NYU | Center for Data Science

# Week 10.2: Collaborative filters

DS-GA 1004: Big Data

# Collaborative filtering

- ***Utility matrix* (R)**: feedback for sparsely observed interactions

Items

Users

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1 | | | 1 | | |
| | | | | 0 | 0 | | 1 |
| | 1 | 1 | | | 1 | | |
| | | 1 | | 0 | | | |

- **Task**: predict the missing entries
- **Evaluation**: depends on the feedback mechanism

# Neighborhood models

Items

Users

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1 | | | 1 | | |
| | | | | 0 | 0 | | 1 |
| | 1 | 1 | | | 1 | | |
| | | 1 | | 0 | | | |

- **User-based** model:
  - Given a user *u*, find the most similar users {*u'*}
  - (Similar rows of the utility matrix)
  - Predict items *v* with high feedback by similar users, not yet consumed by *u*

See [Su and Khoshgoftaar, 2009] for an overview of methods

# Neighborhood models

Items

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Users | | | 1 | | | 1 | | |
| | | | | | 0 | 0 | | 1 |
| | | 1 | 1 | | | 1 | | |
| | | | 1 | | 0 | | | |

- **User-based** model:
  - Given a user *u*, find the most similar users {*u*'}
  - (Similar rows of the utility matrix)
  - Predict items *v* with high feedback by similar users, not yet consumed by *u*

- **Item-based** model:
  - Find items *v'* similar to those consumed by *u*
  - (Similar columns of the utility matrix)
  - Predict those which have not yet been consumed by *u*

See [Su and Khoshgoftaar, 2009] for an overview of methods

# Neighborhood models

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1 | | | 1 | | |
| | | | | 0 | 0 | | 1 |
| | 1 | 1 | | | 1 | | |
| | | 1 | | 0 | | | |

- Conceptually simple, but the details matter!
    - How do you define ***similarity*** between users?  Between items?
    - How do you **aggregate feedback** over the neighborhood?
- Depending on feedback type, can be difficult to scale
    - Binary feedback ⇒ Jaccard similarity, MinHash+LSH will work
    - Otherwise … ?  Most spatial data structures are not robust to missing features!

# Latent factor models

- Flexible framework for feedback modeling
  - Objective can be **tuned to match feedback** mechanism (e.g., ★★★☆☆ vs play counts)
  - Secondary objectives can be added (item bias, regularization, etc)
- Usually easy to **parallelize** and scale up training
  - E.g.: alternating least squares.
  - Users are independent (conditional on items), and vice versa
- Learned representation is **low-rank** and dense
  - Integrates well with spatial data structures
  - Rank parameter provides control on complexity ⇔ expressivity

# Modeling implicit feedback

- Count data is informative, but hard to predict
  - And we don't really care about that anyway!

- Instead, predict **binary interaction**, but use **counts to weight** terms!

$$\min_{U,V} \sum_{(i,j)\in\Omega} c_{ij}(p_{ij} - \langle U_i, V_j \rangle)^2$$

$$p_{ij} = \begin{cases} 1 & R_{ij} > 0 \\ 0 & R_{ij} = 0 \end{cases}$$

$$c_{ij} = 1 + \alpha R_{ij}$$

[Hu, Koren, Volinsky 2008]

# Modeling implicit feedback

- Count data is informative, but hard to predict
  - And we don't really care about that anyway!

- Instead, predict **binary interaction**, but use **counts to weight** terms!

$$\min_{U,V} \sum_{(i,j)\in\Omega} c_{ij}(p_{ij} - \langle U_i, V_j \rangle)^2$$

$$p_{ij} = \begin{cases} 1 & R_{ij} > 0 \\ 0 & R_{ij} = 0 \end{cases}$$

$$c_{ij} = 1 + \alpha R_{ij}$$

- You don't have to use R directly...
  - Drop low counts
  - Compress large values $R_{ij} \rightarrow \log(1 + R_{ij})$

[Hu, Koren, Volinsky 2008]

# Handling new items

- CF gives **no representation** to items with **no interactions**
- ⇒ A new item will never be recommended until it has representation!
- This is known as the ***cold-start*** problem
- Solutions typically involve
  - Active promotion / manual curation
  - Content-based modeling

# Content-based models

Suppose you have observed features $x_j$ for each item

- News ⇒ topics, location, source
- Movies ⇒ genre, year, length, language, director, actors, …
- Music ⇒ metadata + acoustic attributes

- **Content-based model**
  - Each user $i$ gets their own interaction model $u_i$

$$R_{ij} \approx \langle u_i, x_j \rangle$$

  - Like LF model, but the item factors are **explicit**
  - Can be limiting / over-constrained

# Content-based models

Suppose you have observed features $\boldsymbol{x}_j$ for each item

- News ⇒ topics, location, source
- Movies ⇒ genre, year, length, language, director, actors, …
- Music ⇒ metadata + acoustic attributes

- **Content-based model**
  - Each user $i$ gets their own interaction model $u_i$

$$R_{ij} \approx \langle u_i, x_j \rangle$$

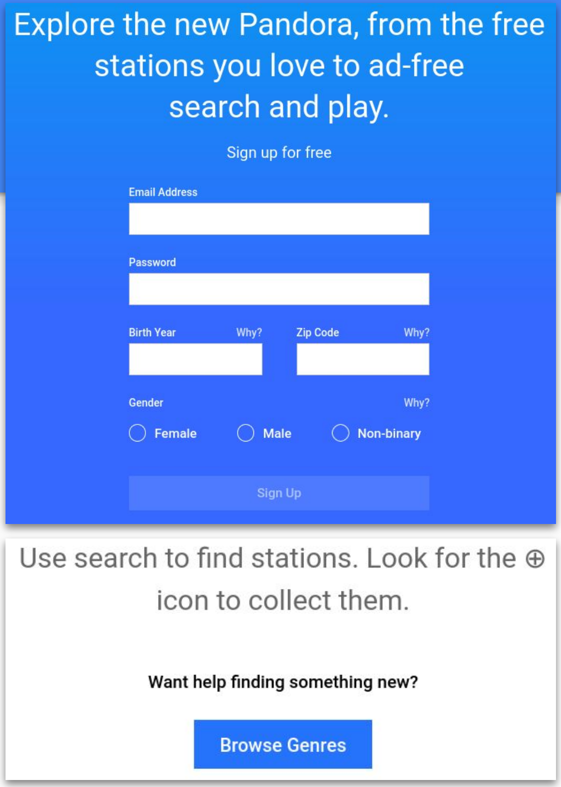
  - Like LF model, but the item factors are **explicit**
  - Can be limiting / over-constrained
- **Content cold-start**
  - Train a LF model as before, item $j \rightarrow V_j$
  - Then regress item factors from features

$$V_j \approx \boldsymbol{f}(x_j)$$

  - A new item can map into embedding space by learned mapping $\boldsymbol{f}(x_k)$ (e.g. linear regression)

# User warm-start

- What happens when a new user enters a system?
- Typical systems request some **demographic data**, and ask for **examples of things you like**
- These data are used to position you in the collaborative filter




Pandora.com, 2022

# Summary

part 2

- Collaborative filtering algorithms estimate the missing entries of the utility matrix $R$
- Recommendations are (usually) formed by selecting interactions with high (estimated) utility
- How do we know if it works?

**… come back for part 3!**