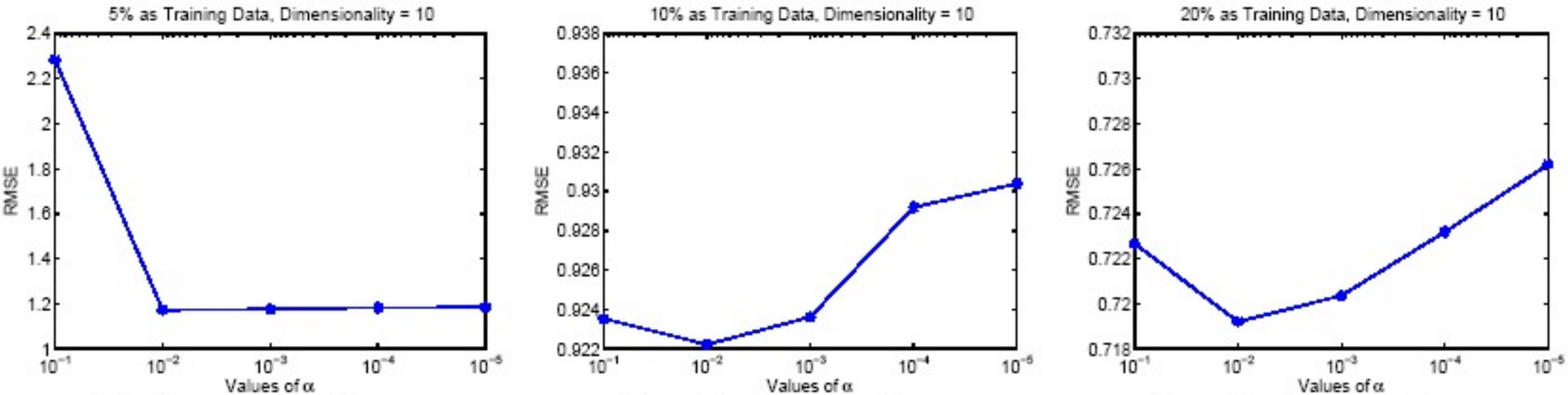
| Statistics | Distrust per User | Be Distrusted per User |
|---|---|---|
| Max. Num. | 1562 | 540 |
| Avg. Num. | 0.94 | 0.94 |

# Experiments

RMSE

| Dataset | Traning Data | Dimensionality | PMF | SoRec | RWD | RWT |
|---|---|---|---|---|---|---|
| Epinions | 5% | 5D | 1.228 | 1.199 | 1.186 | **1.177** |
| | | 10D | 1.214 | 1.198 | 1.185 | **1.176** |
| | 10% | 5D | 0.990 | 0.944 | 0.932 | **0.924** |
| | | 10D | 0.977 | 0.941 | 0.931 | **0.923** |
| | 20% | 5D | 0.819 | 0.788 | 0.723 | **0.721** |
| | | 10D | 0.818 | 0.787 | 0.723 | **0.720** |

# Impact of Parameters




(a) 5% as Training Data (b) 10% as Training Data (c) 20% as Training Data

**Figure 6: Impact of Parameter $\alpha$**

Alpha = 0.01 will get the best performance!
Parameter beta basically shares the same trend!

# Social Recommender Systems

- Introduction
- Collaborative Filtering
- Trust-aware Recommender Systems
- Social-based Recommender Systems

# Comparison

- Trust-aware Recommender systems
  - Trust network
  - Trust relations can be treated as "similar" relations
  - Few dataset available on the web

- Social-based Recommender Systems
  - Social friend network, mutual relations
  - Friends are very divers, and may have different tastes
  - Lots of web sites have social network implementation

# References


- J. Basilico and T. Hofmann. Unifying collaborative and content-based filtering. In ICML, 2004.
- J. S. Breese, D. Heckerman, and C. M. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In UAI, pages 43–52, 1998.
- M. Deshpande and G. Karypis. Item-based top-N recommendation algorithms. ACM Trans. Inf. Syst., 22(1):143–177, 2004.
- J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In SIGIR, pages 230–237. ACM, 1999.
- J. L. Herlocker, J. A. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Inf. Retr., 5(4):287–310, 2002.
- G. Linden, B. Smith, and J. York. Industry report: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Distributed Systems Online, 4(1), 2003.

# References


- H. Ma, I. King, and M. R. Lyu. Effective missing data prediction for collaborative filtering. In SIGIR, pages 39–46, 2007.
- H. Ma, H. Yang, M. R. Lyu, and I. King. SoRec: social recommendation using probabilistic matrix factorization. In CIKM, pages 931–940, 2008.
- H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In SIGIR, pages 203-210, 2009.
- H. Ma, M. R. Lyu, and I. King. Learning to recommend with trust and distrust relationships. In RecSys, pages 189-196, 2009.
- B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In WWW, pages 285–295, 2001.
- R. Salakhutdinov, and A. Mnih. Probabilistic Matrix Factorization. In NIPS, 2007.
- D. D. Lee, and H. S. Seung. Algorithms for Non-negative Matrix Factorization. In NIPS, pages 556-562, 2000.

# Outline

- Introduction
- Social Search Engine
- Social Recommender Systems
- Social Media Analysis

# Social Media Analysis

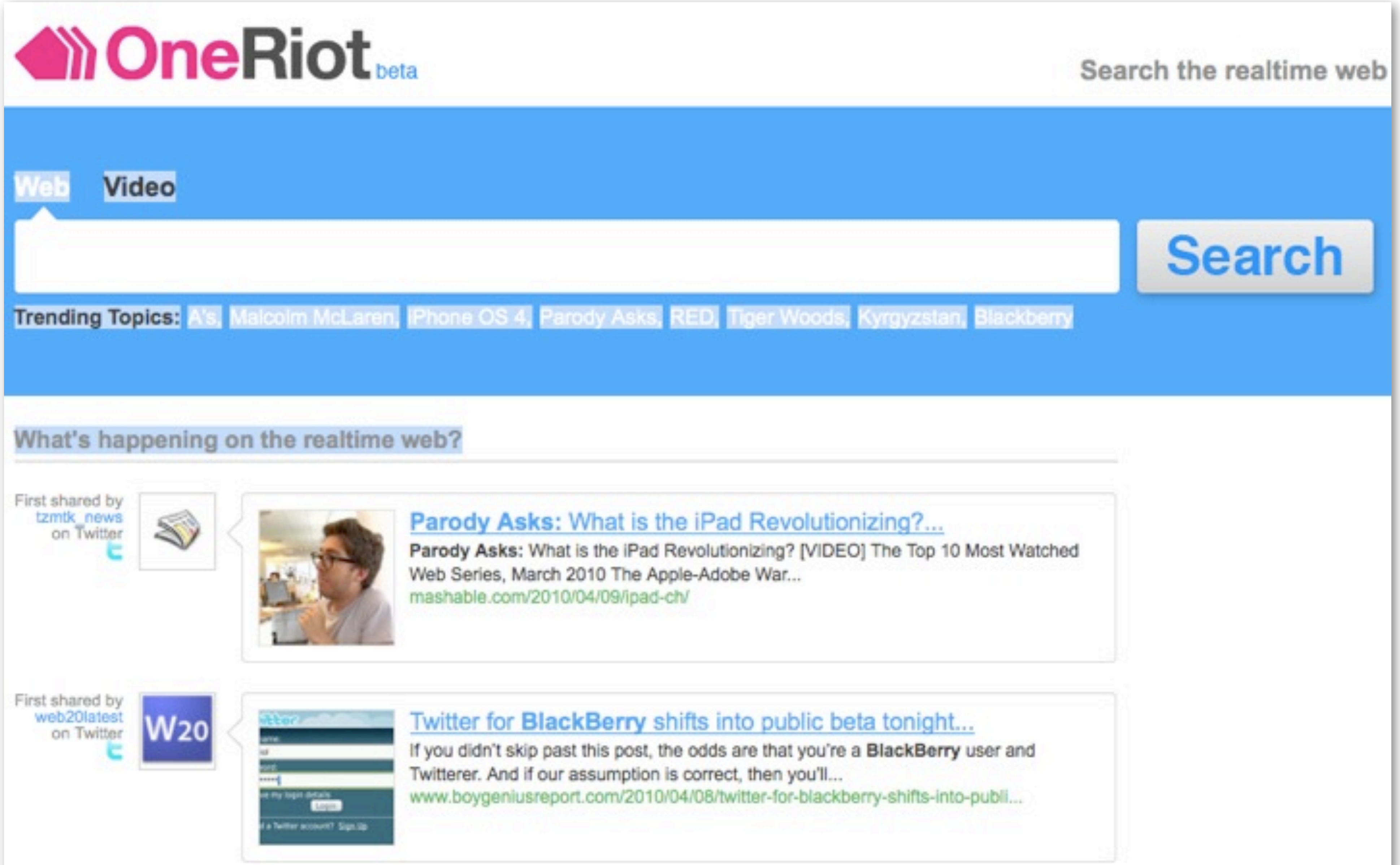
- Social Media Ranking
- Tag Recommendation
- News Recommendation
- User Recommendation
- Twitter-powered Recommendation

# Social Media Ranking

- Pulse Rank - OneRiot
- Reddit Algorithm
- Digg Algorithm
- Google's Page Rank

# Pulse Rank - OneRiot

- A realtime web search engine, which archives and makes searchable news, videos and blogs being discussed on the web, ordered to reflect current social relevance.



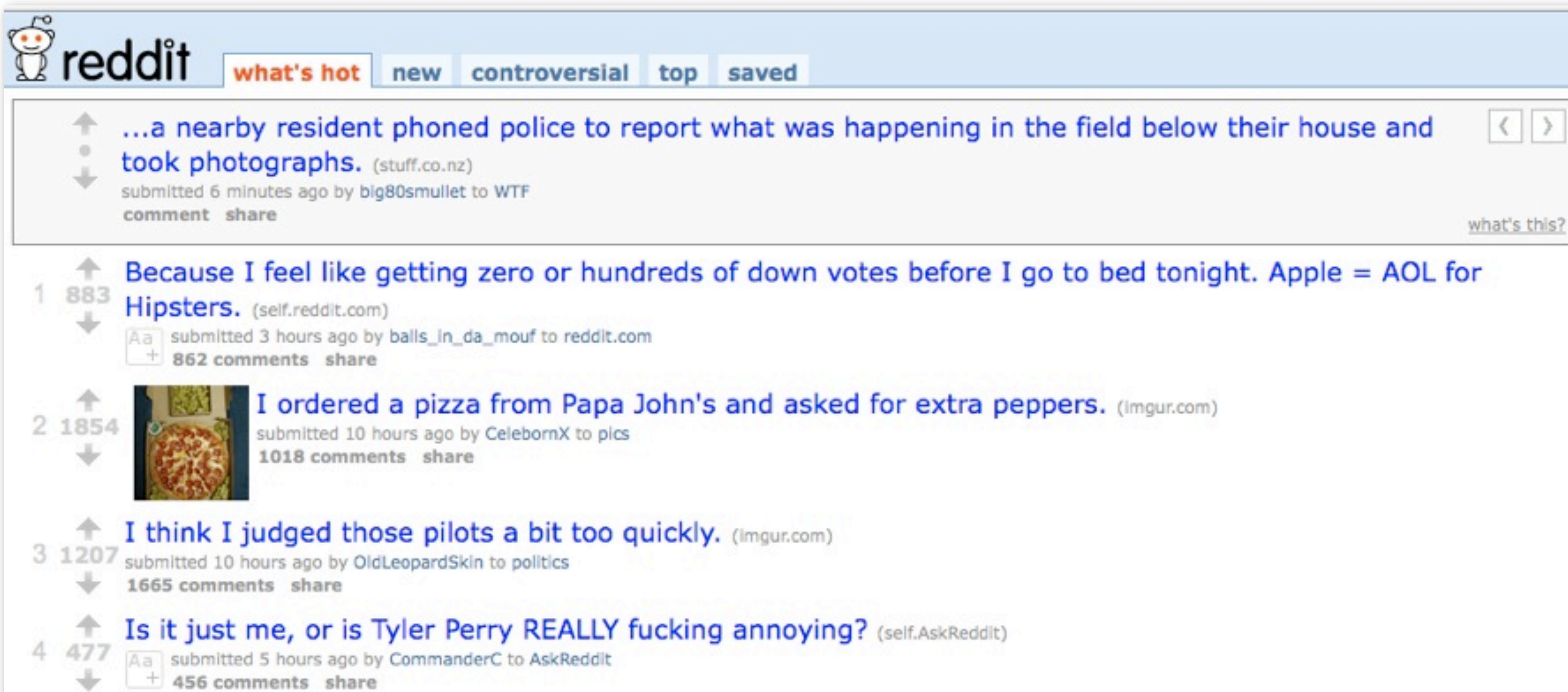
# Pulse Rank - OneRiot

- "Pulse Rank" algorithm looks at dozens of factors that give "weight" to certain results
  - **Freshness**: Is the most recently published content necessarily the most relevant?
  - **Domain Authority:** An article about Obama on New York Times should weight higher than the article on my blog.
  - **People Authority**: Who is sharing this link on the social web?
  - **Acceleration**: Is this page increasing in hotness or decreasing in hotness?

