# Collaborative Filtering

Peter Brusilovsky

with slides by Danielle Lee and Sue Syn

# Where we are?

| | Search | Navigation | Recommendation |
|---|---|---|---|
| Content-based | | | |
| Semantics / Metadata | | | |
| Social | | | |

# Agenda

- Context
- Concepts
- Uses
- CF vs. CB
- Algorithms
- Practical Issues
- Evaluation Metrics
- Future Issues

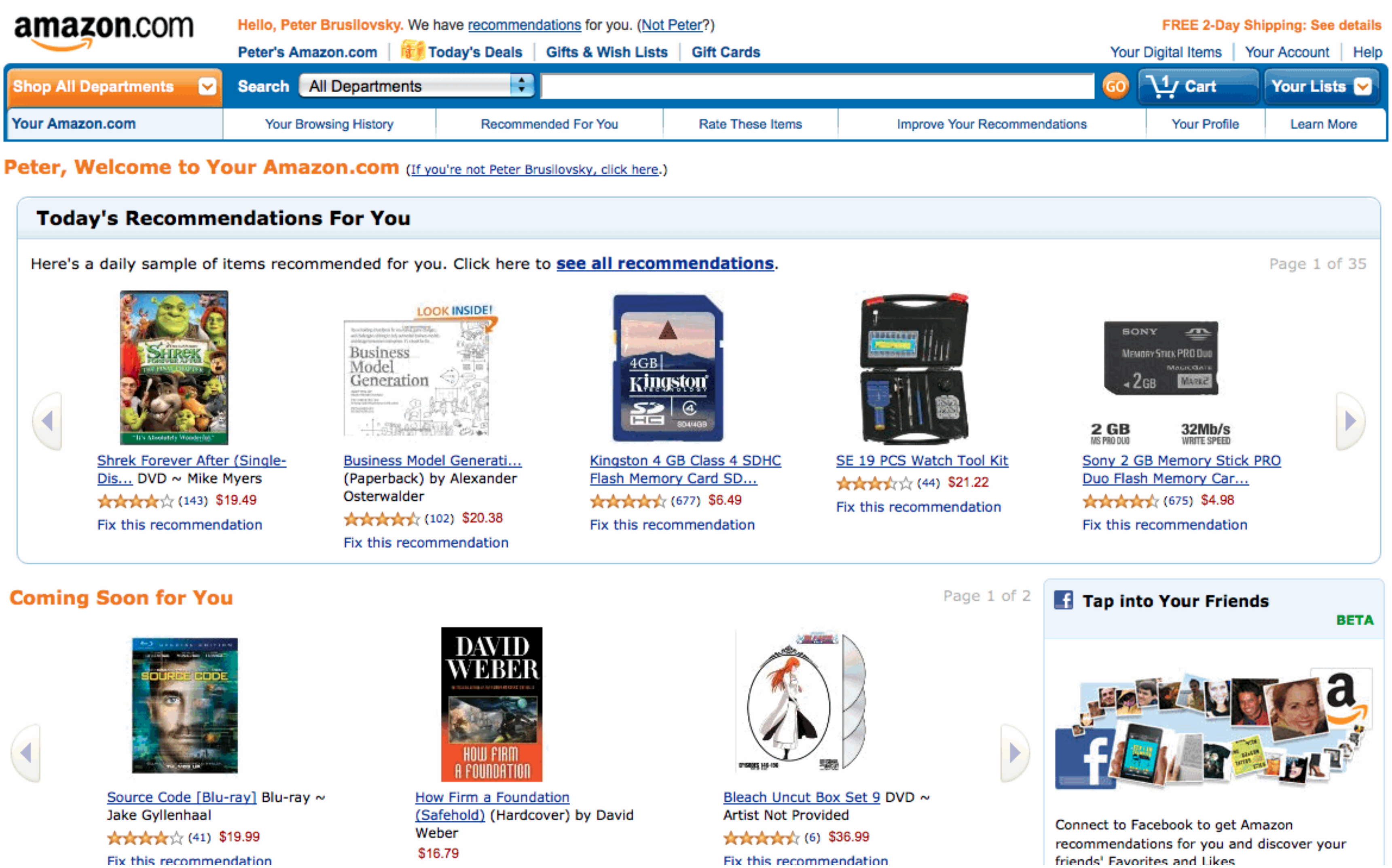
# Types of Recommender Systems

- Collaborative Filtering Recommender System
  - "Word-of-Mouth" phenomenon.
- Content-based Recommender System
  - Recommendation generated from the content features asso ciated with products and the ratings from a user.
- Case-based Recommender System
  - A kind of content-based recommendation. Information are r epresented as case and the system recommends the cases that are most similar to a user's preference.
- Hybrid Recommender System
  - Combination of two or more recommendation techniques to gain better performance with fewer of the drawbacks of an y individual one (Burke, 2002).

