Recommender Systems
Case study: Classical Examples

Joonseok Lee

Georgia Institute of Technology

2011/01/25

Introduction

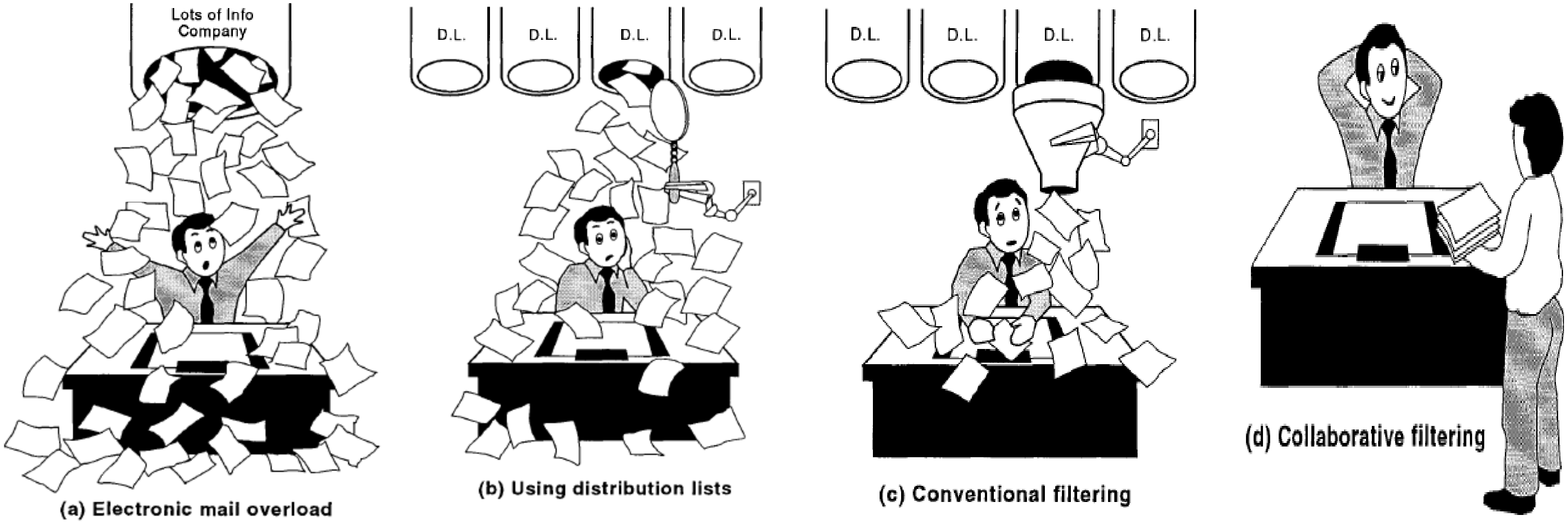
- 4 classic examples of Recommender Systems
  - Tapestry (ACM Communication 1992)
  - GroupLens (ACM CSCW 1994)
  - Virtual Community (CHI 1995)
  - Ringo (CHI 1995)
- Pre-Internet Era
  - Only few people used dedicated news/mail system.
- Terminologies may differ from today's Recommender system or Machine learning terminologies.

Tapestry

- Domain: texts like e-mail, news
- Goal: controlling flood of text information by filtering out unimportant ones.

- Collaborative Filtering definition
  - People collaborate to help one another perform filtering, by recording their reactions to documents they read.