From http://blog.oneriot.com/content/2009/06/oneriot-pulse-rank/

# Reddit Algorithm

- **Reddit** is a social news website on which users can post links to content on the Internet. Other users may then vote the posted links up or down, causing them to become more or less prominent on the reddit home page.



# Reddit Algorithm

- Time differences

$$t_s = A - B$$

- Differences of the up votes and down votes

$$x = U - D$$

$$y = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases} \qquad z = \begin{cases} |x| & \text{if } |x| \geq 1 \\ 1 & \text{if } x < 1 \end{cases}$$

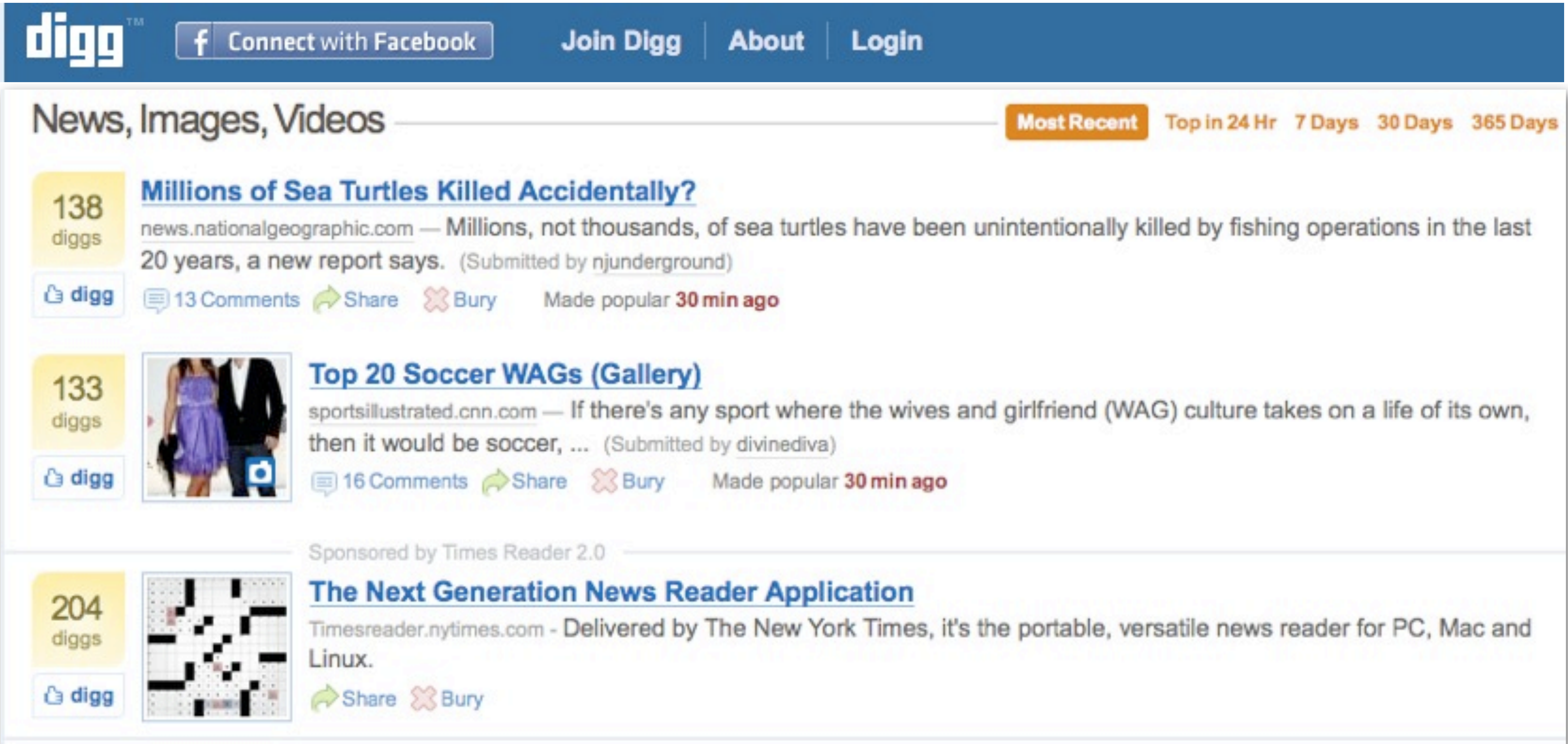
- Ranking functions

$$f(t_s, y, z) = \log_{10} z + \frac{y t_s}{45000}$$

From http://uggedal.com/reddit.cf.algorithm.png

# Digg Algorithm

- A social news website made for people to discover and share content from anywhere on the Internet, by submitting links and stories, and voting and commenting on submitted links and stories

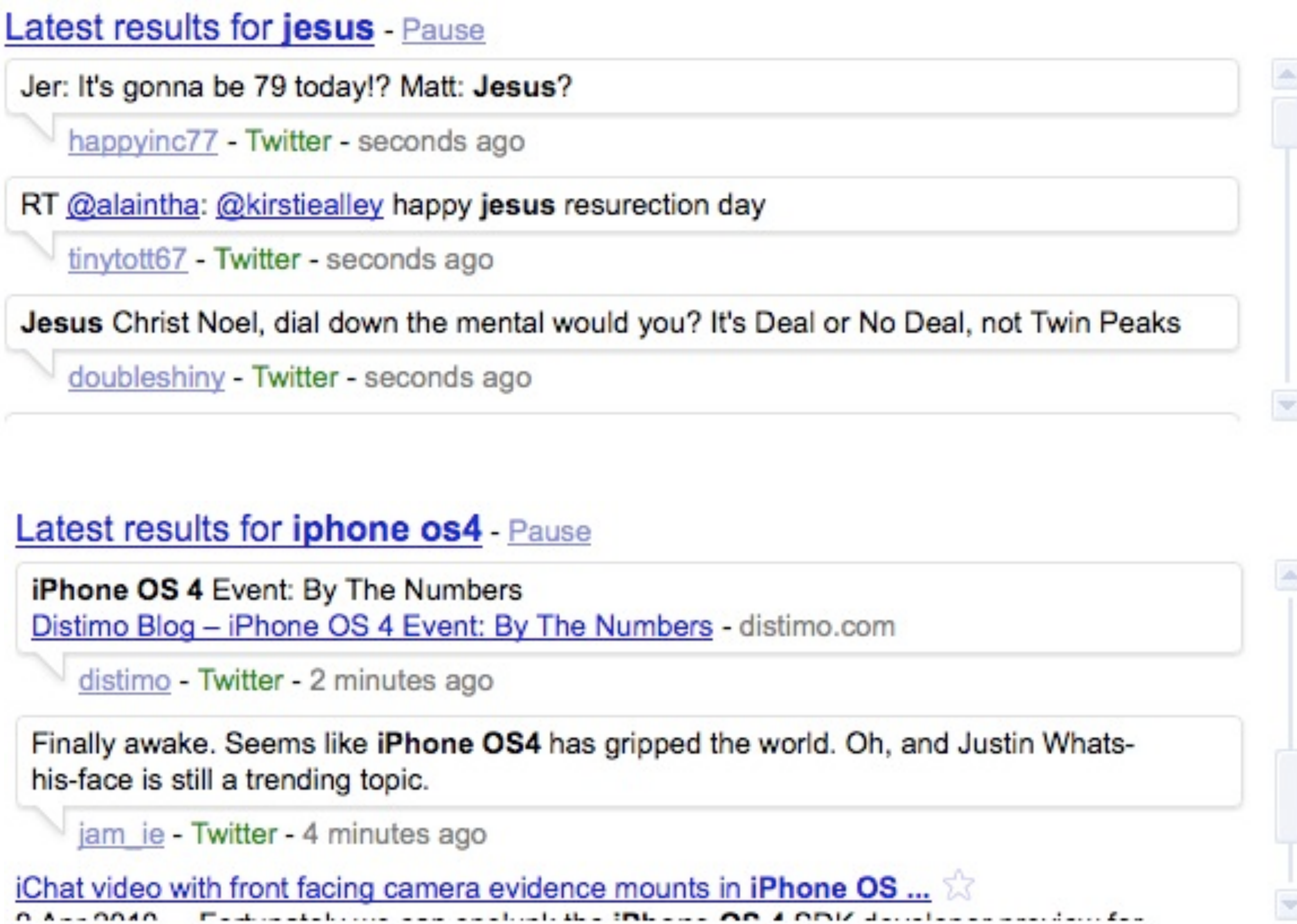


# Digg Algorithm

- The rapidity of the votes
  If you get 40-50 votes (no matter what users digg) in the first 30 minutes, you're probably on the frontpage.
- The rank of the users that vote the article
  The highest it is on the top list, the better.
- The number of comments, and the positive diggs that each article receives
  If you have a lot of negative rated comments that can hurt more then help actually.
- The number of buries your story gets
- The submitted / promoted stories ratio of the users that vote
  If 12-14 users with at least a 70% ratio, vote your article, you can make the frontpage much easier.