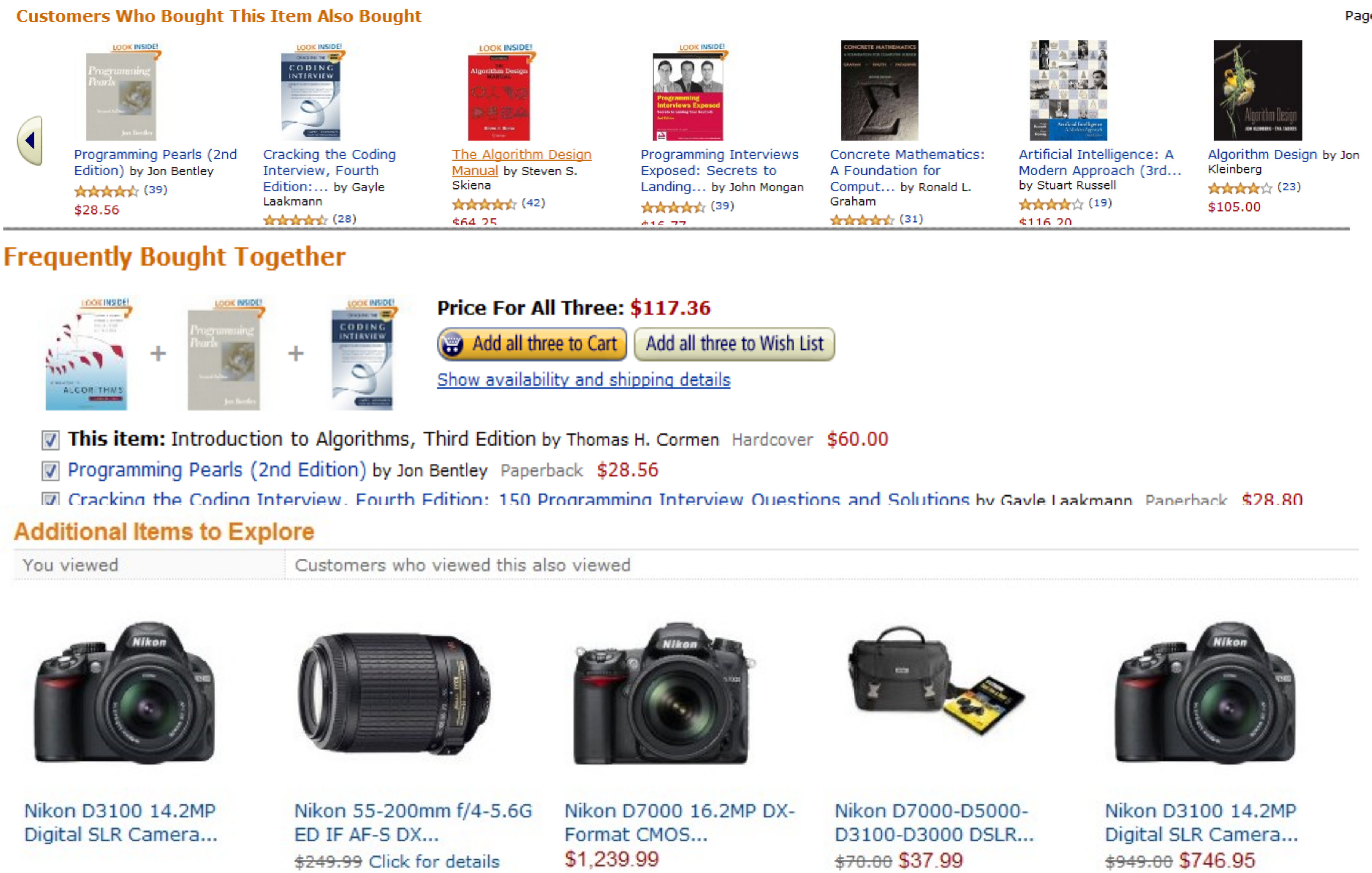
# Recommendation Procedure

1. Understand and model users
2. Collect candidate items to recommend.
3. Based on your recommendation method, predict target users' preferences for each candidate item.
4. Sort the candidate items according to the prediction probability and recommend them.

# Example: Amazon.com



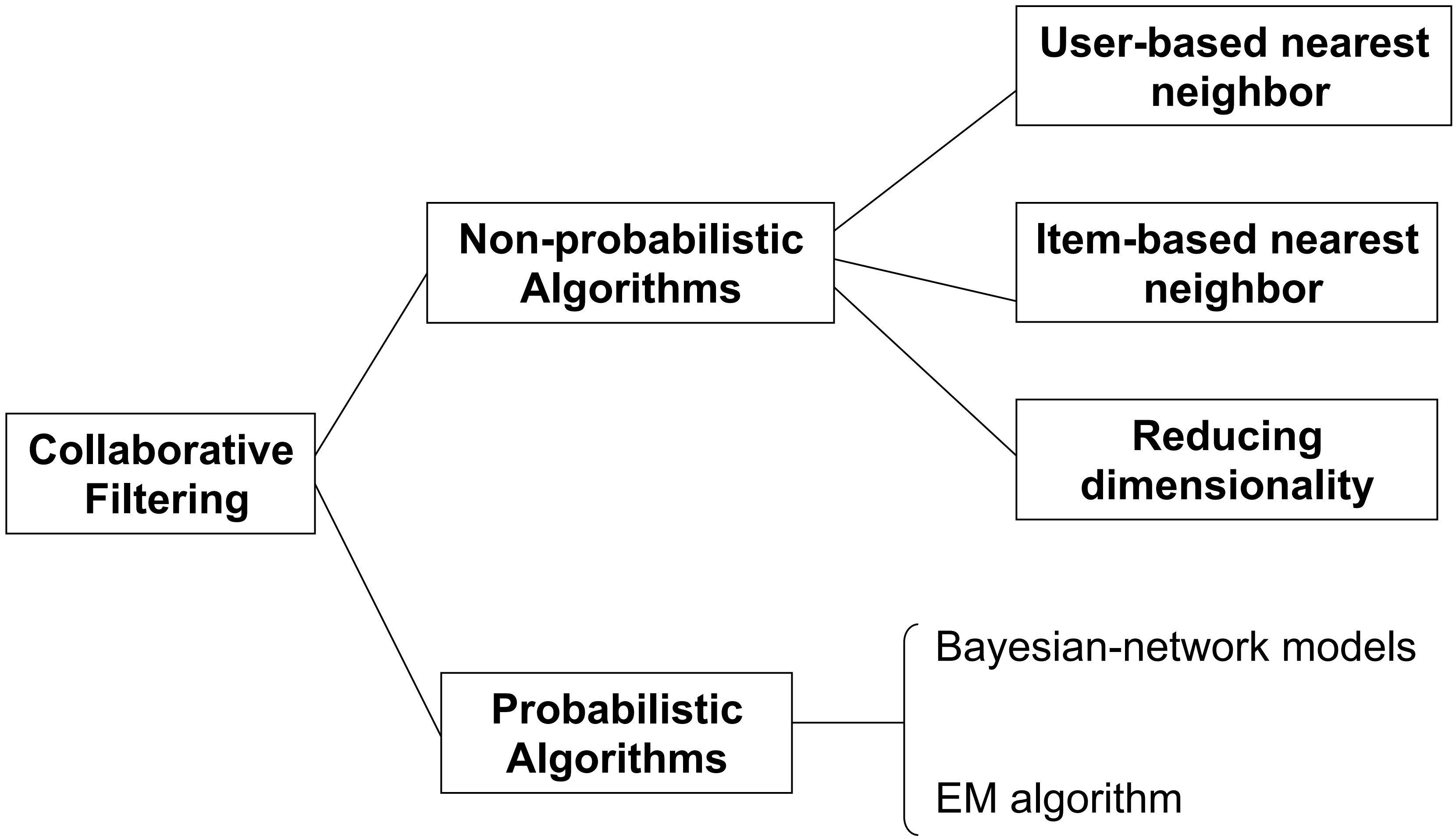
# Amazon’s Source of Wisdom



# What is Collaborative Filtering?

- Traced back to the *Information Tapestry* project at Xerox PARC
  - It allowed its users to annotate the documents that they read and system recommends
- Expanded to “automatic” CF in the works of Resnick, Riedl, Maes
- More general definition as ‘the process of filtering or evaluating items using the opinions of other people.’
- CF recommends items which are likely interesting to a target user based on the evaluation averaging the opinions of people with similar tastes
- Key idea: people who agreed with me in the past, will also agree in the future.
  - On the other hand, the assumption of Content-based recommendation is that Items with similar objective features will be rated similarly.

