Introduction to Item-Item Collaborative Filtering

Motivation

- User-User CF was great, except …
- Issues of Sparsity
  - With large item sets, small numbers of ratings, too often there are points where no recommendation can be made (for a user, for an item to a set of users, etc.)
  - Many solutions proposed here, including “filterbots”, item-item, and dimensionality reduction

Motivation (2)

- Computational performance
  - With millions of users (or more), computing all-pairs correlations is expensive
  - Even incremental approaches were expensive
  - And user profiles could change quickly – needed to compute in real time to keep users happy

Item-Item: An Alternative

- Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (WWW '01). ACM, New York, NY, USA, 285-295.

Winner of the 2016 Seoul Test of Time Award

The Item-Item Insight

- Item-Item similarity is fairly stable …
  - This is dependent on having many more users than items
    - Average item has many more ratings than an average user
    - Intuitively, items don't generally change rapidly – at least not in ratings space (special case for time-bound items)
- Item similarity is a route to computing a prediction of a user's item preference

A little more detail …

- Two step process:
  - Compute similarity between pairs of items
    - Correlation between rating vectors
      - co-rated cases only (only useful for multi-level ratings)
    - Cosine of item rating vectors
      - can be used with multi-level or unary ratings
      - or adjusted ratings (normalize before computing cosine)
    - Some use conditional probability (unary)
  - Predict user-item rating
    - Weighted sum of rated "item-neighbors"
    - Linear regression to estimate rating

Item-Item Top-N

- Item-Item similarity model can be used to compute top-N directly:
  - Simplify model by limiting items to small "neighborhoods" of k most-similar items (e.g., 20)
  - For a profile set of items, compute/merge/sort the k-most similar items for each profile item
    - Straightforward matrix operation from Deshpande and Karypis (Mukund Deshpande and George Karypis. 2004. Item-based top-*N* recommendation algorithms.*ACM Trans. Inf. Syst.* 22, 1 (January 2004), 143-177.)

Benefits of Item-Item

- It actually works quite well
  - Good prediction accuracy
  - Good performance on top-N predictions
- Efficient implementation
  - At least in cases where |U| >> |I|
  - Benefits of precomputability
- Broad applicability and flexibility
  - As easy to apply to a shopping cart as to a user profile

Core Assumptions/Limitations

- Item-item relationships need to be stable …
  - Mostly a corollary of stable user preferences
  - Could have special cases that are difficult (e.g., calendars, short-lived books, etc.)
  - Many of these issues are general temporal issues
- Main limitation/complaint: lower serendipity
  - This is a user/researcher complaint, not fully studied; intuition is clear

Moving Forward

- Next Lectures
  - Breaking down the core item-item algorithm
  - Looking at the special cases of implicit feedback (unary ratings)
  - Extending item-item to incorporate other sources of information
  - Item-item in practice – retracing the industry history
  - Programming item-item (honors track)

Learning Objectives

After completing this module, you should be able to:

- Explain the concept and algorithm for item-item collaborative filtering, including the key tuning parameters and its strengths and weaknesses.
- Implement item-item collaborative filtering, both manually on small datasets (all learners) and by programming it (honors track).
- Identify the relative strengths and weaknesses of user-based and item-based algorithms, and identify which algorithm is a better fit for a particular use case.

Assignments and Assessments

- Spreadsheet IICF Assignment
- Module Quiz
- LensKit IICF Programming Assignment (Honors Track)

Introduction to
Item-Item Collaborative Filtering