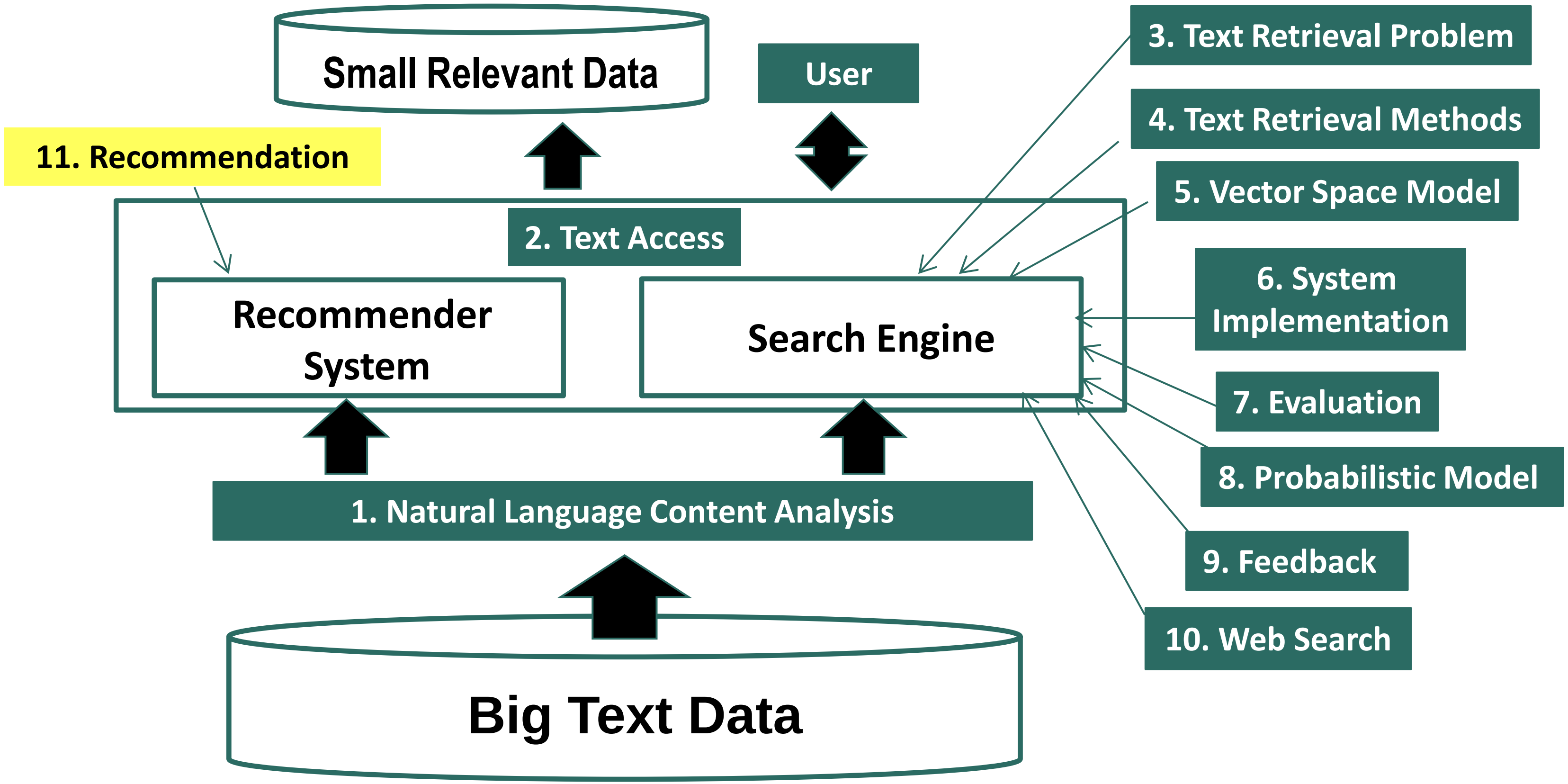
Text Retrieval and Search Engines

Recommender Systems: Content-Based Filtering - Part 1 - 2

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

Recommender Systems

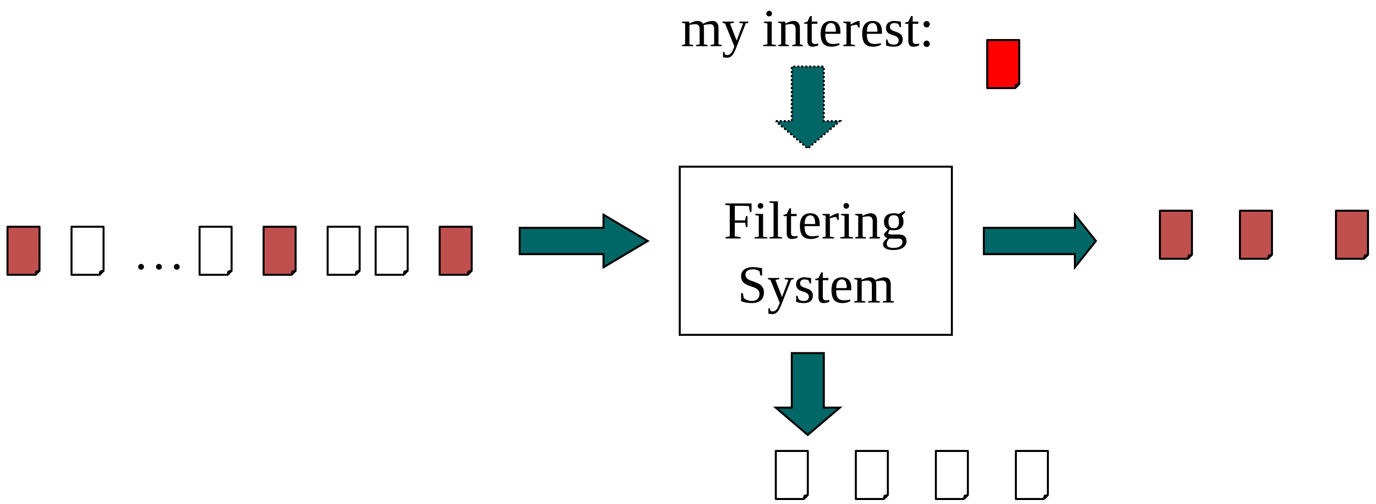


Two Modes of Text Access: Pull vs. Push

- **Pull** Mode (**search engines**)
  – Users take initiative
  – Ad hoc information need
- **Push** Mode (**recommender systems**)