From http://www.seopedia.org/tips-tricks/social-media/the-digg-algorithm-unofficial-faq/

# How Google Ranks Tweets



# How Google Ranks Tweets

- The key is to identify "reputed followers"
- You earn reputation, and then you give reputation
- One user following another in social media is analogous to one page linking to another on the Web. Both are a form of recommendation
- Page Rank on follow graph

From http://www.technologyreview.com/web/24353/?a=f

# Social Media Analysis

- Social Media Ranking
- Tag Recommendation
- News Recommendation
- User Recommendation
- Twitter-powered Recommendation

# Why Users Tag?

- Tagging means something specific to the user
- It is easy -- anyone can do it
- Finding things on the Internet
- Serendipitous discovery
- It is social
- New ways to share and discover

# Why need Tag Recommendation?

- User tags contain noises
- Automating the tagging process
- Assisting users to tag

# Flickr Tag Recommendation based on Collective Knowledge

[B. Sigurbjörnsson, et al., WWW2008]

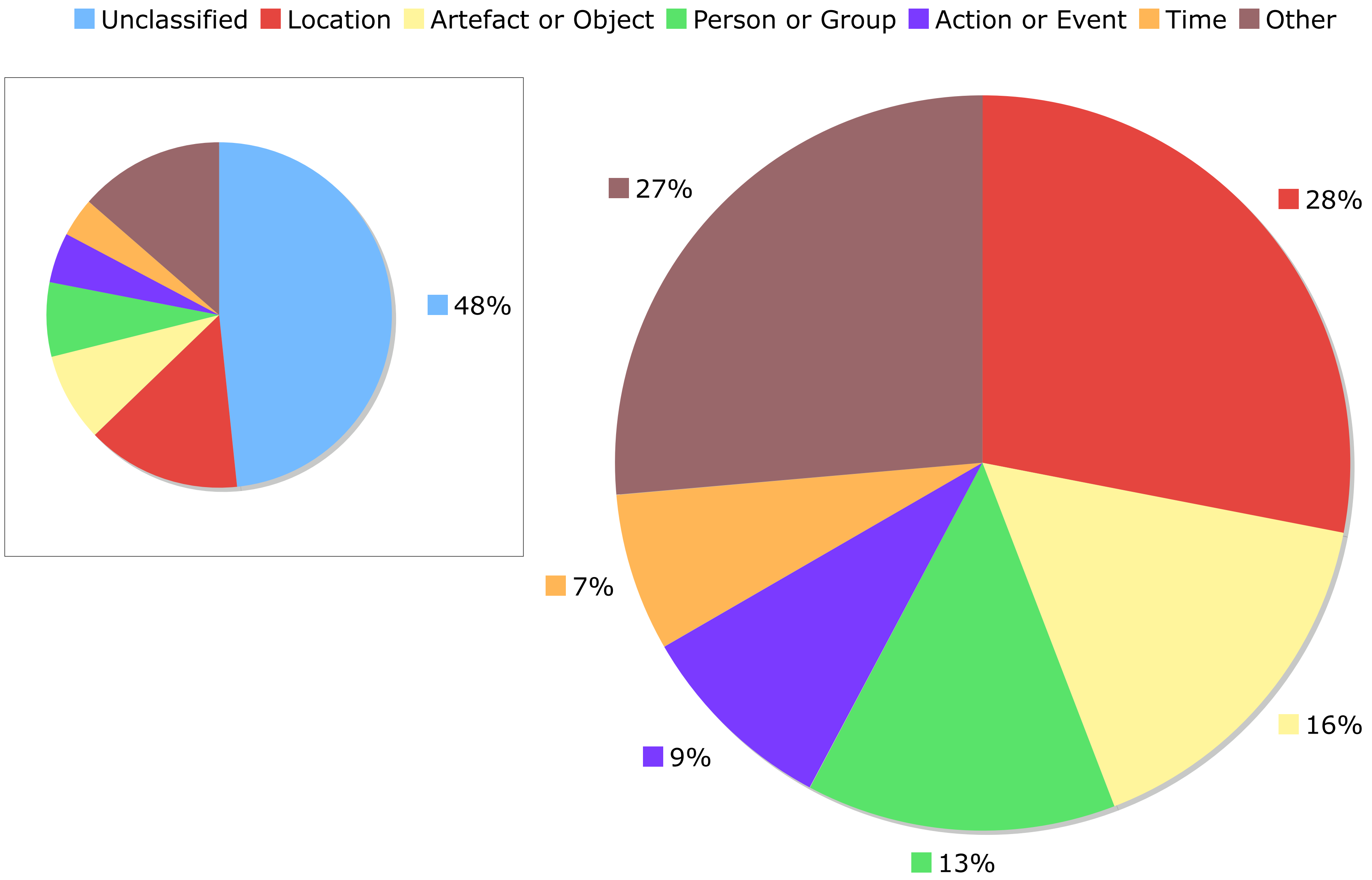



**Figure 3: Most frequent WordNet categories for Flickr tags.**

# Flickr Tag Recommendation based on Collective Knowledge

[B. Sigurbjörnsson, et al., WWW2008]

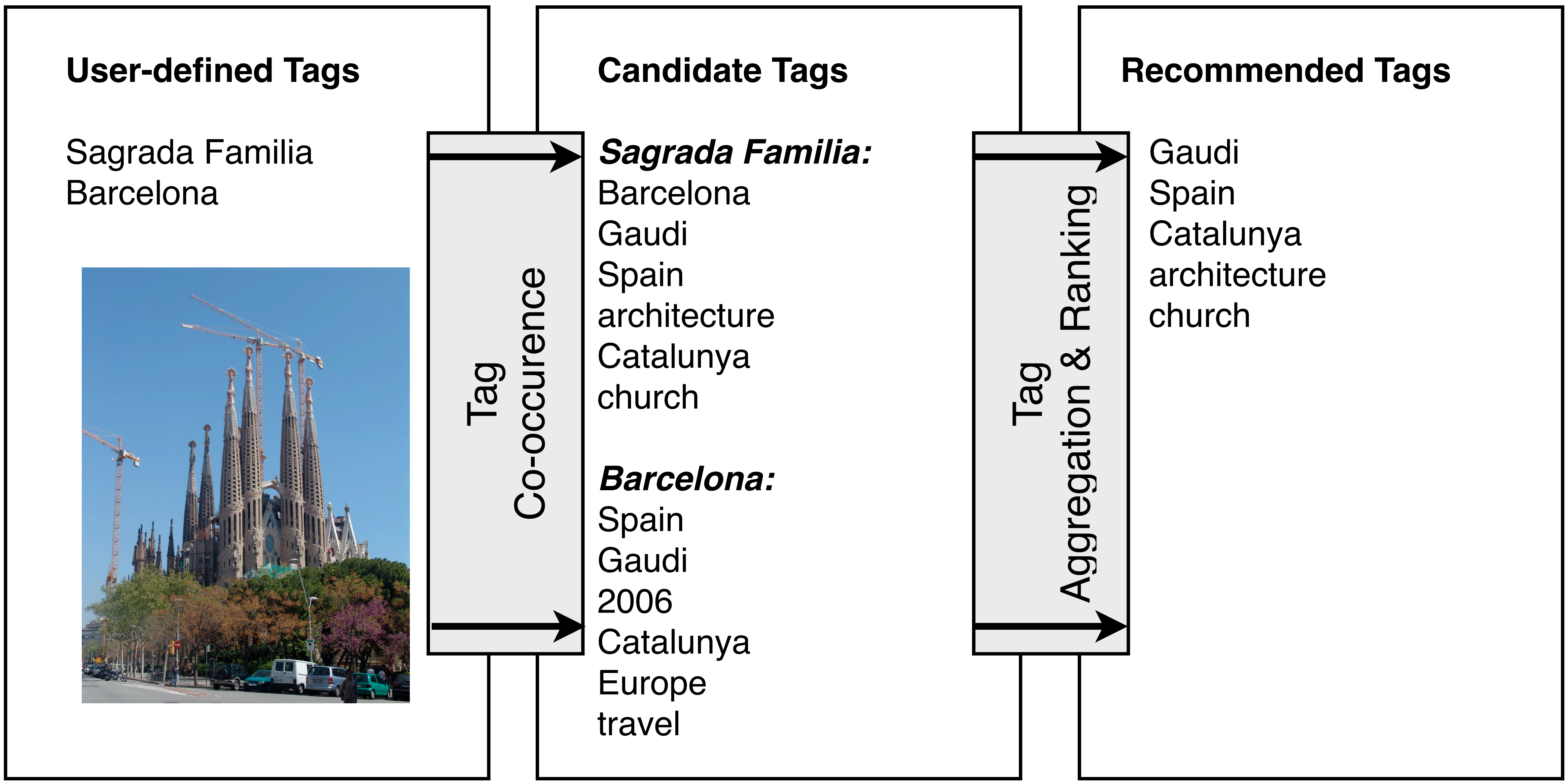



Figure 4: System overview of the tag recommendation process.

# Flickr Tag Recommendation based on Collective Knowledge

[B. Sigurbjörnsson, et al., WWW2008]


- Define the Tag Co-occurrence between two tags to be the number of photos where both tags are used in the same annotation

- Symmetric measure: Jaccard Coefficient

$$J(t_i, t_j) := \frac{|t_i \cap t_j|}{|t_i \cup t_j|}$$

- Asymmetric measure:

$$P(t_j|t_i) := \frac{|t_i \cap t_j|}{|t_i|}$$

# Flickr Tag Recommendation based on Collective Knowledge

[B. Sigurbjörnsson, et al., WWW2008]

Tag: Eiffel Tower

Symmetric Measure:

Tour Eiffel
Eiffel
Seine
La Tour Eiffel
Paris

Good at identifying equivalent tags

Aymmetric Measure:

Paris
France
Tour Eiffel
Eiffel
Europe

Good at suggesting diverse tags

# Flickr Tag Recommendation based on Collective Knowledge

[B. Sigurbjörnsson, et al., WWW2008]

- Aggregation


  - Vote
    - The voting strategy computes a score for each candidate tag *c*

$$vote(u,c) = \begin{cases} 1 & \text{if } c \in C_u \\ 0 & \text{otherwise} \end{cases}$$

A score is therefore computed as

$$score(c) := \sum_{u \in U} vote(u,c)$$

  - Sum
    - The summing strategy sums over the co-occurrence values of the tags

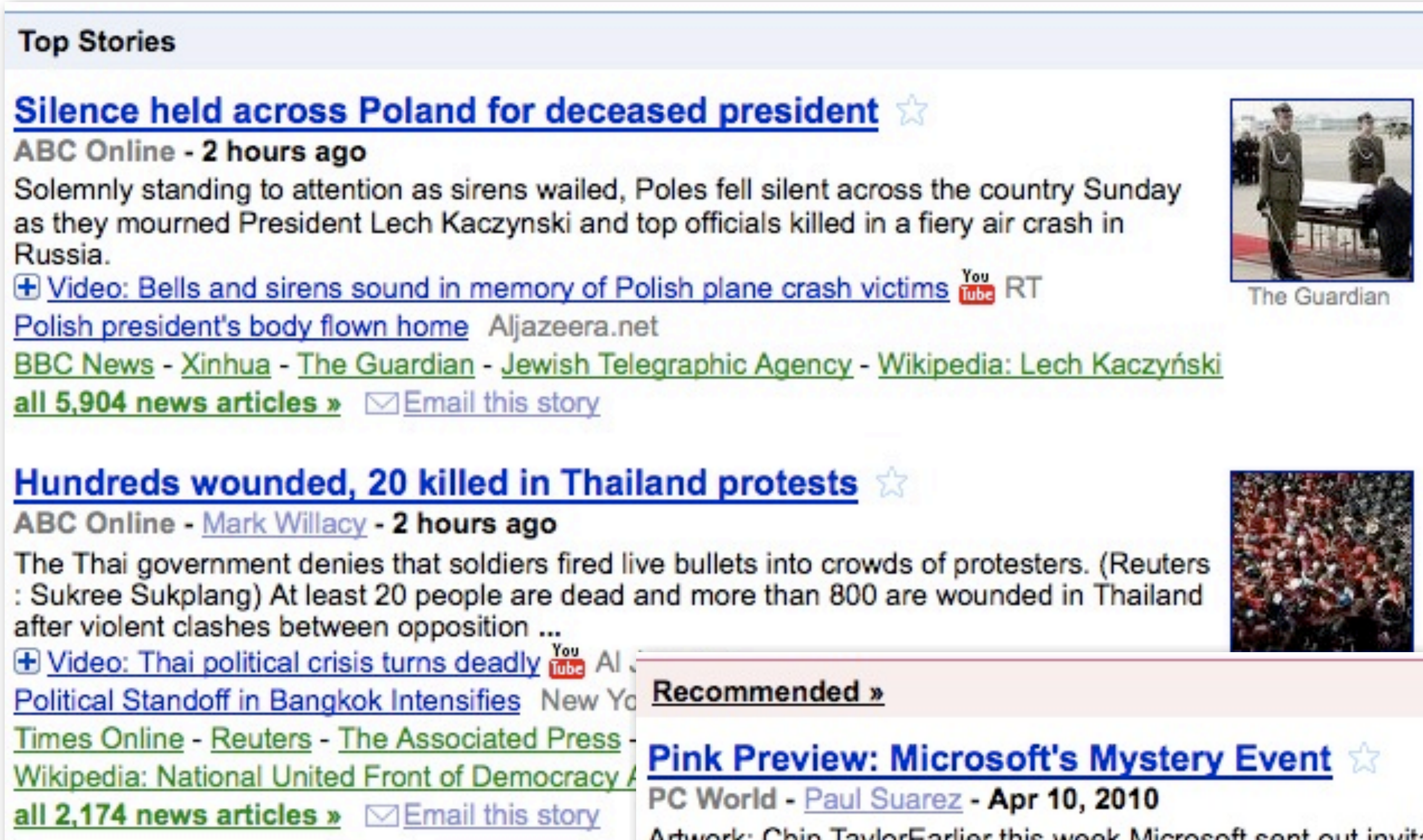
$$score(c) := \sum_{u \in U} (P(c|u) \quad \text{, if } c \in C_u)$$

