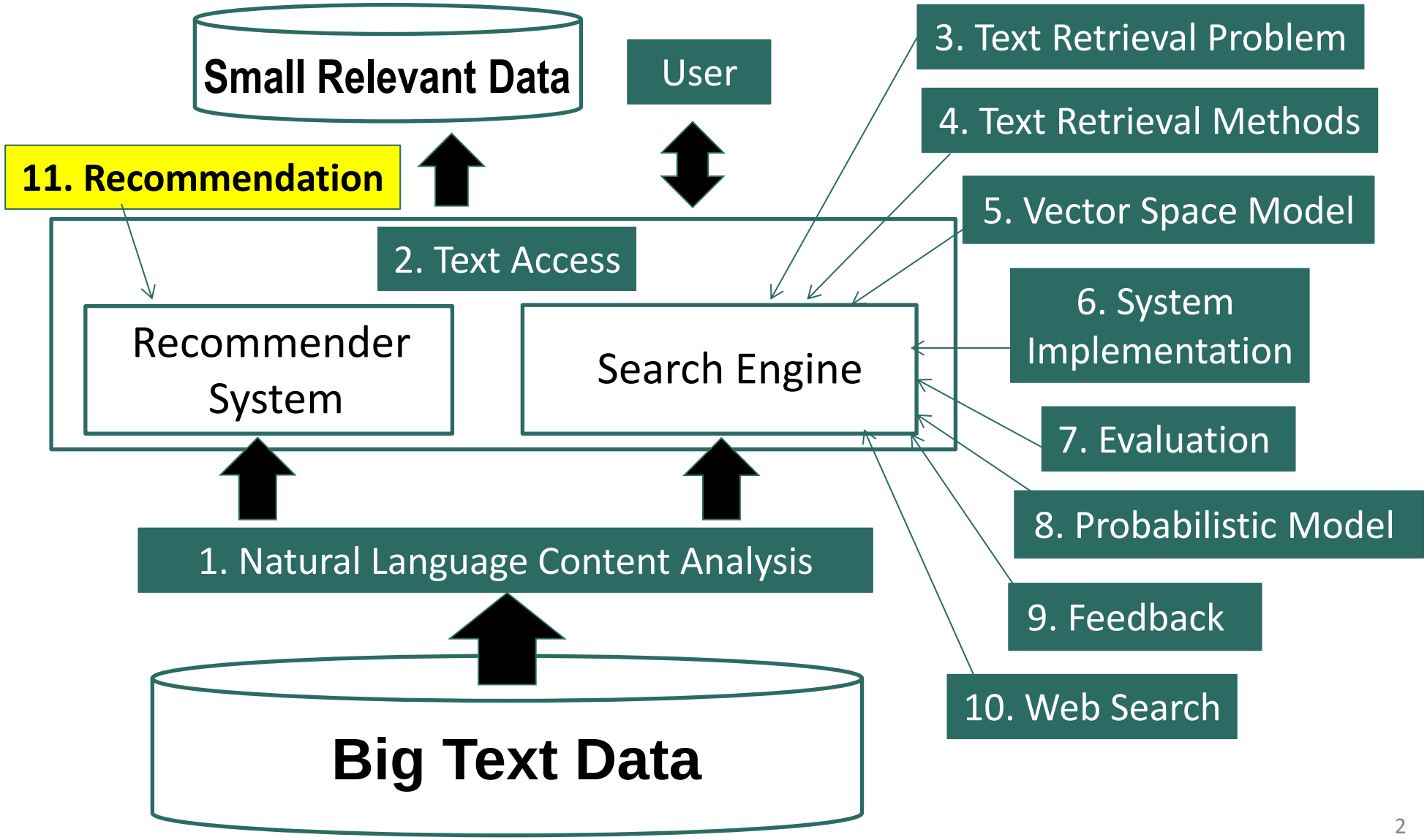
Text Retrieval and Search Engines

Recommender Systems: Collaborative Filtering - Part 1 - 2

ChengXiang "Cheng" Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign

Recommender Systems: Collaborative Filtering



Basic Filtering Question: Will user *U* like item *X*?

- Two different ways of answering it
  - Look at what items U likes, and then check if X is similar

Item similarity => content-based filtering

  - Look at who likes X, and then check if U is similar

User similarity => collaborative filtering

- Can be combined

What is Collaborative Filtering (CF)?