  – Systems take initiative
  – Stable information need or system has good knowledge about a user’s need

Recommender ≈ Filtering System

- Stable & long term interest, dynamic info source
- System must make a delivery decision immediately as a document “arrives”



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

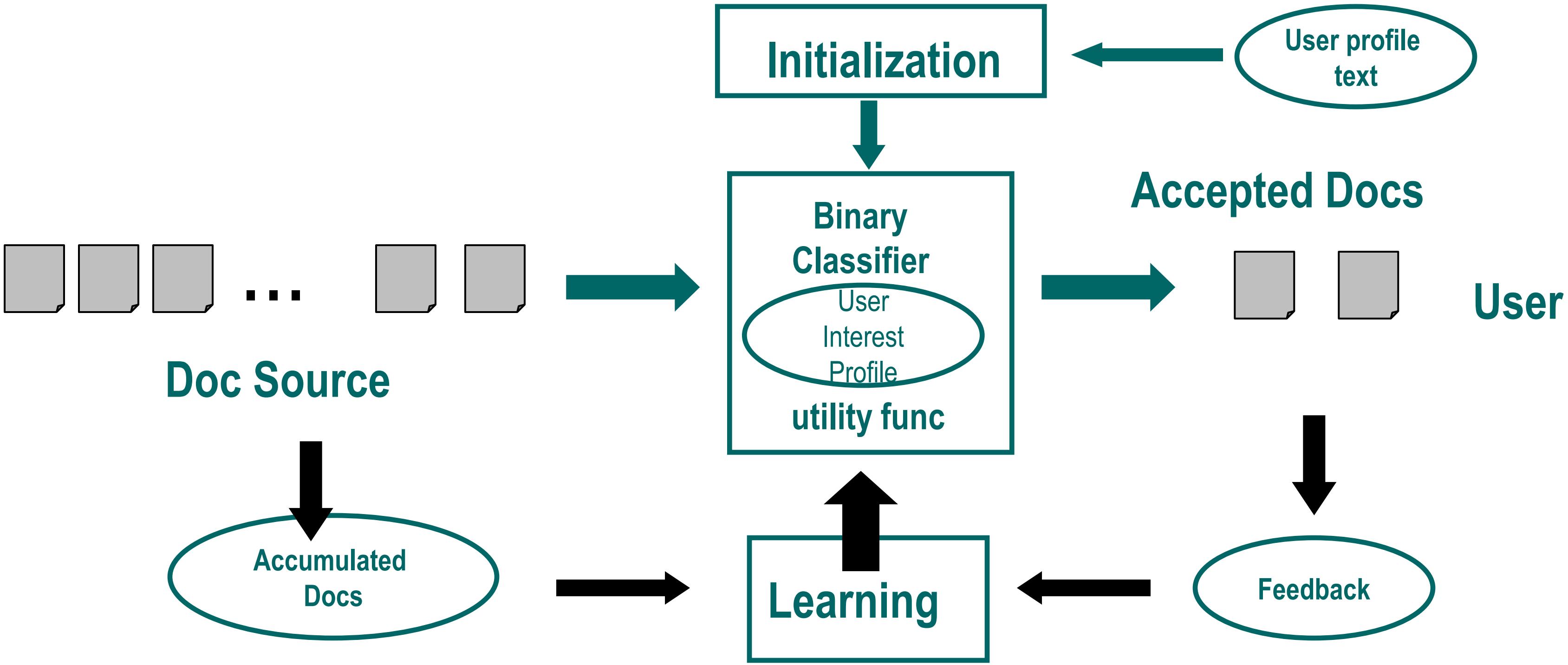
Item similarity => content-based filtering