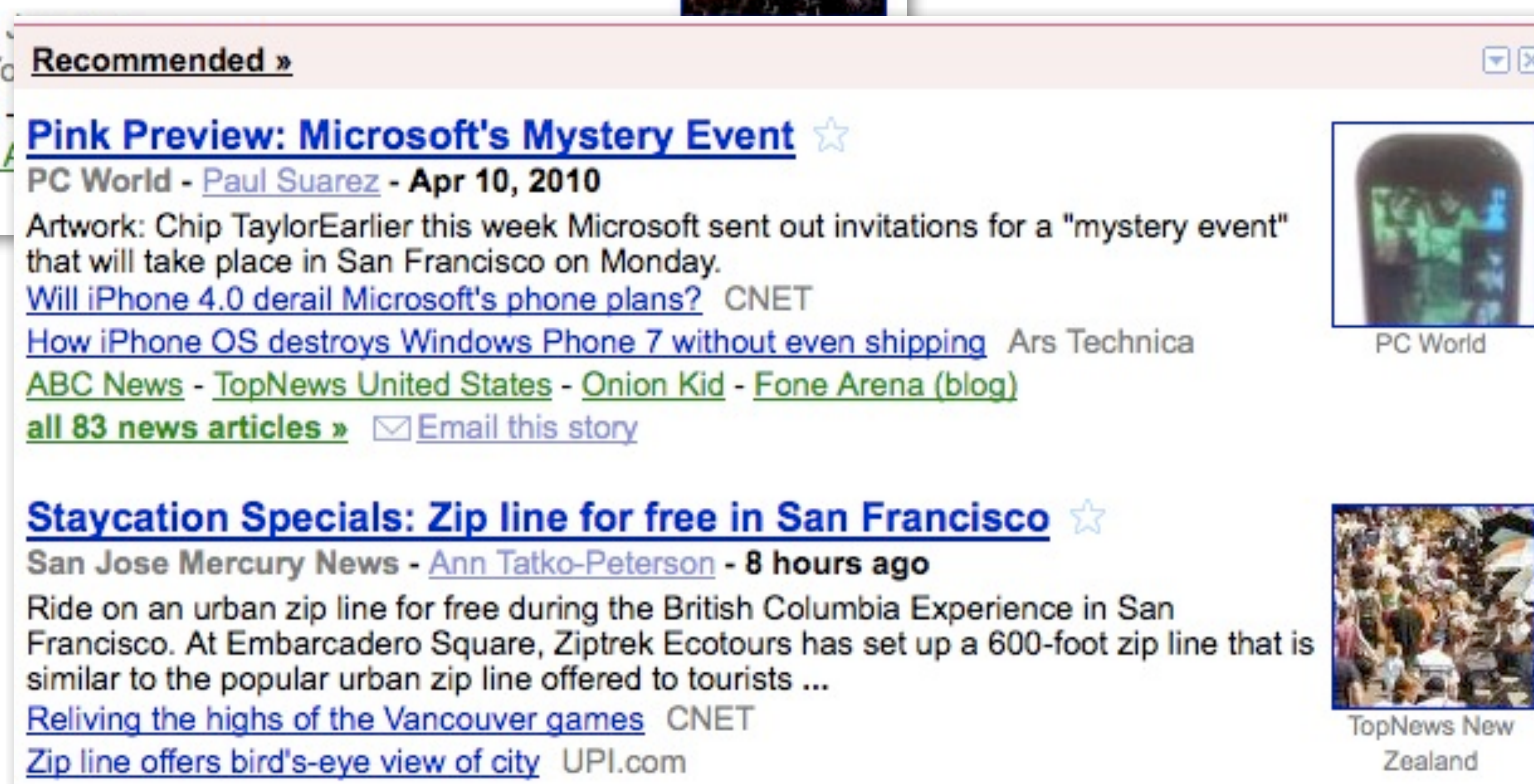
where *P(c|u)* calculates the asymmetric co-occurrence values, and *u* is the user defined tags

# Social Media Analysis

- Social Media Ranking
- Tag Recommendation
- News Recommendation
- User Recommendation
- Twitter-powered Recommendation

# Google News Recommendation



# News Recommendation

- Online news reading has become very popular
- Web provides access to news articles from millions of sources around the world
- Key challenge: help users find the articles that are interesting to read

# Personalized News Recommendation Based on Click Behavior

[J. Liu, et al., IUI2008]

- News click logs analysis
  - Data
    - Google News, over 12-month period, from 2007/07/01 to 2008/06/30
    - Randomly sampled 16,848 users from users who made at least 10 clicks per month
    - Users are from more than 10 different countries and regions

# Personalized News Recommendation Based on Click Behavior

[J. Liu, et al., IUI2008]

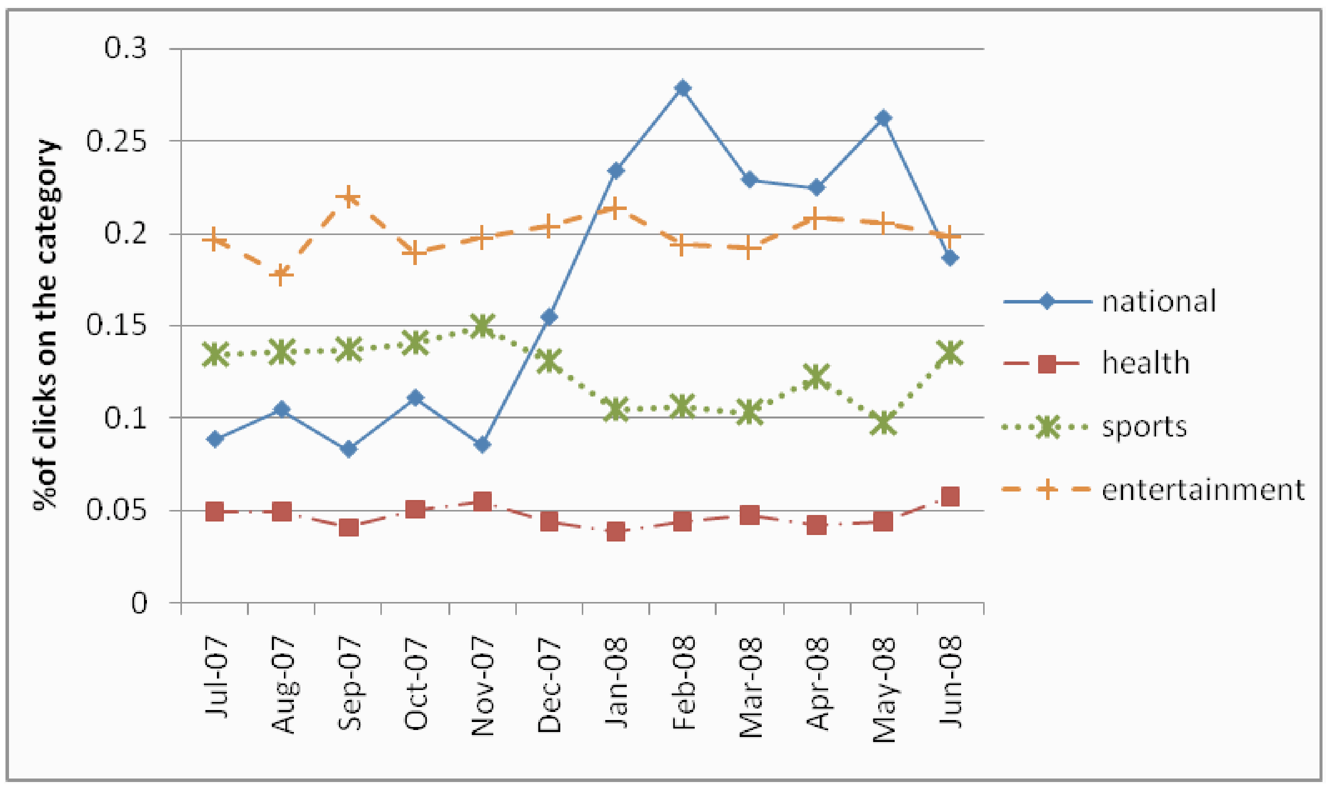



**Figure 2. Interest distribution of US users over time**

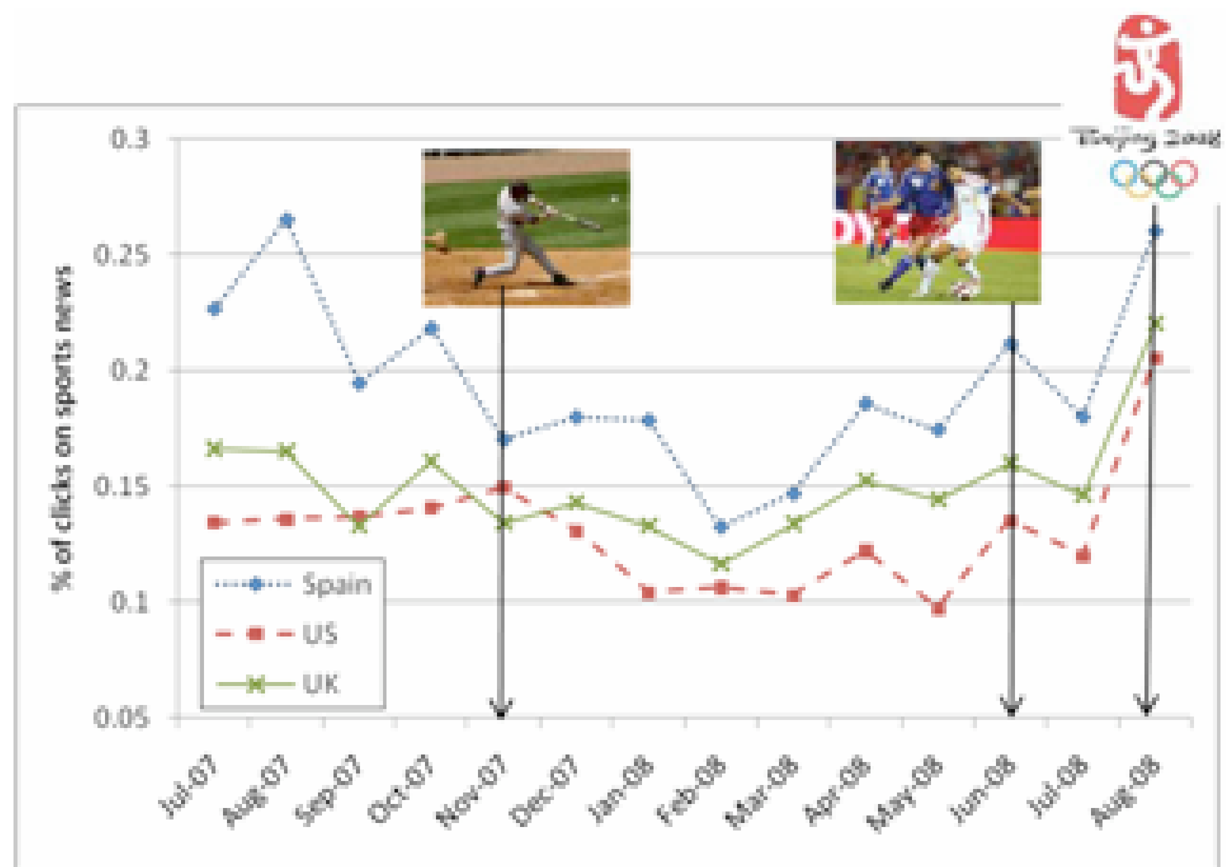



**Figure 3. Change of interests in sports news over time**

# Personalized News Recommendation Based on Click Behavior

[J. Liu, et al., IUI2008]

