Recommender Systems

Implementing Collaborative Filtering

Mining of Massive Datasets
Leskovec, Rajaraman, and Ullman
Stanford University

Collaborative Filtering: Complexity

- Expensive step is finding ***k*** most similar users (or items): **O(|U|)**
  - |U| = size of utility matrix
- Too expensive to do at runtime
  - Could pre-compute
  - Naïve pre-computation takes time **O(n ·|U|)**
    - Where n = number of users (items)
- **We already know how to do this!**
  - Near-neighbor search in high dimensions (**LSH**)
  - Clustering
  - Dimensionality reduction (coming soon!)

Pros/Cons of Collaborative Filtering

- **+ Works for any kind of item**
  - No feature selection needed
- **- Cold Start:**
  - Need enough users in the system to find a match
- **- Sparsity:**
  - The user/ratings matrix is sparse
  - Hard to find users that have rated the same items
- **- First rater:**
  - Cannot recommend an unrated item
  - New items, Esoteric items
- **- Popularity bias:**
  - Tends to recommend popular items

Hybrid Methods

- **Add content-based methods to collaborative filtering**
  - Item profiles for new item problem
  - Demographics to deal with new user problem
- **Implement two or more different recommenders and combine predictions**
  - Perhaps using a linear model
  - Example: global baseline + collaborative filtering

Global Baseline Estimate

- Estimate Joe's rating for the movie *The Sixth Sense*
  - Problem: Joe has not rated any movie "similar" to *The Sixth Sense*

- Global Baseline approach
  - Mean movie rating: **3.7 stars**
  - *The Sixth Sense* is **0.5** stars above avg.
  - Joe rates **0.2** stars below avg.
  - **Baseline estimate: 3.7 + 0.5 – 0.2 = 4 stars**

Combining Global Baseline with CF

- Global Baseline estimate
  - Joe will give *The Sixth Sense* 4 stars
- Local neighborhood (CF/NN)
  - Joe didn't like related movie *Signs*
  - Rated it 1 star below his average rating
- Final estimate
  - Joe will rate *The Sixth Sense* 4 -1 = 3.5 stars

CF: Common Practice

Before:

$$r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

- Define **similarity s_{ij}** of items ***i*** and ***j***
- Select ***k*** nearest neighbors ***N(i; x)***
  - Items most similar to ***i***, that were rated by ***x***
- Estimate rating r_{xi} as the weighted average:

$$r_{xi} = b_{xi} + \frac{\sum_{j \in N(i;x)} s_{ij} \cdot (r_{xj} - b_{xj})}{\sum_{j \in N(i;x)} s_{ij}}$$

baseline estimate for r_{xi}

$b_{xi} = \mu + b_x + b_i$

- μ = overall mean movie rating
- b_x = rating deviation of user ***x***
  = (*avg. rating of user **x***) – μ
- b_i = rating deviation of movie ***i***