Tapestry: Goal

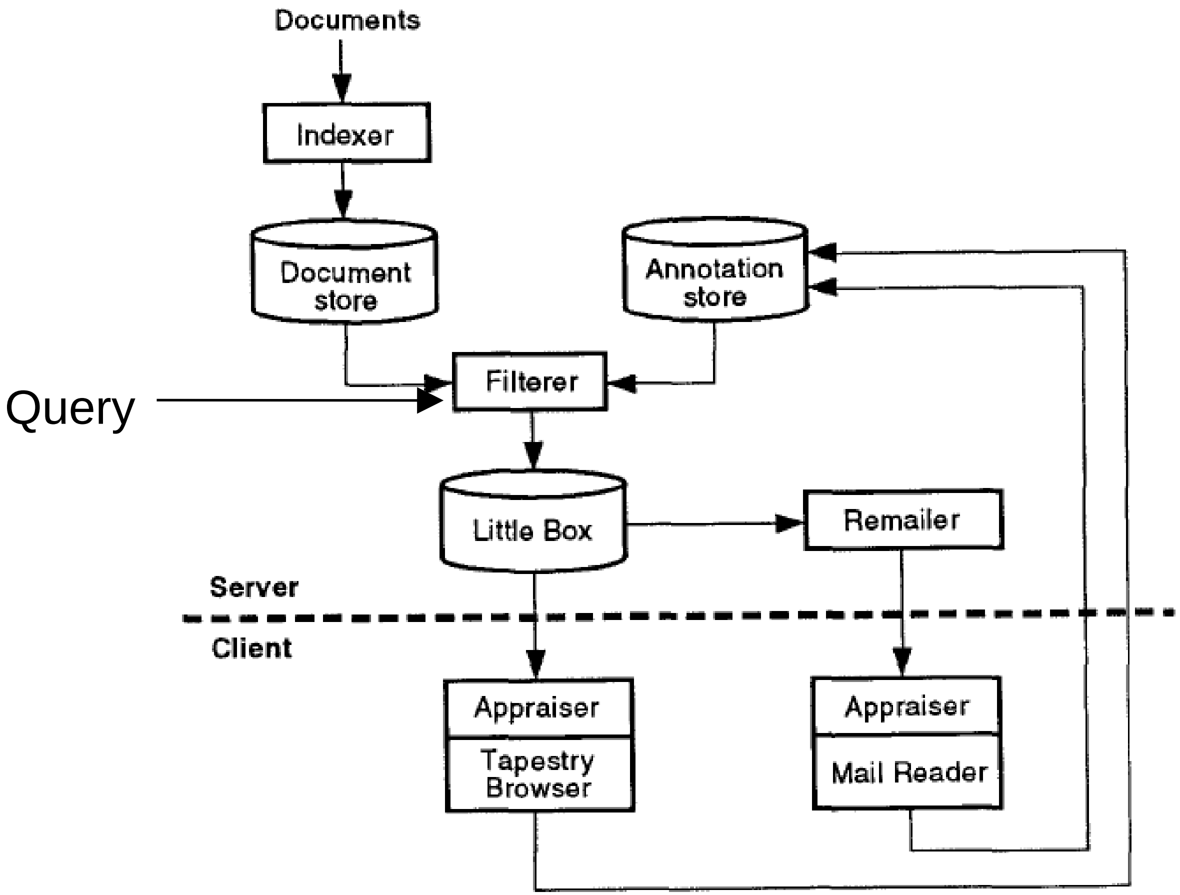


(a) Electronic mail overload (b) Using distribution lists (c) Conventional filtering (d) Collaborative filtering

Tapestry: Idea

- Annotation
- DB-based, with index
- TQL (Tapestry Query Language)
- Two clients
  - Mail Reader: typical mail reader
  - Tapestry browser: annotation, filter define, TQL query

Tapestry: Architecture



Tapestry: Discussion

- First idea of filtering useful information, based on other users' feedback.
- Did not discuss detailed algorithm for how to find similar users, how to do personalized recommendation.

GroupLens

- Domain: NetNews
- Goal: enable users to predict the quality of news articles before reading it.