- Observations
  - The news interests of individual users do change over time
  - The click distributions of the general public reflect the news trend, which correspond to the big news events
  - There exists different news trends in different locations
  - To a certain extent, the individual user's news interests correspond with the news trend in the location that the users belongs to

# Personalized News Recommendation Based on Click Behavior

[J. Liu, et al., IUI2008]

- Bayesian Framework for User Interest Prediction
  - Predicting user's genuine news interest
    - For a specific time period t in the past, the genuine interest of a user in topic category $c_i$ is modeled as

$$p^t(click \mid category = c_i)$$

    - Using Bayesian rule

$$interest^t(category = c_i) = p^t(click \mid category = c_i)$$

$$= \frac{p^t(category = c_i \mid click)\, p^t(click)}{p^t(category = c_i)}$$

# Personalized News Recommendation Based on Click Behavior

[J. Liu, et al., IUI2008]

- Bayesian Framework for User Interest Prediction
  - Combining predictions of past time periods

$$interest(category = c_i) = \frac{\sum_t \left(N^t \times interest^t(category = c_i)\right)}{\sum_t N^t}$$

$$= \frac{\sum_t \left( N^t \times \frac{p^t(category = c_i \mid click) p^t(click)}{p^t(category = c_i)} \right)}{\sum_t N^t}$$

$N^t$ is the total number of clicks by the user in time period *t*

  - Assume $p^t(click)$ is a constant, then we get

$$interest(category = c_i)$$

$$= \frac{p(click) \times \sum_t \left( N^t \times \frac{p^t(category = c_i \mid click)}{p^t(category = c_i)} \right)}{\sum_t N^t}$$

# Personalized News Recommendation Based on Click Behavior

[J. Liu, et al., IUI2008]

- Bayesian Framework for User Interest Prediction
  - Predicting user's current news interest
    - Use the click distribution of the general public in a short current time period (e.g. in the past hour), represented as $p^0(category = c_i)$, by using Bayesian rule:

$$p^0(category = c_i \mid click) = \frac{p^0(click \mid category = c_i)\, p^0(category = c_i)}{p^0(click)}$$

    - Estimate $p^0(click \mid category = c_i)$ with genuine interests $interest(category = c_i)$

$$p^0(category = c_i \mid click) \propto \frac{interest(category = c_i)\, p^0(category = c_i)}{p(click)} \propto \frac{p^0(category = c_i) \times \sum_t \left( N^t \times \frac{p^t(category = c_i \mid click)}{p^t(category = c_i)} \right)}{\sum_t N^t}$$

# Personalized News Recommendation Based on Click Behavior

[J. Liu, et al., IUI2008]

- Bayesian Framework for User Interest Prediction
  - Predicting user's current news interest
    - Adding a set of virtual clicks *G*, which is set to be 10 in the system. It can be regarded as a smooth factor.

$$p^0(category = c_i \mid click)$$

$$\propto \frac{p^0(category = c_i) \times \left( \sum_t \left( N^t \times \frac{p^t(category = c_i \mid click)}{p^t(category = c_i)} \right) + G \right)}{\sum_t N^t + G}$$

# Personalized News Recommendation Based on Click Behavior

[J. Liu, et al., IUI2008]

- Live traffic experiment
  - Experiments conducted on a fraction (about 10,000 users) of the live traffic at Google News
  - Users were randomly assigned to a control group and a test group. Two groups have the same size
  - Control group uses old recommendation algorithm, while the test group uses the proposed recommendation algorithm

# Personalized News Recommendation Based on Click Behavior

[J. Liu, et al., IUI2008]