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

User similarity => collaborative filtering

- Can be combined

A Typical Content-Based Filtering System



Linear Utility = 3* #good - 2 *#bad

#good (#bad): number of good (bad) documents delivered to user

Are the coefficients (3, -2) reasonable? What about (10, -1) or (1, -10)?

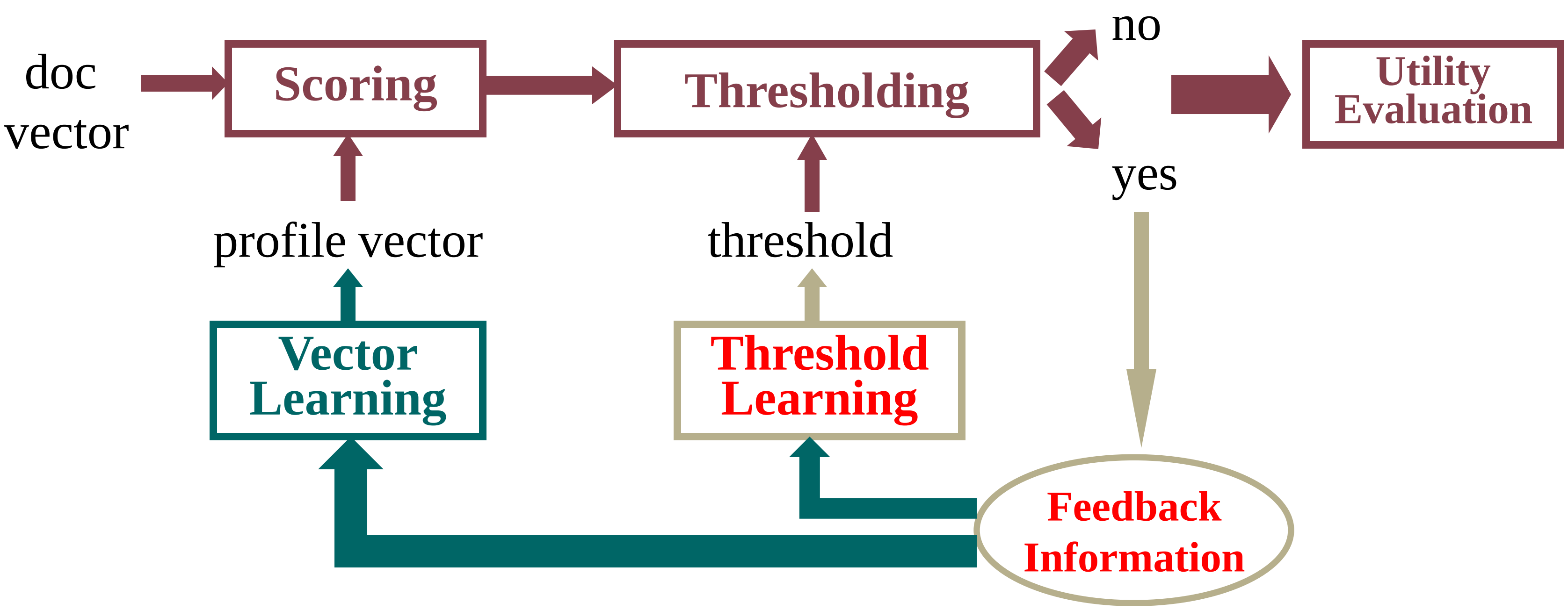
Three Basic Problems in Content-Based Filtering