# Algorithms



Collaborative Filtering
Non-probabilistic Algorithms
User-based nearest neighbor
Item-based nearest neighbor
Reducing dimensionality
Probabilistic Algorithms
Bayesian-network models
EM algorithm

# Concepts

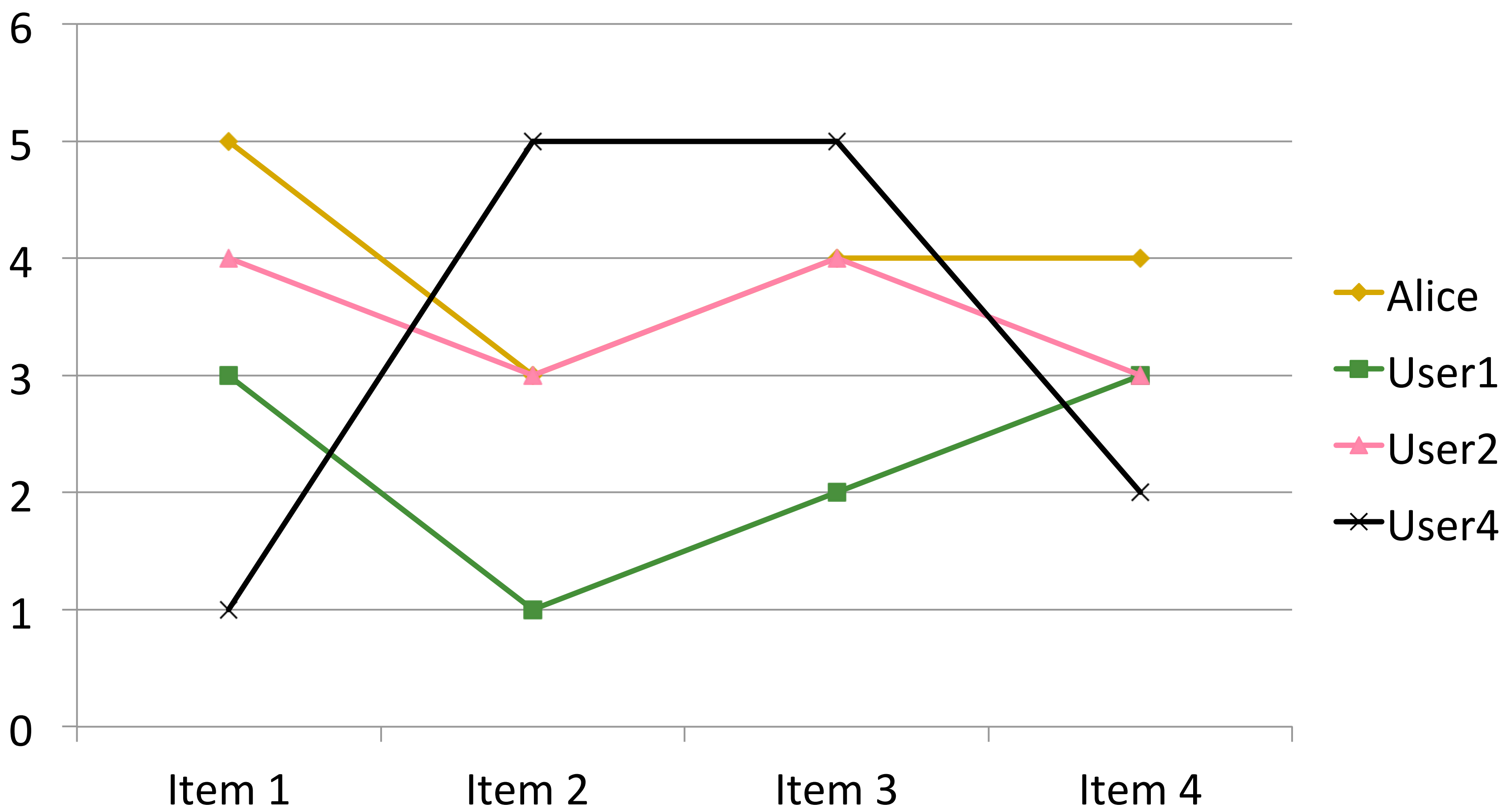
- Collaborative Filtering
  - The goal of collaborative filtering is to predict how well a user will like an item that he has not rated given a se t of historical preference judgments for a community of users.
- User
  - Any individual who provides ratings to a system
- Items
  - Anything for which a human can provide a rating

# User-based CF

The input for the CF prediction algorithms is a matrix of users' ratings on items, referred as the **ratings matrix**.

| Target User | Item 1 | Item 2 | Item 3 | Item 4 | Item 5 | Average |
|---|---|---|---|---|---|---|
| Alice | 5 | 3 | 4 | 4 | ??? | 16/4 |
| User1 | 3 | 1 | 2 | 3 | 3 | 9/4 |
| User2 | 4 | 3 | 4 | 3 | 5 | 14/4 |
| User3 | 3 | 3 | 1 | 5 | 4 | 12/4 |
| User4 | 1 | 5 | 5 | 2 | 1 | 13/4 |

# User-based CF (2)



# User-Based NN Recommendation

1.Select like-minded peer group for a target user

2. Choose candidate items which are not in the list of the target user but in the list of peer group.

3.Score the items by producing a weighted score and predict the ratings for the given items.

4.Select the best candidate items and recommend them to a target user.

Redo all the procedures through 1 ~ 4 on a timely basis
.

# User-based NN: User Similarity

- Pearson's Correlation Coefficient for User a and User b for all rated Products, P.

$$sim(a,b) = \frac{\sum_{p \in product(P)} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in product(P)} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in product(P)} (r_{b,p} - \bar{r}_b)^2}}$$

Average rating of user ***b***

- Pearson correlation takes values from +1 (Perfectly positive correlation) to -1 (Perfectly negative correlation) .