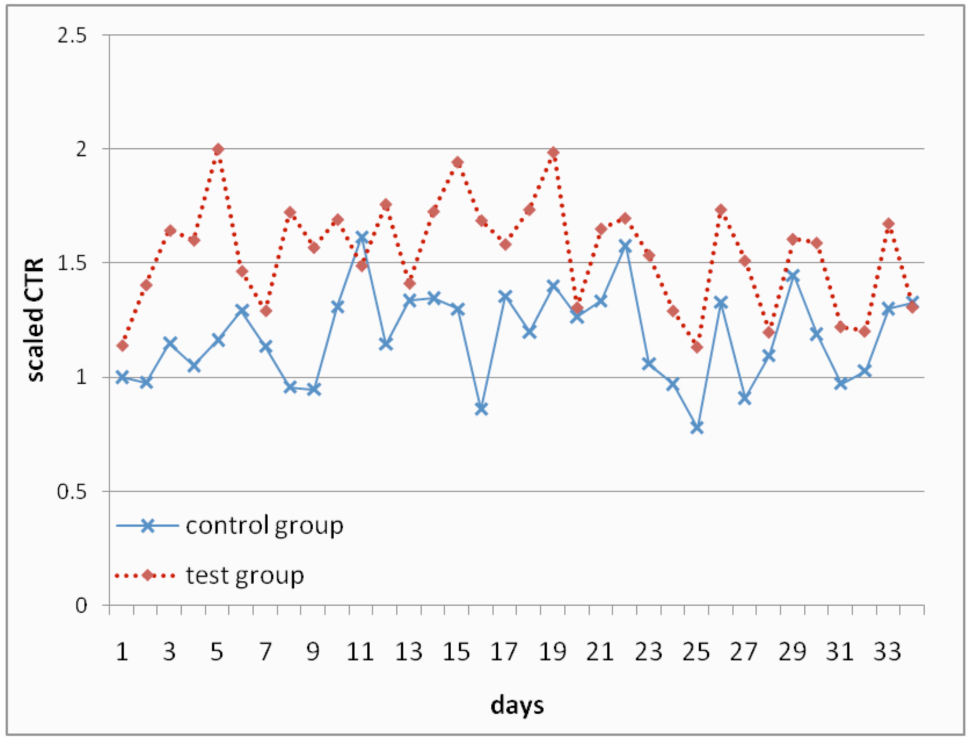



**Figure 4. CTR of the recommended news section**

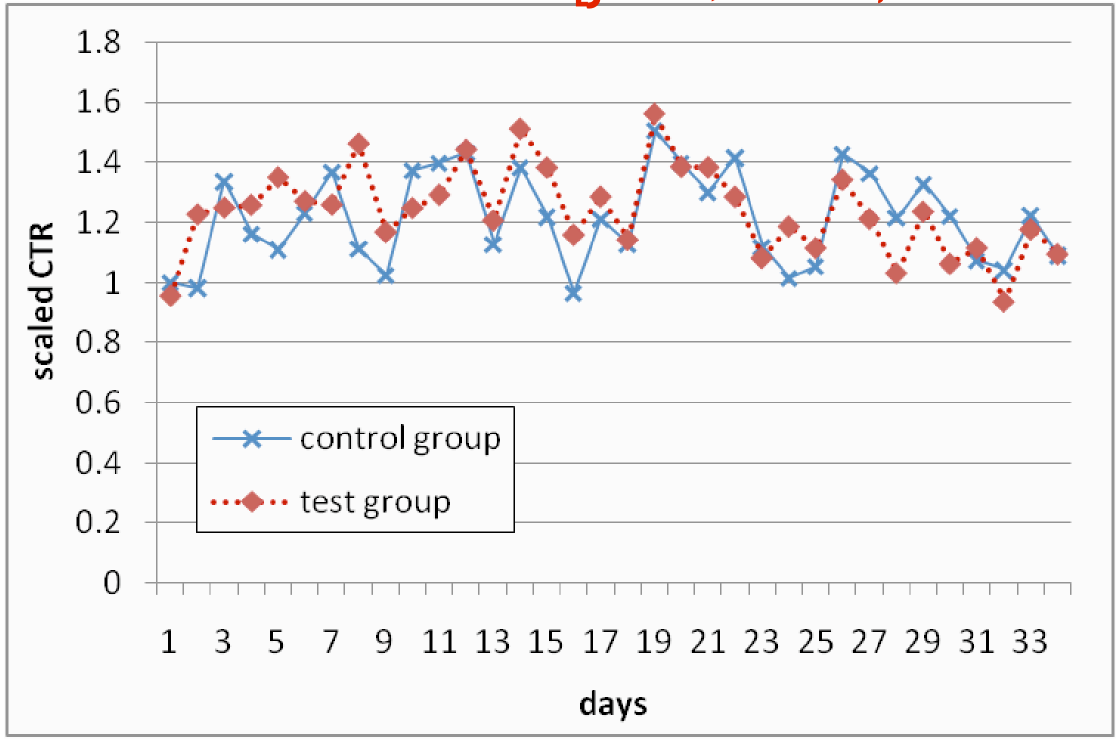



**Figure 5. CTR of the Google News homepage**

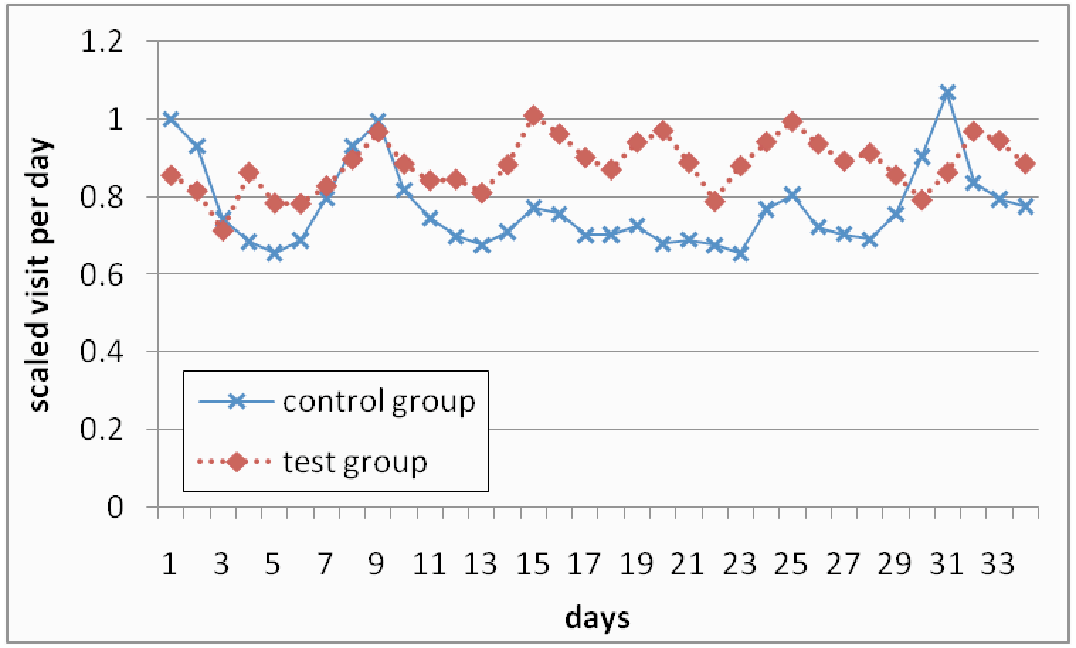


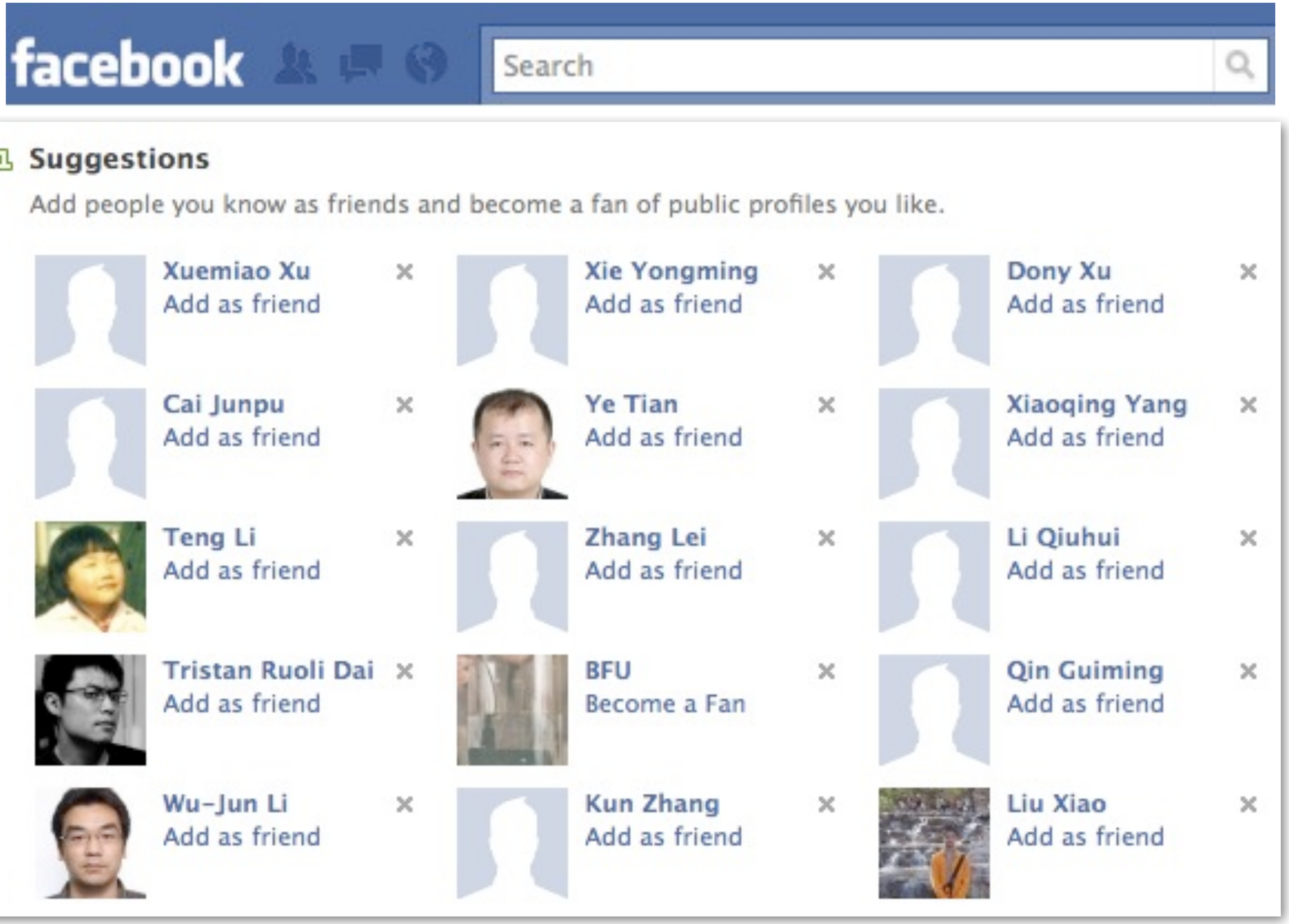

**Figure 6. Frequency of website visit per day**

# Social Media Analysis

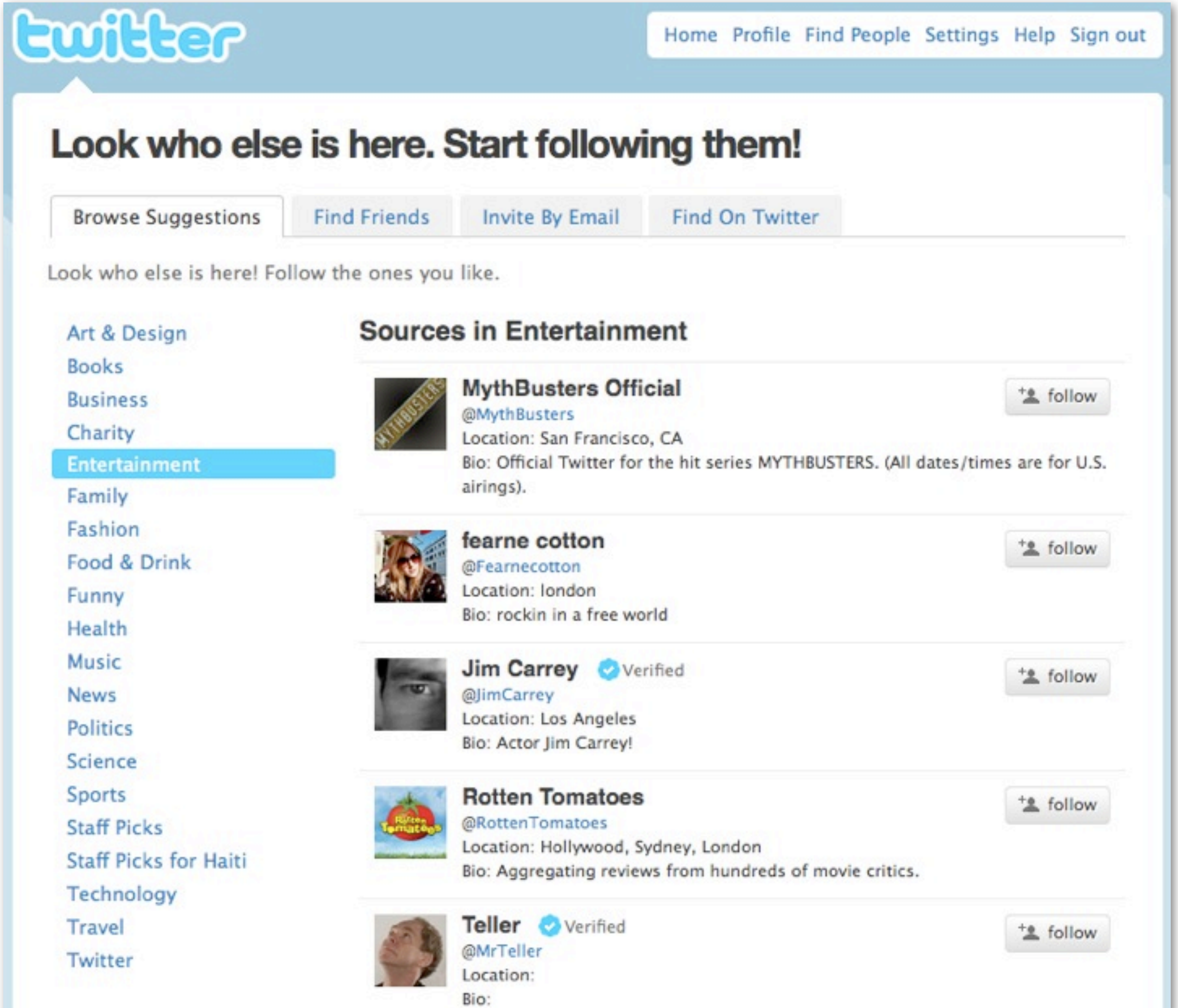
- Social Media Ranking
- Tag Recommendation
- News Recommendation
- User Recommendation
- Twitter-powered Recommendation

# User Recommendation

- Facebook Service - People You May Know
  - Based on "friend-of-a-friend" approach



# User Recommendation



# "Make New Friends, but Keep the Old" - Recommending People on Social Networking Sites

[Jilin Chen, et al., CHI2009]

- On social networking sites, people recommendation algorithms are designed to help users:
  - Find known, offline contacts
  - Discover new friends
- Both are challenging problems

# "Make New Friends, but Keep the Old" - Recommending People on Social Networking Sites

[Jilin Chen, et al., CHI2009]

- Two research questions:
  - How effective are different algorithms in recommending people as potential friends?
  - Can a people recommender system effectively increase the number of friends a user has?

# "Make New Friends, but Keep the Old" - Recommending People on Social Networking Sites

[Jilin Chen, et al., CHI2009]

- Test bed
  - Beehive, an enterprise social networking site within IBM
- Four different algorithms are tested
- The survey was targeted at a group of 500 users who were asked to answer questions related to their friending behavior

# “Make New Friends, but Keep the Old” - Recommending People on Social Networking Sites

[Jilin Chen, et al., CHI2009]

- Algorithms

1. Content Matching
   - Based on the intuition that “if we both post content on similar topics, we might be interested in getting to know each other”
   - Based on TFxIDF method
2. Content-plus-Link (CplusL)
   - Enhances the content matching algorithm with social link information derived from social network structure
   - Based on the intuition that “By disclosing a network path to a weak tie or unknown person, the recipient will be more likely to accept the recommendation.”

# "Make New Friends, but Keep the Old" - Recommending People on Social Networking Sites

[Jilin Chen, et al., CHI2009]

- Algorithms

3. Friend-of-Friend (FoF)
   - Only leverages social network information of friending
   - Based on the intuition that "if many of my friends consider Alice a friend, perhaps Alice could be my friend too"
4. SONAR
   - Based on the SONAR system, which aggregates social relationship information from different public data sources within IBM: (1) Organizational chart; (2) Publication database; (3) Patent database; (4) Friending system; (5) People tagging system; (6) Project wiki; and (7) Blogging system.

# “Make New Friends, but Keep the Old” - Recommending People on Social Networking Sites

[Jilin Chen, et al., CHI2009]