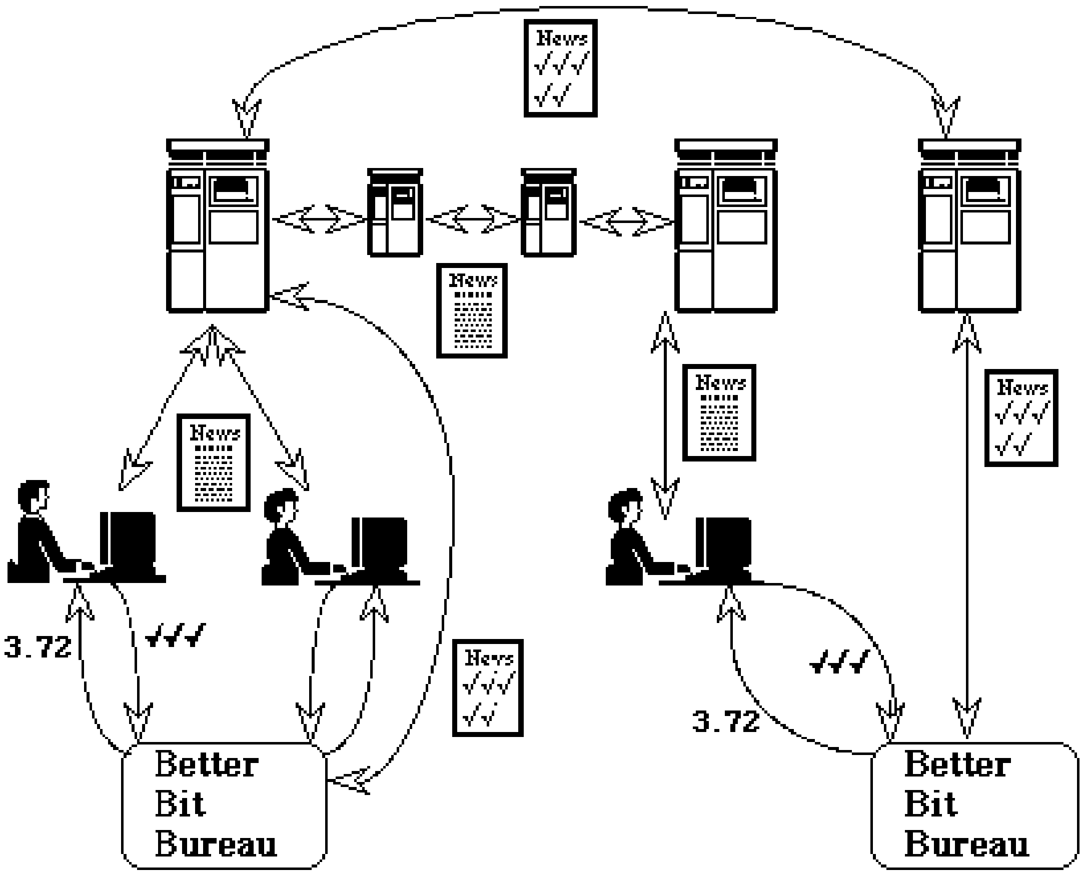
- Problem
  - There are large number of garbage articles, so it is becoming difficult to filter useful ones.
  - Previously, this problem was handled by manual filtering or splitting threads of articles.

GroupLens: Approach

- User-based Collaborative Filtering
  - Find users who have similar taste with me.
  - Then, show their ratings on that article.

- Method
  - Similar user calculation: Pearson-r correlation
  - Score prediction: weighted sum of ratings from similar user
  - Score is:
    - simply displayed
    - filtered out below a threshold
    - sorted
    - graphically represented
    - A-F scale, familiar for students

GroupLens: Architecture



GroupLens: Discussion

- Social implication
  - Recommender system will reduce garbage documents.
  - Incentive problem: who's effort? and who's benefit?

- Discussed how to use calculated score, including graphical representation.

Virtual Community

- Domain: movie
- Goal: personalized movie recommendation, based on subject ratings of others.

- HCI-perspective
  - More and more multimedia data we have, make difficult to search or recommend them, or develop user interface for such systems.

Virtual Community: Concept

- Virtual Community
  - Community: a group of people who share characteristics and interact.
  - Virtual: in essence or effect only.
  - Virtual Community: we interact and influence others, without causing communication costs.