# User-based NN: Rating Prediction

$$pred(a,p) = \bar{r}_a + \frac{\sum_{b \in neighbors(n)} sim(a,b) \cdot (r_{b,p} - \bar{r}_b)}{\sum_{b \in neighbors(n)} sim(a,b)}$$

# One Typical CF recommendation

**m o v i e l e n s**
helping you find the *right* movies

**Welcome suleehs@yahoo.co.kr** (Log Out)
You've rated **10** movies.
*You're the 26th visitor in the past hour.*

★★★★★ = Must See
★★★★☆ = Will Enjoy
★★★☆☆ = It's OK
★★☆☆☆ = Fairly Bad
★☆☆☆☆ = Awful

So far you have rated **10** movies.
MovieLens needs at least **15** ratings from you to generate predictions for you.
Please rate as many movies as you can from the list below.

**next >**

| | Your Rating | Movie Information |
|---|---|---|
| | 4.5 stars | **Butterfly Effect, The (2004)**<br>Drama, Sci-Fi, Thriller |
| ??? | Not seen | **Chain Reaction (1996)**<br>Action, Adventure, Thriller |
| ??? | Not seen | **High Noon (1952)**<br>Western |
| | 3.5 stars | **Joy Luck Club, The (1993)**<br>Drama |
| ??? | Not seen | **Matilda (1996)**<br>Children, Comedy, Fantasy |
| | 4.0 stars | **Out of Africa (1985)**<br>Drama, Romance |
| ??? | Not seen | **Scrooged (1988)**<br>Comedy, Fantasy, Romance |
| | 1.5 stars | **Six Days Seven Nights (1998)**<br>Adventure, Comedy, Romance |
| ??? | Not seen | **Sword in the Stone, The (1963)**<br>Animation, Children, Fantasy, Musical |
| | 4.0 stars | **Working Girl (1988)**<br>Comedy, Drama, Romance |

**next >**

# One Typical CF recommendation

# Benefits of Collaborative Filtering

- Collaborative filtering systems work by people in system, and it is expected that people to be better at evaluating information than a computed function
- CF doesn’t require content analysis & extraction
- Independent of any machine-readable represent ation of the objects being recommended.
  - Works well for complex objects (or multimedia) such as music, pictures and movies
- More diverse and serendipitous recommendation

# CF vs. CB

| | CF | CB |
|---|---|---|
| Compare | Users interest | Item info |
| Similarity | Set of users<br>User profile | Item info<br>Text document |
| Shortcoming | Needs other users' feedback -> cold start<br>Coverage<br>Unusual interest | Feature matters<br>Over-specialize<br>Eliciting user feedback |

# Uses for CF : Domains

- Many items
- Many ratings
- Many more users than items recommended
- Users rate multiple items
- For each user of the community, there are other users with common needs or tastes
- Item evaluation requires personal taste
- Items persists
- Taste persists
- Items are homogenous

# More on User-Based NN

- Other difference measures: Adjusted Cosine Similarity, Spearman's rank correlation coefficient
- Reduce the relative importance of the agreement on universally liked items : *inverse user frequency* (Breese, et al., 1998) and *variance weighting factor* (Herlocker, et al., 1999).
- Skewed neighboring is possible: *Significance weighting* (Herlocker, et al., 1999).
- Calculating a user's perfect neighborhood is imme-nsely resource intensive calculations

# Item-based NN Recommendation

| Target User | Item 1 | Item 2 | Item 3 | Item 4 | Item 5 | Average |
|---|---|---|---|---|---|---|
| Alice | 5 | 3 | 4 | 4 | ??? | 4.0 |
| User1 | 3 | 1 | 2 | 3 | 3 | 2.4 |
| User2 | 4 | 3 | 4 | 3 | 5 | 3.8 |
| User3 | 3 | 3 | 1 | 5 | 4 | 3.2 |
| User4 | 1 | 5 | 5 | 2 | 1 | 2.8 |

# Item-based Nearest Neighbor

| | Item 1 | Item 2 | Item 3 | Item 4 | Item 5 |
|---|---|---|---|---|---|
| Alice | 1 | −1 | 0 | 0 | |
| User1 | 0.6 | −1.4 | −0.4 | 0.6 | 0.6 |
| User2 | 0.2 | −0.8 | 0.2 | −0.8 | 1.2 |
| User3 | −0.2 | −0.2 | −2.2 | 1.8 | 0.8 |
| User4 | −1.8 | 2.2 | 2.2 | −0.8 | −1.8 |

# Item-Based NN Recommendation

- Generate predictions based on similarities between items
  - Usually a cosine similarity used
- Prediction for a user *u* and item *i* is composed of a weighted sum of the user *u*'s ratings for items most similar to *i*.

$$pred(u,i) = \frac{\sum_{j \in ratedItems(u)} sim(i,j) \cdot r_{ui}}{\sum_{j \in ratedItems(u)} sim(i,j)}$$

# Item-based Nearest Neighbor

- More computationally efficient than user-based near est neighbors.
- Compared with user-based approach that is affected by the small change of users' ratings, item-based approach is more stable.
- Recommendation algorithm used by Amazon.com (Linden et al., 2003).

# Uses for CF : User Tasks

- What tasks users may wish to accomplish
  - Help me find new items I might like
  - Advise me on a particular item
  - Help me find a user (or some users) I might like
  - Help our group find something new that we might like
  - Domain-specific tasks
  - Help me find an item, new or not

# Uses for CF : System Tasks

- What CF systems support
  - Recommend items
    - Eg. Amazon.com
  - Predict rating for a given item
  - Constrained recommendations
    - Recommend best items from a set of items

# Other Non-Probabilistic CF Algorithms

- Association Rule Mining
  - I.e., "If a customer purchases baby food then the customer also buys diapers in 70% of the cases."
  - Build Models based on commonly occurring patterns in the ratings matrix.
  - "If user X liked both item 1 and item 2, then X will most pro bably also like item 5."

$$\text{Support}(X \rightarrow Y) = \frac{\text{Number of Transactions containing } X \cup Y}{\text{Number of Transactions}}$$

$$\text{Confident}(X \rightarrow Y) = \frac{\text{Number of Transactions containing } X \cup Y}{\text{Number of Transactions containing } X}$$

# Simple Probabilistic Algorithms

- Represent probability distributions
- Given a user *u* and a rated item *i*, the user assigned the item a rating of *r* : $p(r|u, i)$.

$$E(r \mid u,i) = \sum_r r \cdot p(r \mid u,i)$$

- Bayesian-network models, Expectation maximization (EM) algorithm

# Dimensionality Reduction Algorithms

- Map item space to a smaller number of underlying “dimensions”
- Matrix Factorization/Latent Factor models:
  - Singular Value Decomposition,
  - Principal Component Analysis,
  - Latent Semantic Analysis, etc.
- Expensive offline computation and mathematical complexity
- Will be presented in a separate lecture

# Dimensionality Reduction Algorithms

- Matrix Factorization got an attention since Netflix Prize competition.