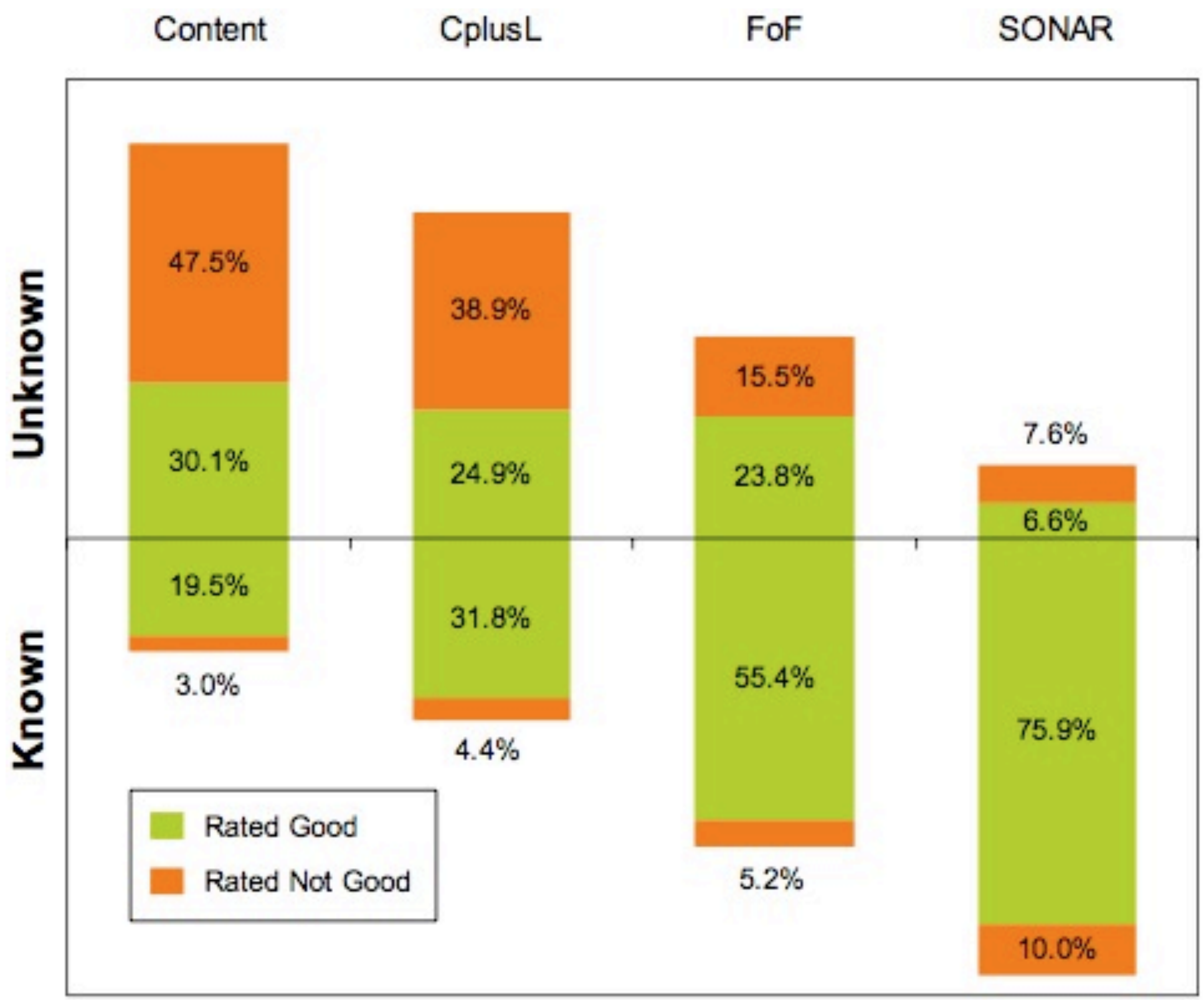



**Figure 1. Known vs. unknown, Good vs. not good.**

# "Make New Friends, but Keep the Old" - Recommending People on Social Networking Sites

[Jilin Chen, et al., CHI2009]

| | Content | CplusL | FoF | SONAR |
|---|---|---|---|---|
| Content | | 52.8% | 1.8% | 8.3% |
| CplusL | | | 3.3% | 9.6% |
| FoF | | | | 13.1% |

**Table 1. Overlap ratios between recommendations generated by different algorithms.**

# “Make New Friends, but Keep the Old” - Recommending People on Social Networking Sites

[Jilin Chen, et al., CHI2009]

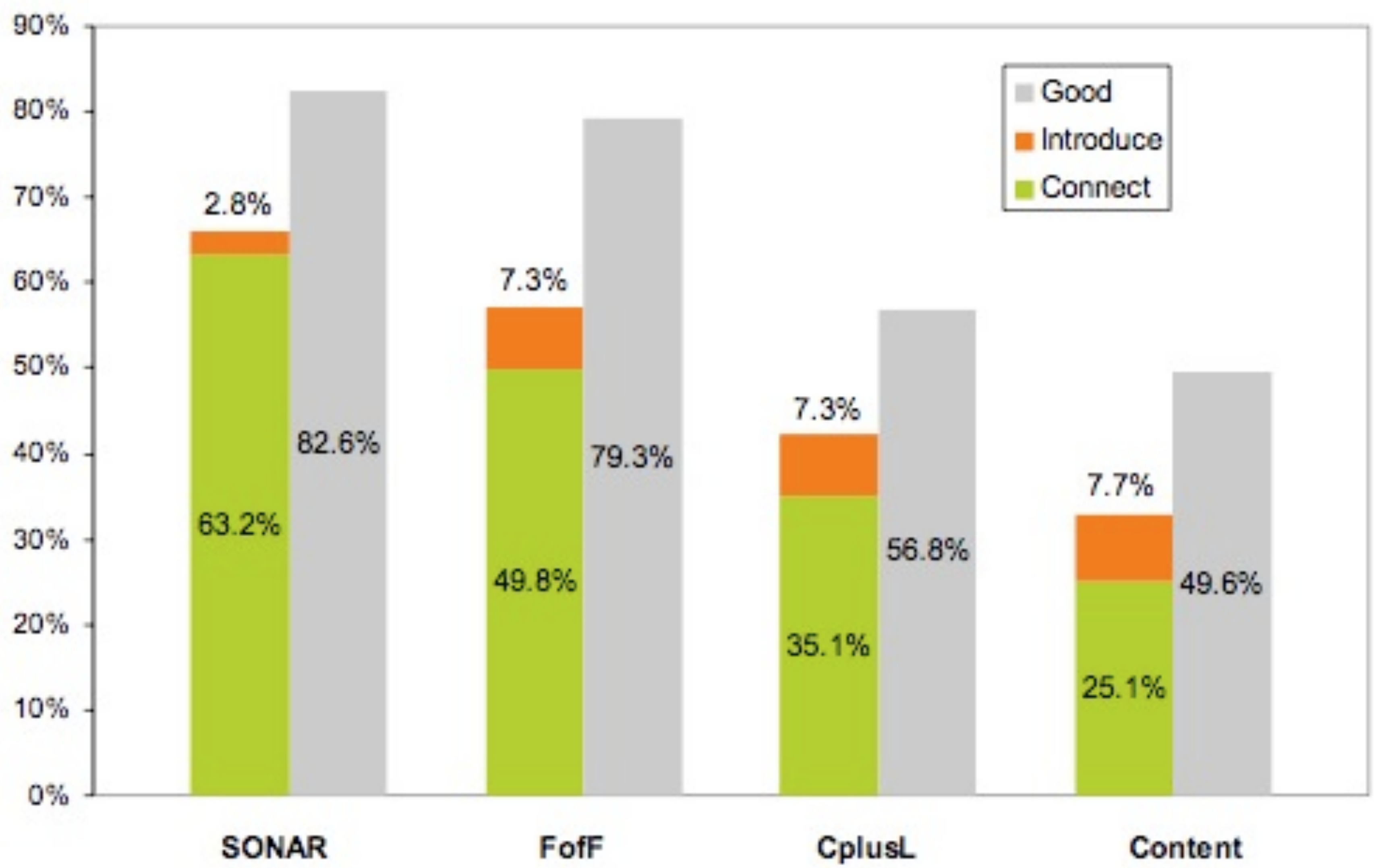



**Figure 2. Good recommendations that resulted in actions.**

# "Make New Friends, but Keep the Old" - Recommending People on Social Networking Sites

[Jilin Chen, et al., CHI2009]

| SONAR | FoF | CplusL | Content |
|---|---|---|---|
| 59.7% | 47.7% | 40.0% | 30.5% |