- Making **filtering decision** (Binary classifier)
  - Doc text, profile text $\rightarrow$ yes/no
- **Initialization**
  - Initialize the filter based on only the profile text or very few examples
- **Learning** from
  - Limited relevance judgments (only on "yes" docs)
  - Accumulated documents
- All trying to maximize the utility

Extend a Retrieval System for Information Filtering

- “Reuse” retrieval techniques to score documents
- Use a score threshold for filtering decision
- Learn to improve scoring with traditional feedback
- New approaches to threshold setting and learning

A General Vector-Space Approach



Difficulties in Threshold Learning

36.5	Rel
33.4	NonRel
32.1	Rel
29.9	?
27.3	?
…	
...	

θ=30.0

No judgments are available for these documents

- **Censored data (judgments only available on delivered documents)**
- **Little/none labeled data**
- **Exploration vs. Exploitation**

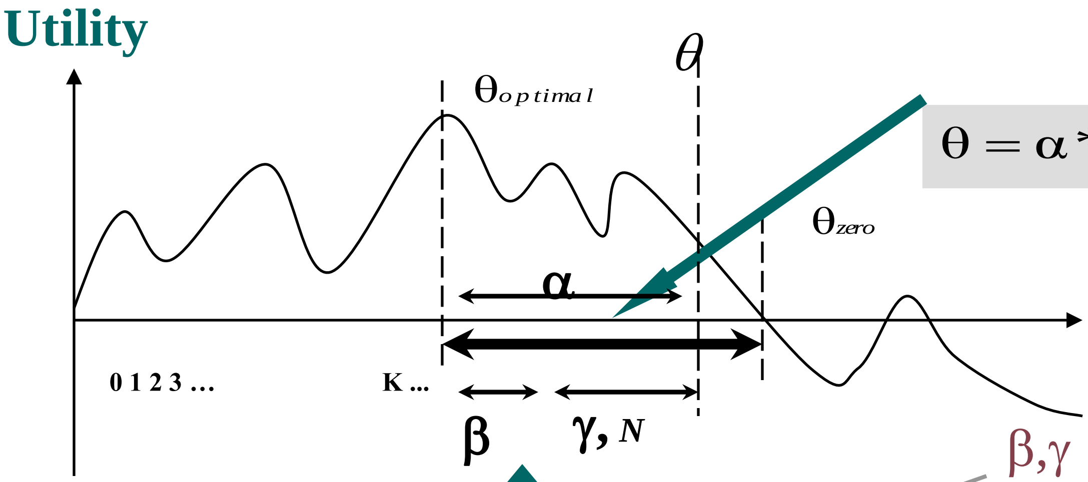
Empirical Utility Optimization

- Basic idea
  - Compute the utility on the training data for each candidate score threshold
  - Choose the threshold that gives the maximum utility on the training data set
- Difficulty: Biased training sample!
  - We can only get an upper bound for the true optimal threshold
  - Could a discarded item be possibly interesting to the user?
- Solution:
  - Heuristic adjustment (lowering) of threshold

Beta-Gamma Threshold Learning

Encourage exploration up to θ_{zero}

$$\theta = \alpha * \theta_{zero} + (1-\alpha) * \theta_{optimal}$$



Cutoff position (descending order of doc scores)

$\beta, \gamma \in [0,1]$

$$\alpha = \beta + (1-\beta) * e^{-N*\gamma}$$

$$N = \#\ training\ \ examples$$

The more examples, the less exploration (closer to $\theta_{optimal}$)

Beta-Gamma Threshold Learning (cont.)

- Pros
  - Explicitly addresses exploration-exploitation tradeoff ("Safe" exploration)
  - Arbitrary utility (with appropriate lower bound)
  - Empirically effective
- Cons
  - Purely heuristic
  - Zero utility lower bound often too conservative

Summary

- Two strategies for recommendation/filtering
  - Content-based (item similarity)
  - Collaborative filtering (user similarity)
- Content-based recommender system can be built based on a search engine system by
  - Adding threshold mechanism
  - Adding adaptive learning algorithms