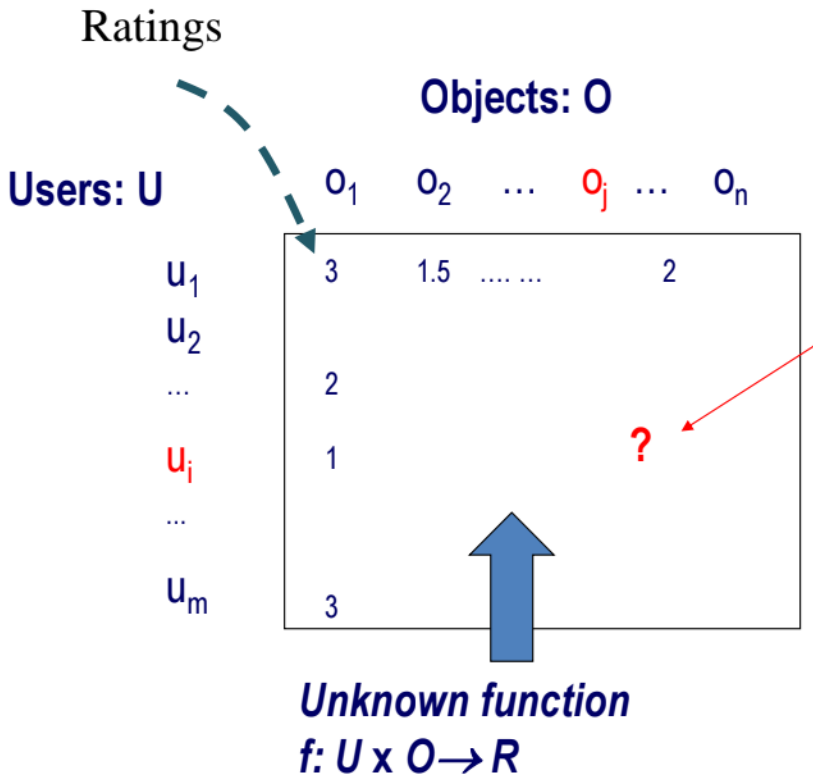
- Making filtering decisions for an individual user based on the judgments of other users
- Inferring individual's interest/preferences from that of other similar users
- General idea
  - Given a user u, find similar users $\{u_1, \ldots, u_m\}$
  - Predict u's preferences based on the preferences of $u_1, \ldots, u_m$
  - User similarity can be judged based on their similarity in preferences on a common set of items

通过与u相似的其他用户来找到u的preferences

CF: Assumptions 前提假设

- Users with the same interest will have similar preferences
- Users with similar preferences probably share the same interest
- Examples
  - "interest is information retrieval" => "favor SIGIR papers"
  - "favor SIGIR papers" => "interest is information retrieval"
- Sufficiently large number of user preferences are available (if not, there will be a "cold start" problem)

The Collaboration Filtering Problem



The task

- Assume known f values for some (u,o)'s
- Predict f values for other (u,o)'s
- Essentially function approximation, like other learning problems

Memory-based Approaches

- General ideas:
  - X_{ij}: rating of object o_j by user u_i
  - n_i: average rating of all objects by user u_i
  - Normalized ratings: $V_{ij} = X_{ij} - n_i$
  - Prediction of rating of object o_j by user u_a

$$\hat{v}_{aj} = k\sum_{i=1}^{m} w(a,i)v_{ij} \qquad \hat{x}_{aj} = \hat{v}_{aj} + n_a \qquad k = 1/\sum_{i=1}^{m} w(a,i)$$

- Specific approaches differ in w(a,i) -- the distance/similarity between user u_a and u_i

User Similarity Measures

- Pearson correlation coefficient (sum over commonly rated items)

$$w_p(a,i) = \frac{\sum_j (x_{aj} - n_a)(x_{ij} - n_i)}{\sqrt{\sum_j (x_{aj} - n_a)^2 \sum_j (x_{ij} - n_i)^2}}$$

- Cosine measure

$$w_c(a,i) = \frac{\sum_{j=1}^{n} x_{aj} x_{ij}}{\sqrt{\sum_{j=1}^{n} x_{aj}^2 \sum_{j=1}^{n} x_{ij}^2}}$$

- Many other possibilities!

Improving User Similarity Measures

- Dealing with missing values: set to default ratings (e.g., average ratings)
- Inverse User Frequency (IUF): similar to IDF

use predict values → improve more

rare item → more similarity

popular item → less similarity

Summary of Recommender Systems

- Filtering/Recommendation is "easy"
  - The user's expectation is low
  - Any recommendation is better than none
- Filtering is "hard"
  - Must make a binary decision, though ranking is also possible
  - Data sparseness (limited feedback information)
  - "Cold start" (little information about users at the beginning)
- Content-based vs. Collaborative filtering vs. Hybrid
- Recommendation can be combined with search ➔ Push + Pull
- Many advanced algorithms have been proposed to use more context information and advanced machine learning

Additional Readings

- Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor: Recommender Systems Handbook. Springer 2011.

http://www.cs.bme.hu/nagyadat/Recommender_systems_handbook.pdf