Implicit Feedback and Cold-Start in Collaborative Filtering

Recommender Systems

Dimitris Sacharidis

Dealing with Implicit Feedback

Explicit Feedback

- so far we've dealt with ratings on some scale
  - e.g., on a five-star scale, thumbs up/down

- ratings are considered **explicit feedback**
  - because users directly express their opinion to items
- strongest **signal** to build recommendations upon
  - but sometimes rare

Implicit Feedback

- often we need to make recommendations based on other *weaker signals,* but more *abundant*

- **implicit feedback** is a term for all other things we can collect about a user-item pair: e.g.,
  - consumption/purchase data (why is this a weaker signal?)
  - other related actions, e.g., put on a cart/queue
  - clicks, page views, time spent on page

- challenge: (not going to be addressed here)
  - we don't know exactly what implicit feedback means
    - user saw the movie, but did s/he like it?

modeling Implicit Feedback

- implicit feedback is typically on a **binary** scale
  - **1** if there was a user-item interaction,
  - and **0** most of the time

- but also on **count** scales
  - e.g., number of plays, clicks, purchases
- or on a **progress** scale
  - e.g., how long a song is played, watched episodes from a series

- can be represented as a sparse **interaction matrix** (also called click/rating/etc. matrix)
  - with 0s ignored

interpretation of Implicit Feedback

- what does a **1** mean?
  - does the user actually like the item?
    - we *don't* know! implicit feedback is a weaker signal
    - but it's reasonable to assume that 1 is **positive feedback**
      - and **counts** allow us to estimate our *confidence*

- what does a **0** mean?
  - does the user not like the item?
    - we don't know! user might not be aware of the item (same as with ratings)
    - we might assume that 0 is **negative feedback**

computing similarities over Implicit Feedback

- mean-centering is *not* meaningful for implicit feedback data

- what we usually do is **normalize** user vectors
  - make them have a unit norm
  - the reason is to *discount* feedback from users who provide lots of it
  - also when large counts are typical (e.g., for song plays), we might work with the log of counts

- **cosine similarity** works well over the normalized user vectors

- can also consider non-symmetric similarity functions
  - conditional probabilities; connection to association rules
  - see [2004 TOIS M. Deshpande, G. Karypis]

[2004 TOIS M. Deshpande, G. Karypis] *Item-Based Top-N Recommendation Algorithms*

prediction formula for binary data

- a note for binary scales on item-item CF

- weighted average over **1**s does not make sense
  - why? you always get a result of 1

- instead, compute the average weight in the neighborhood:

$$s(u,i) = \frac{\sum_{j \in N(i)} w_{ij}}{|N(i)|}$$

Cold Start Problems

cold start problems

- several problems for CF recommenders
  - **new user**
  - **new item**
  - **new system**

new user

- **no rating history** for a new user
  - zero similarity with other users (for U-U)
  - no similar items exist (for I-I)

- what can we do?
  - make *non-personalized* recommendations: popular items, demographics
  - gradually learn from *implicit feedback*: browsing history
  - *elicit preferences* upon registering new user
    - e.g., movie genres, topic tags
  - social-based recommendations: learn user's friends, from social network, referrals

new item

- **no rating history** for a new item
  - need to collect feedback for it; less of a problem

- what can we do?
  - content-based approaches: similarities based on item descriptions
  - recommend to random users, or a well-chosen set of users:
    - early adopters
    - influencers
    - with broad taste
    - tolerant to potentially bad recommendations

new system

- no users and no items
  - worst of both cases!

- what can we do?
  - rely on data from outside sources
  - knowledge-based recommenders: given item attributes, ask for user preferences
  - slowly transition to CF-based recommenders
    - when do you have enough data?

Acknowledgements

some ideas from:

- Joseph A. Konstan, Michael D. Ekstrand. Recommender Systems Specialization on coursera