# Practical Issues : Ratings

- Explicit vs. Implicit ratings
  - Explicit ratings
    - Users rate themselves for an item
    - Most accurate descriptions of a user’s preference
    - Challenging in collecting data
  - Implicit ratings
    - Observations of user behavior
    - Can be collected with little or no cost to user
    - Ratings inference may be imprecise.

# Practical Issues : Ratings

- Rating Scales
  - Scalar ratings
    - Numerical scales
    - 1-5, 1-7, etc.
  - Binary ratings
    - Agree/Disagree, Good/Bad, etc.
  - Unary ratings
    - Good, Purchase, etc.
    - Absence of rating indicates no information

# Practical Issues : Cold Start

- New user
  - Rate some initial items
  - Non-personalized recommendations
  - Describe tastes
  - Demographic info.
- New Item
  - Non-CF : content analysis, metadata
  - Randomly selecting items
- New Community
  - Provide rating incentives to subset of community
  - Initially generate non-CF recommendation
  - Start with other set of ratings from another source outside community

# Evaluation Metrics

- Accuracy
  - Predict accuracy
    - The ability of a CF system to predict a user's rating for an item
    - Mean absolute error (MAE)
    - Classic, but now often criticized
  - Rank accuracy
    - Precision – percentage of items in a recommendation list that the user would rate as useful
    - Half-life utility – percentage of the maximum utility achieved by the ranked list in question

# Evaluation Metrics

- Novelty
  - The ability of a CF system to recommend items that the user was not already aware of.
- Serendipity
  - Users are given recommendations for items that they would not have seen given their existing channels of discovery.
- Coverage
  - The percentage of the items known to the CF system for which the CF system can generate predictions.

# Evaluation Metrics

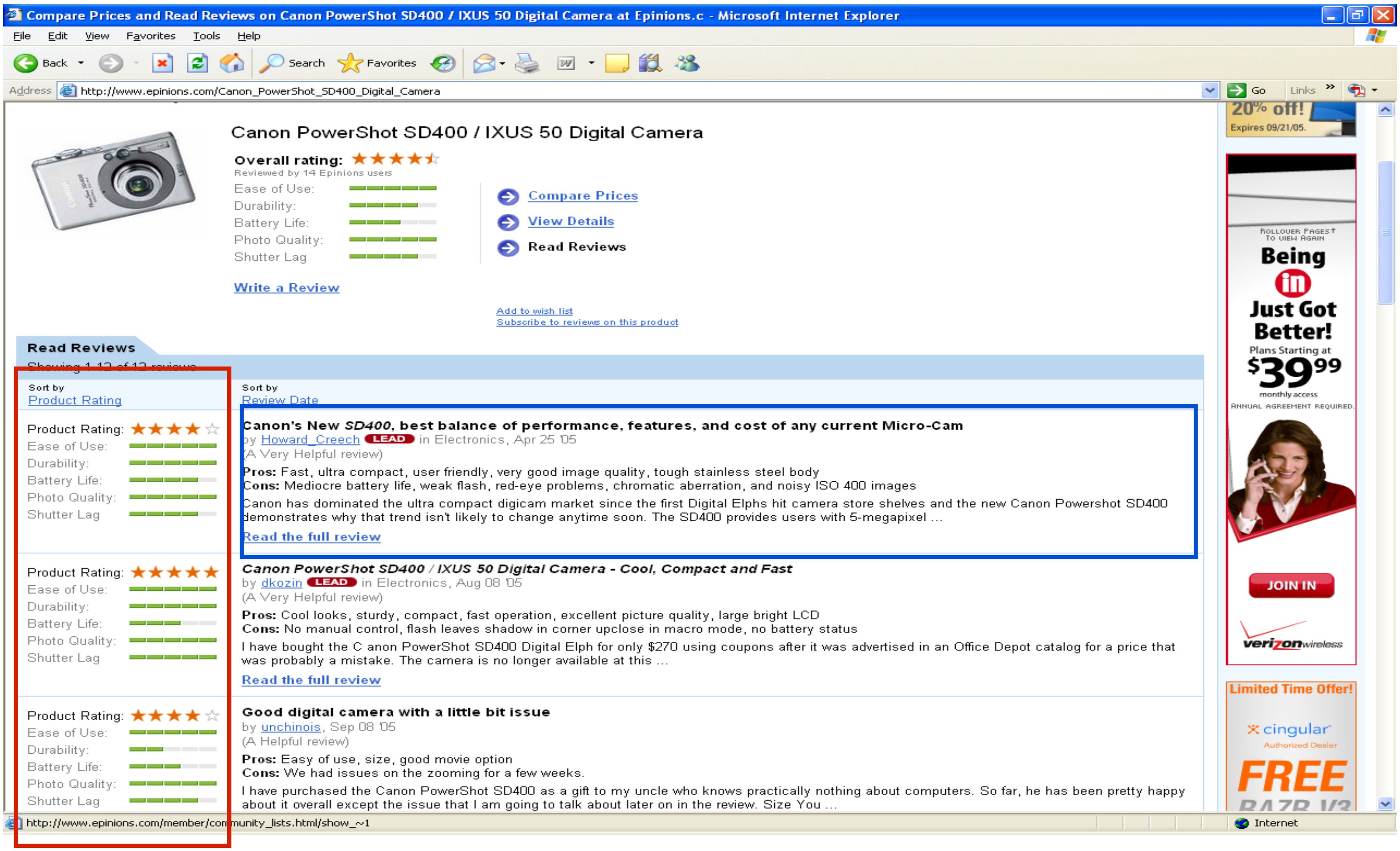
- Learning Rate
  - How quickly the CF system becomes an effective predictor of taste as data begins to arrive.
- Confidence
  - Ability to evaluate the likely quality of its prediction s.
- User Satisfaction
  - By surveying the users or measuring retention and use statistics