- Different from
  - Virtual Reality
  - Intelligent Agent

Virtual Community: Goal

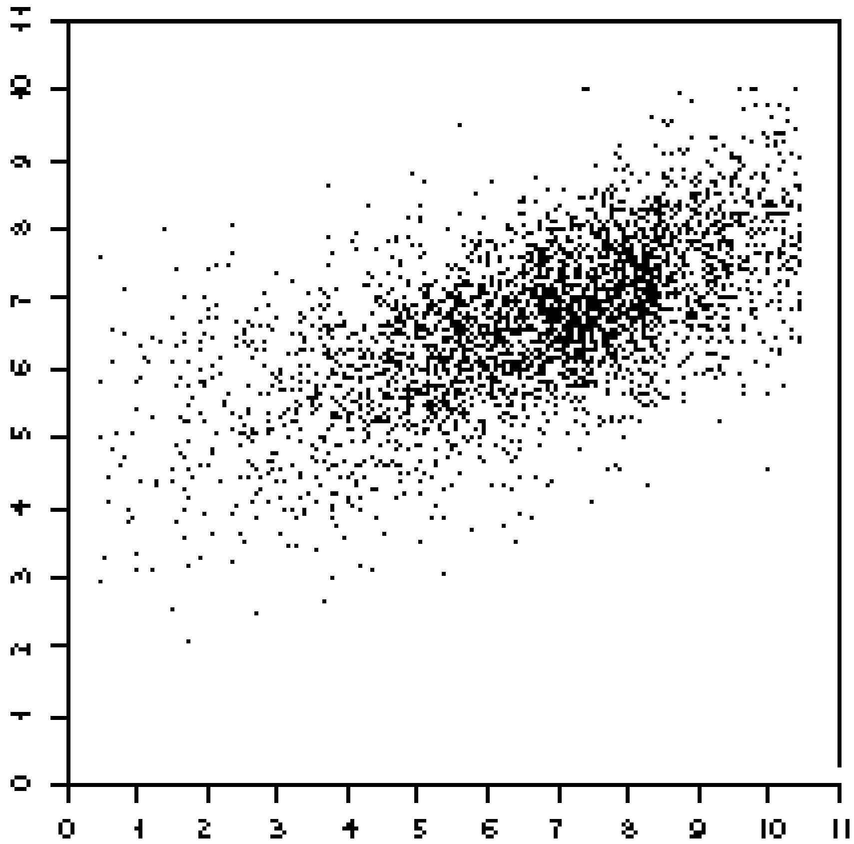
- Interface Design Goals
  - Ease of use
  - Confident recommendation
    - How much the recommendation is accurate?
  - Scalability
    - Should be able to support large amount of data.
    - With more data, better accuracy should be achieved.
  - Generalized framework
    - Not making use of domain knowledge.
    - Collaborative filtering

Virtual Community: Method

- Database structure
  - (user, movie, score)
- I/O using E-mail
  - Rating input: form filling on an e-mail, then parsed.
  - Recommendation output: send an e-mail listing must-see movies, categorical information.
- Joint recommendation
  - Good movies for two users to see together

Virtual Community: Evaluation

- ML-style: train/test set validation
  - Compare the prediction and observed rating for available data.
- HCI-style: user study
  - Gather feedback from users of the system.



r = 0.62

Virtual Community: Discussion

- HCI-perspective
  - HCI terminologies
  - Comparison with existing HCI systems
  - HCI evaluation techniques
  - Focusing on user interface development
- No explanation on recommendation algorithms
  - How to find similar users?
  - How to predict estimated rating?

Ringo

- Domain: music, artist
- Goal: personalized music recommendation, based on subject ratings of others.
  - Assumption: there are general trends and patterns within the taste of a person as well as between group of people.
- User-based collaborative filtering
  - Recommendation based on similarities between the interest profile of the active user and other users.
  - Overcome drawbacks of content-based filtering
    - Content parsing cost
    - No serendipitous finding
    - Unable to distinguish products with same features

Ringo: Method

- Procedure
  - Build and maintain user profile, from their subjective rating on items.
  - Compare the profile with other users, and find ones having similar interests.
  - Find out the list of items that those similar users like.
  - Recommend those items.

Ringo: Method

- Characteristics
  - Use absolute scale for rating.

7 :	BOOM! One of my FAVORITE few! Can't live without it.
6 :	Solid. They are up there.
5 :	Good Stuff.
4 :	Doesn't turn me on, doesn't bother me.
3 :	Eh. Not really my thing.
2 :	Barely tolerable.
1 :	Pass the earplugs.

  - Two groups of artists are included for rating request.
    - Popular artists
    - Unpopular artists
  - Users also can add new music or artists.

Ringo: Method

- Similarity calculation
  - Mean squared differences: $\overline{(U_x - U_y)^2}$
  - Pearson-r correlation: $\dfrac{\sum(U_x - \overline{U_x})(U_y - \overline{U_y})}{\sqrt{\sum(U_x - \overline{U_x})^2}\sqrt{\sum(U_y - \overline{U_y})^2}}$
  - Constrained Pearson-r correlation: $\dfrac{\sum(U_x - 4)(U_y - 4)}{\sqrt{\sum(U_x - 4)^2}\sqrt{\sum(U_y - 4)^2}}$
  - Artist-artist: item-based CF

Ringo: Evaluation

- Evaluation criteria
  - Mean absolute error (MAE): $|\overline{E}| = \frac{1}{N}\sum_{i=1}^{N} |\epsilon_i|$
  - Standard deviation of errors: $\sigma = \sqrt{\frac{\sum (E - \overline{E})^2}{N}}$
  - Prediction coverage

Ringo: Discussion

- Detailed explanation for how to implement recommendation algorithms.
- Compare user-based vs. item-based CF algorithms.
- Consider several evaluation criteria.

Any question?

THE END

Thank you very much!