**Table 2. Recommendations resulting in connect actions.**

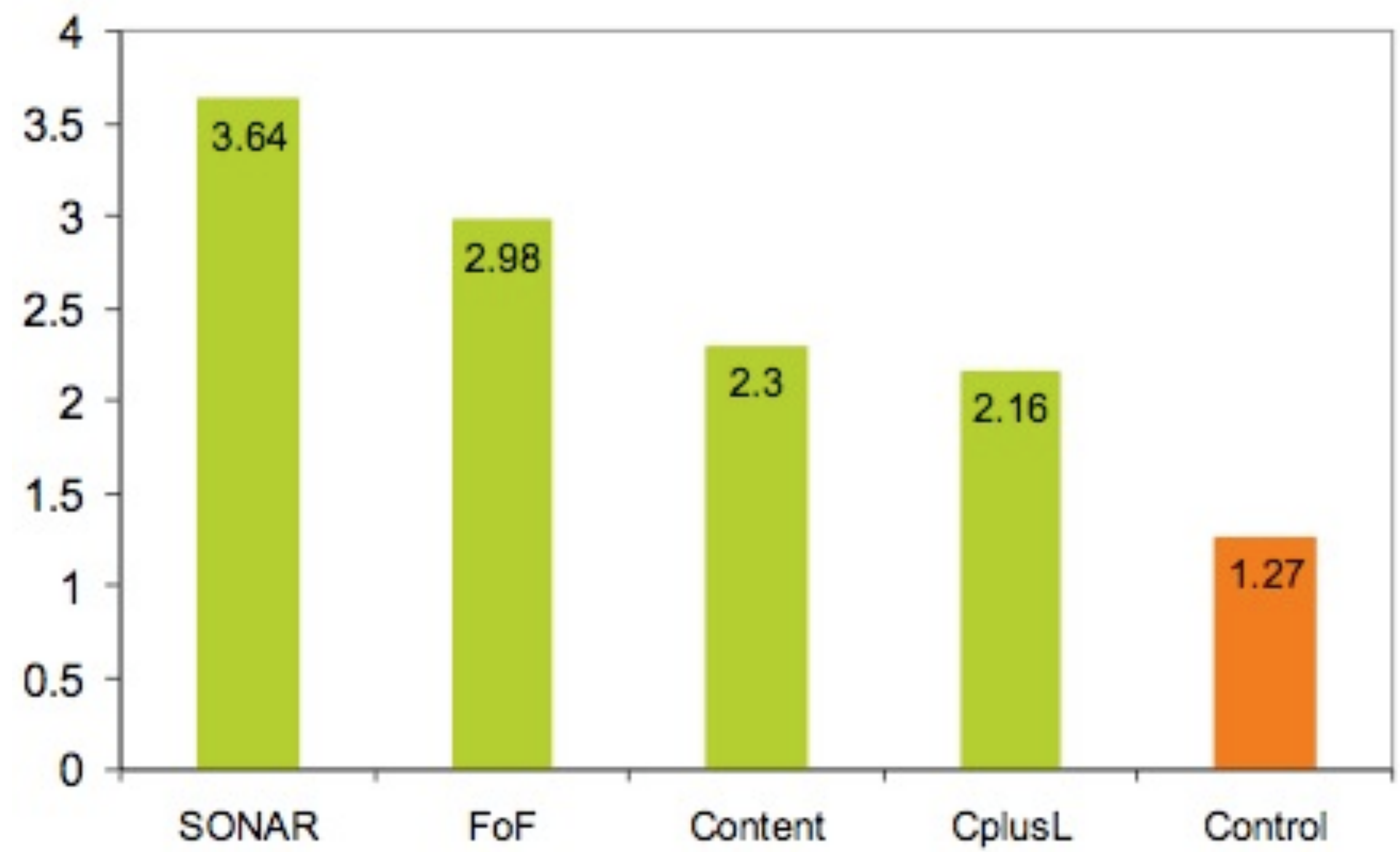



**Figure 4. Increase in number of friends.**

# "Make New Friends, but Keep the Old" - Recommending People on Social Networking Sites

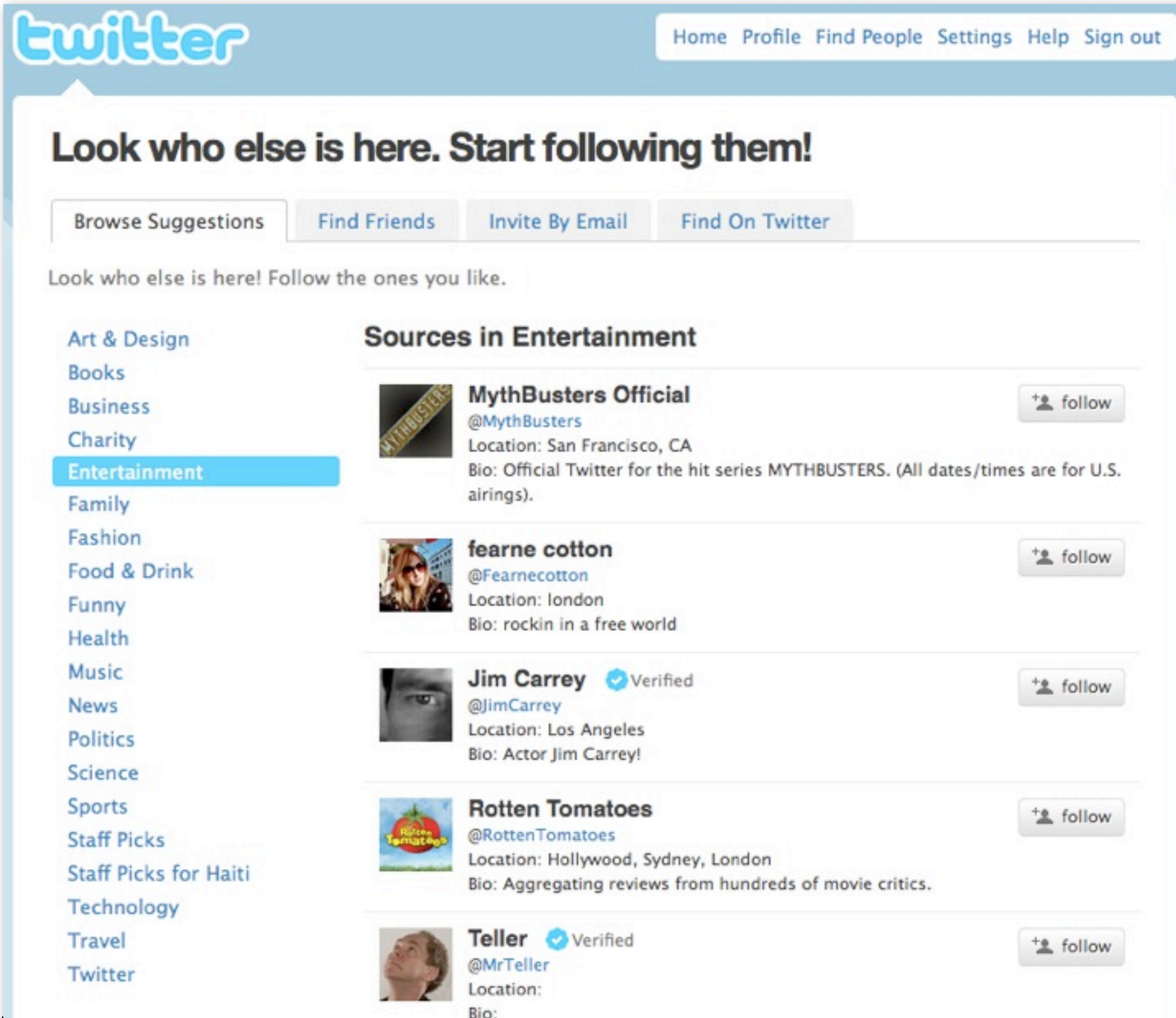
[Jilin Chen, et al., CHI2009]

- Conclusions
  - Relationship based algorithms (FoF and SONAR) outperform content similarity ones (Content and CplusL) in terms of user response
  - Relationship based algorithms are better at finding known contacts whereas content similarity algorithms were stronger at discovering new friends

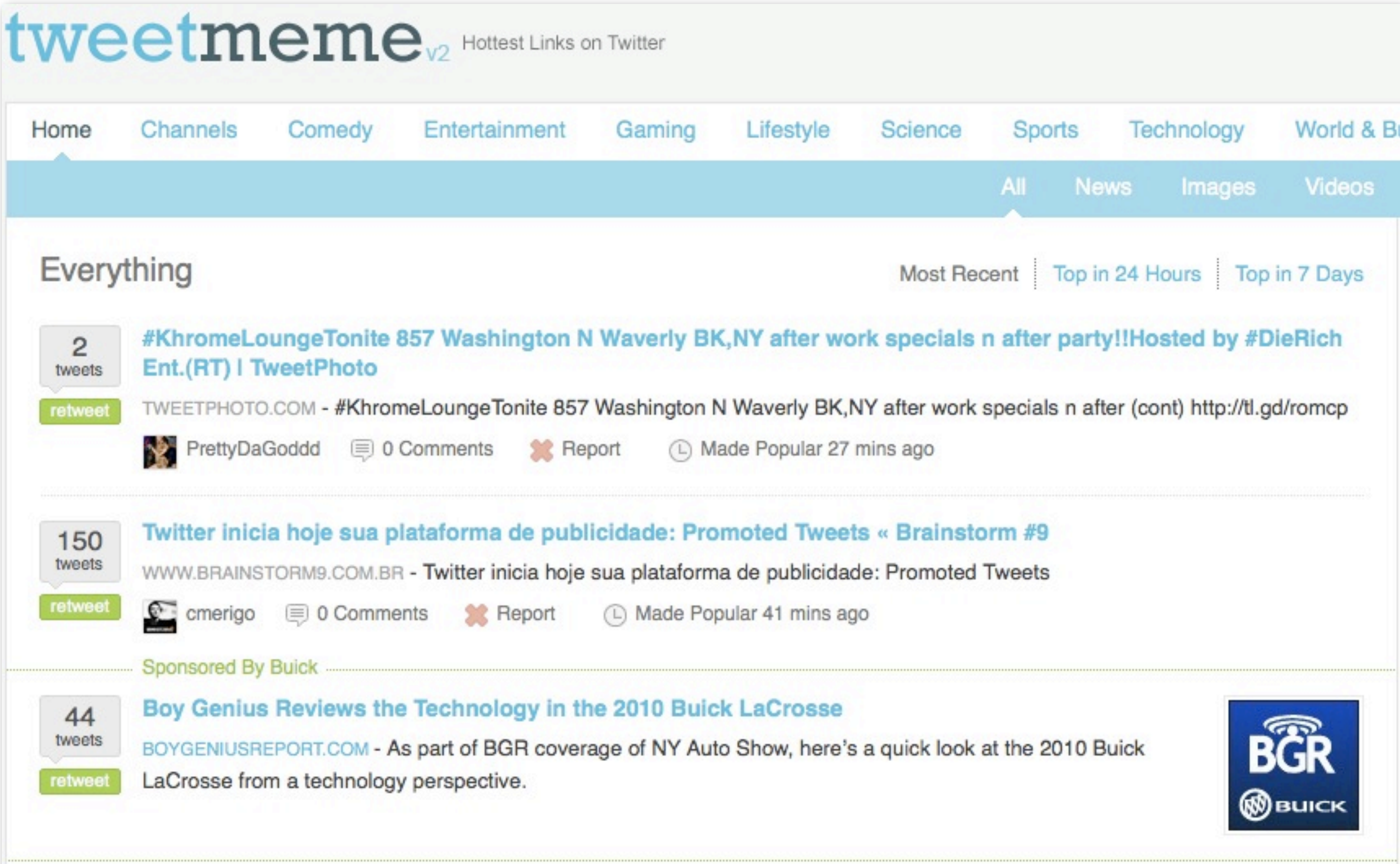
# Social Media Analysis

- Social Media Ranking
- Tag Recommendation
- News Recommendation
- User Recommendation
- Twitter-powered Recommendation

# Twitter Recommendation Engine



# Twitter-powered Recommendation



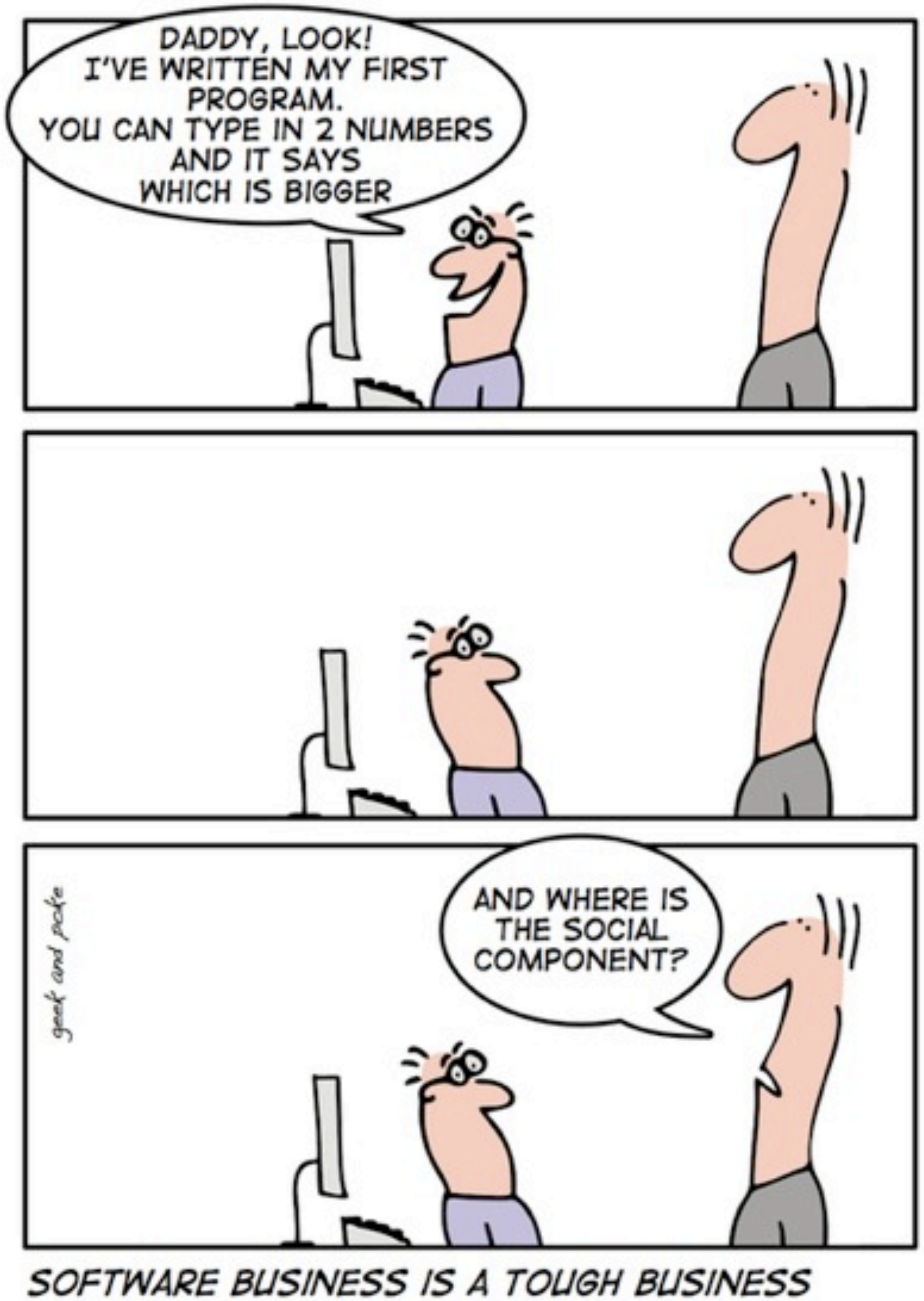
# Twitter-powered Recommendation

# Twitter-powered Recommendation

# References


- http://blog.oneriot.com/content/2009/06/oneriot-pulse-rank/
- http://uggedal.com/reddit.cf.algorithm.png
- http://www.seopedia.org/tips-tricks/social-media/the-digg-algorithm-unofficial-faq/
- http://www.technologyreview.com/web/24353/?a=f
- B. Sigurbjörnsson, and R. van Zwol. Flickr tag recommendation based on collective knowledge. In WWW, pages 327-336, 2008.
- J. Liu, P. Dolan, and E. R. Pedersen. Personalized news recommendation based on click behavior. In IUI, pages 31-40, 2010.
- J. Chen, W. Geyer, C. Dugan, M. J. Muller, and I. Guy. Make new friends, but keep the old: recommending people on social networking sites. In CHI, pages 201-210, 2009

# Are You Social Computing Ready?



# Q & A