# Additional Issues : Interfaces

- Social Navigation
  - Make the behavior of community visible
  - Leaving “footprints” : read-wear / edit-wear
  - Attempt to mimic more accurately the social proce ss of word-of-mouth recommendations
  - Epinions.com

# Additional Issues : Interfaces

Epinions.com (http://www.epinions.com)



# Additional Issues : Interfaces

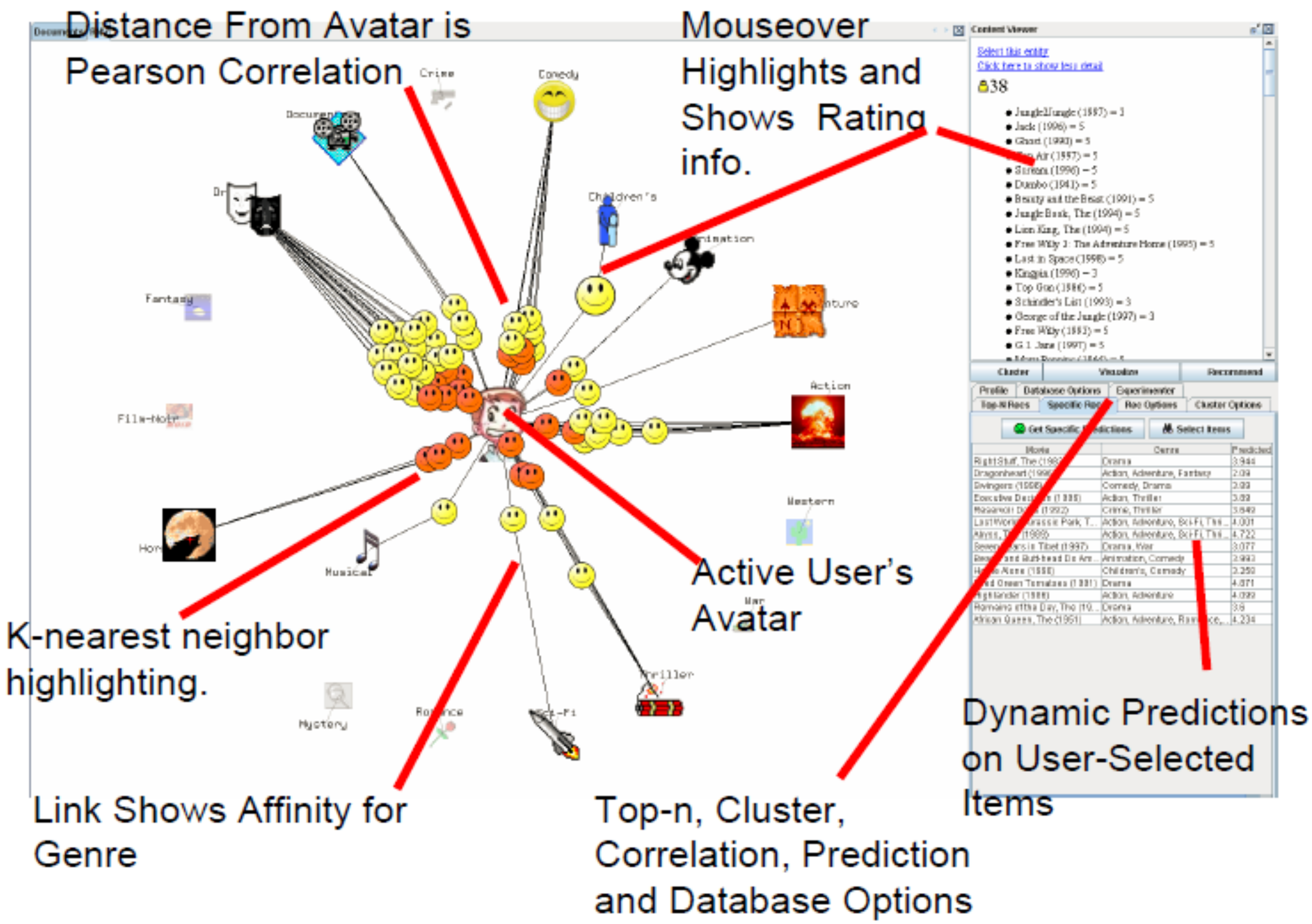
- Explanation
  - Where, how, from whom the recommendations are generated.
  - Do not make it too much!
    - Not showing reasoning process
    - Graphs, key items
    - Reviews

# Additional Issues : Privacy & Trust

- User profiles
  - Personalized information
- Distributed architecture

- Recommender system may break trust when malicious users give ratings that are not repre sentative of their true preferences.

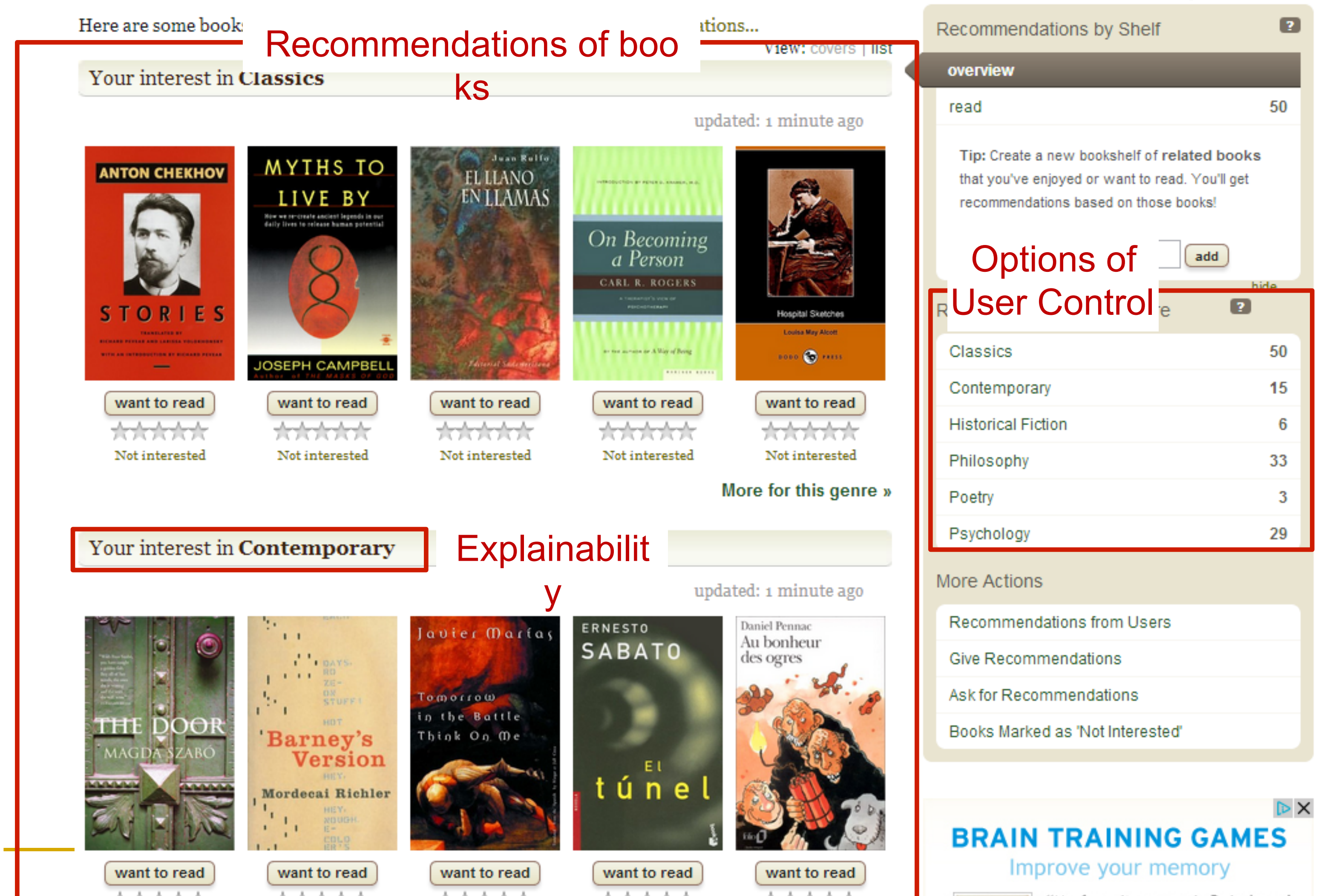
# Choose your Peers




- PeerChooser (CHI 2008) John O'Donovan and Barry Smyth (UCD)

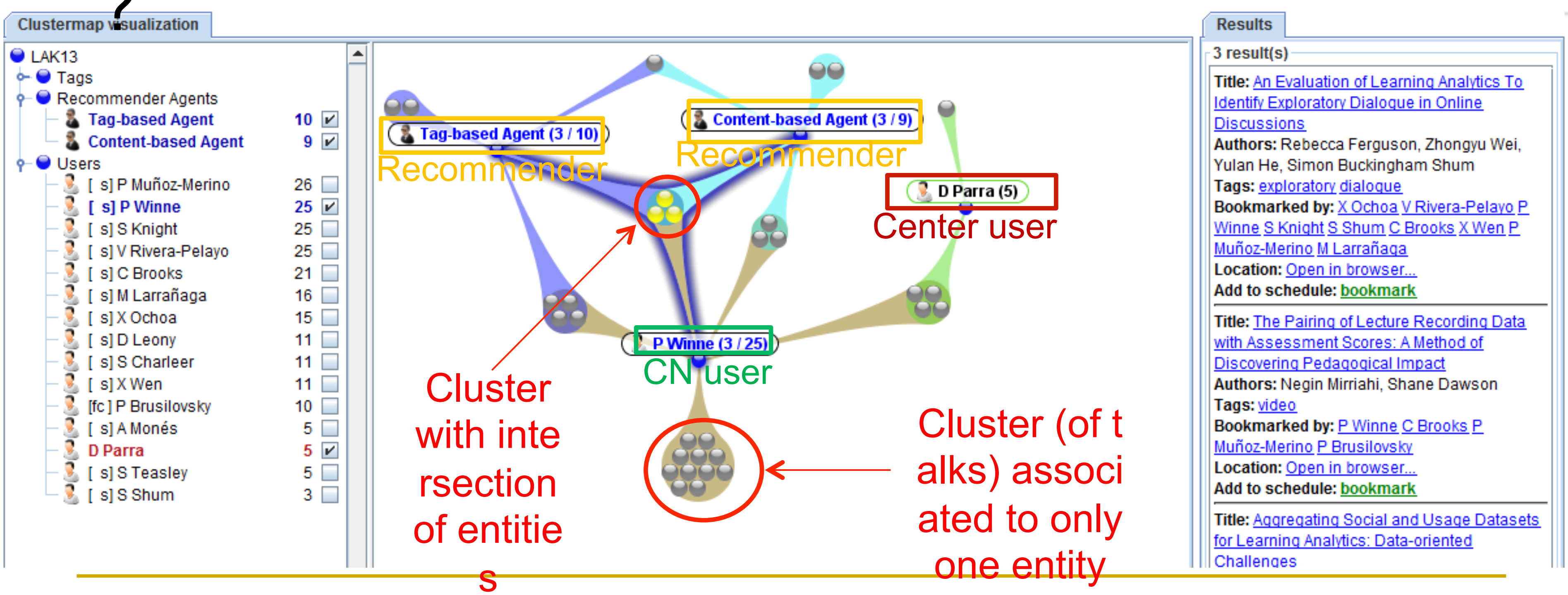
# The Concept of *Controllability*

## GoodReads: Book recommender system

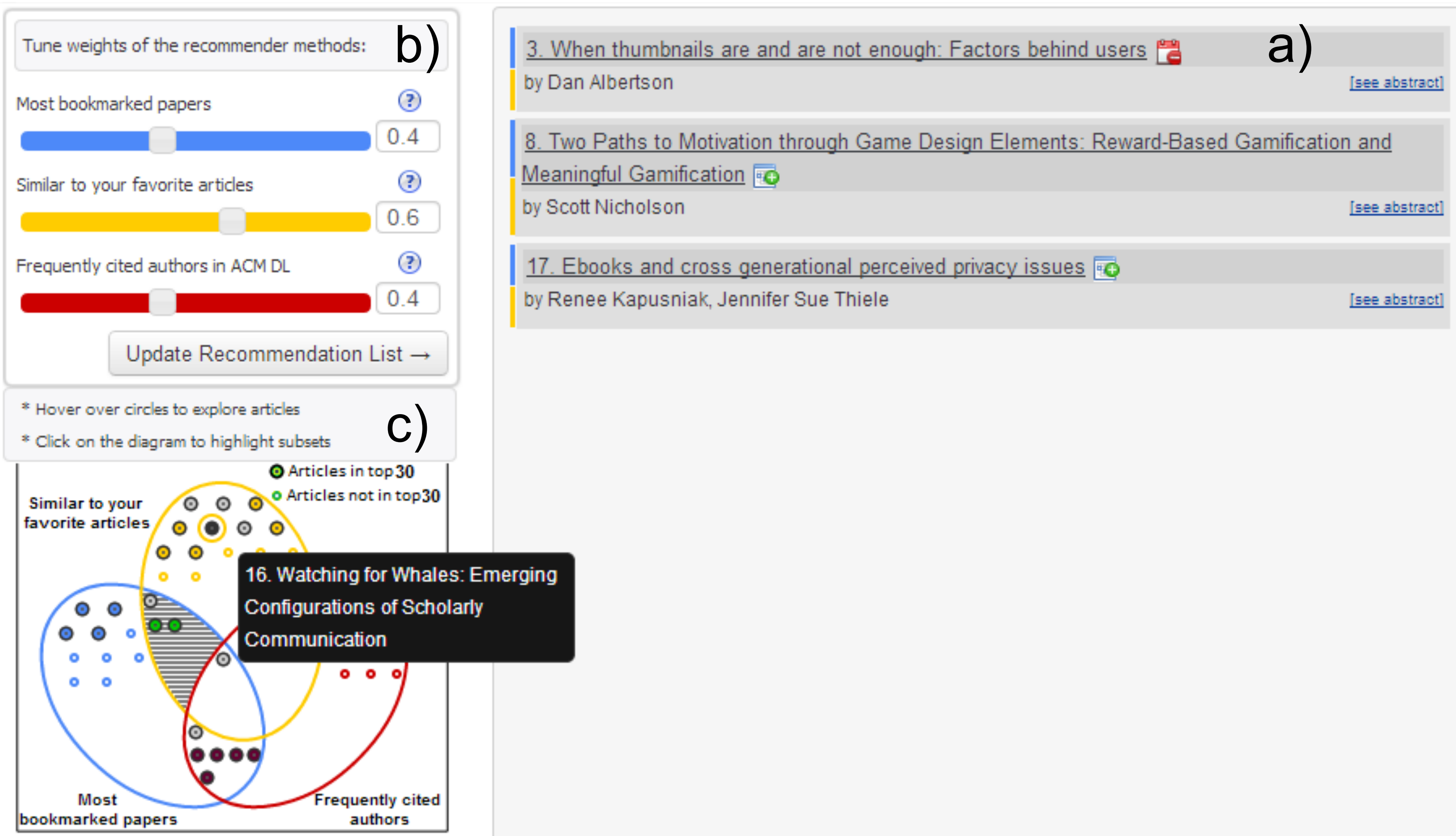


# TalkExplorer

- Adaptation of Aduna Visualization in CN
- Main research question: Do fusion (intersection) of contexts of relevance improve user experience ?



# SetFusion



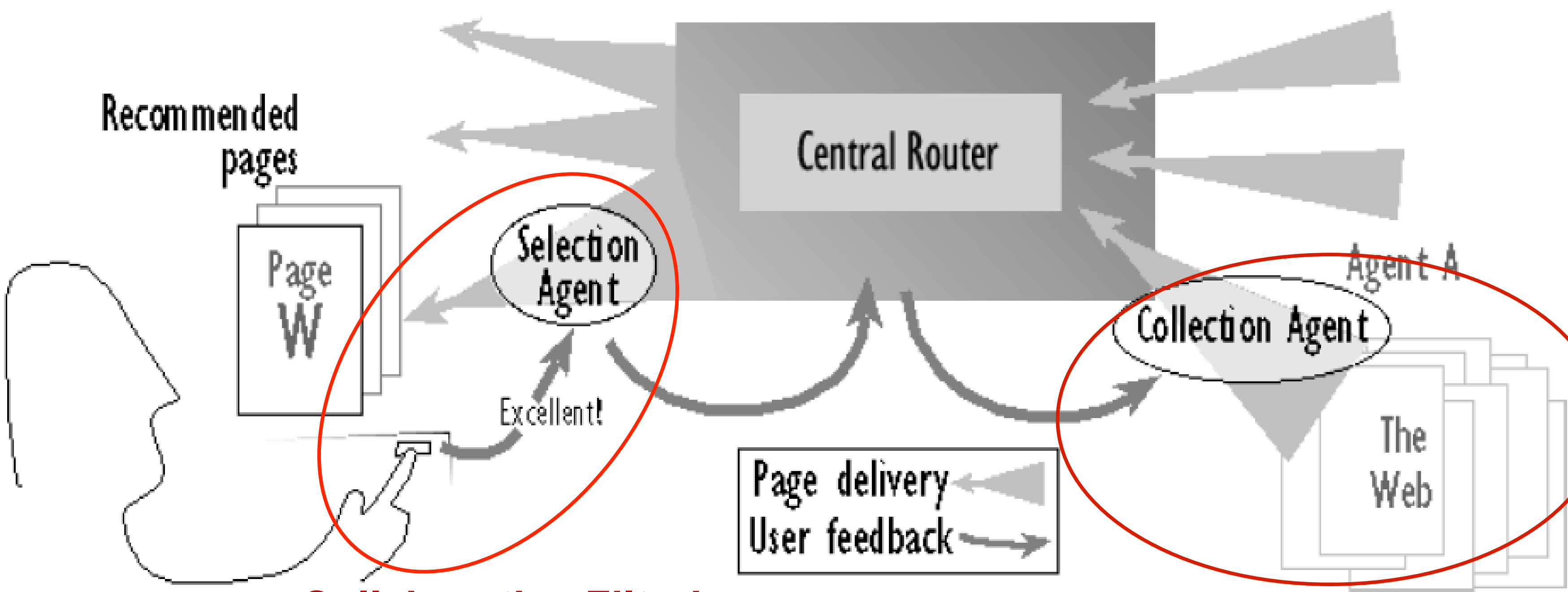

a) Recommended Talks, b) Controllable Sliders, c) Venn Diagram to Control (click) & Inspect (hover)

# Additional Issues : Hybrid Approach

- CF + CB
- Content based system
  - Maintain user profile based on content analysis
- Collaborative system
  - Directly compare profiles to determine similar user s for recommendation
- Fab system

# Additional Issues : Hybrid Approach

**Example : Fab System Architecture**




**Collaborative Filteri ng**

**Content-Based Filtering**