DMML, 11 Apr 2019

Recommendation Systems

Amazon suggestions

Netflix ...

Make suggestions that are useful

- Restaurants, books, movies ...

- Novelty – new items related to your tastes

- Serendipity – unknown items that you may like

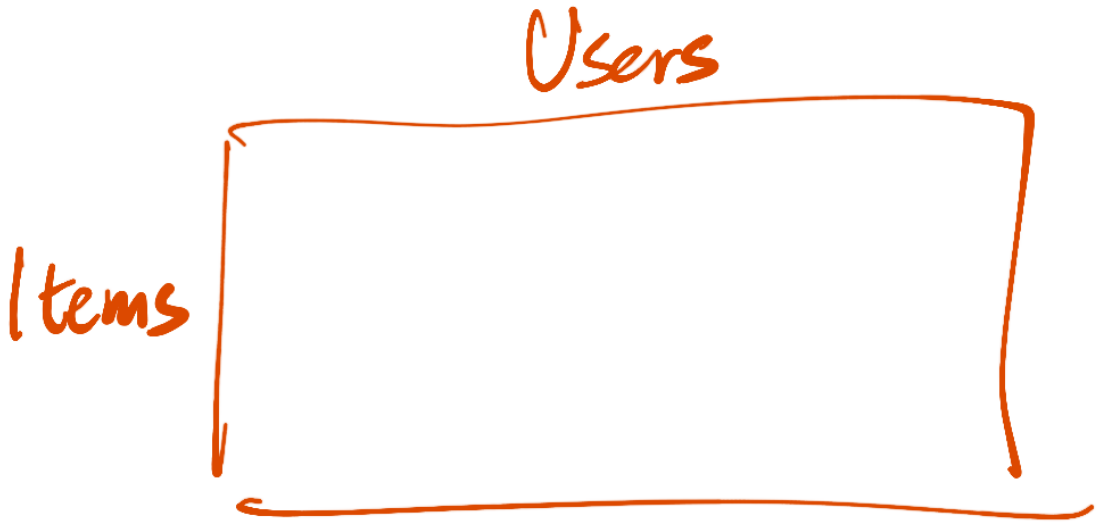
How to build a recommendation system?

1. Collaborative filtering
   - Recommend based on similarities between items / users

2. Content-based
   - Metadata about items to match
   - User profile
     - Novelty? Serendipity?

Collaborative filtering

Rating Matrix



Entries are ratings

- Numeric score - no constraint
- From a restricted range (1-5)
- Categorical - Good / Bad ..

Subcase: 0/1

Binary – 1 is good, 0 is bad

Unary – only 1 is meaningful

└ e.g. buying a product
following a link

Implicit rating

Similar rows/columns

Cosine measure

Pearson coefficient

items rated by both

rating

mean

$$\text{Pearson}(u,v) = \frac{\sum_{k \in I_u \cap I_v} (r_{uk} - \mu_u) \cdot (r_{vk} - \mu_v)}{\sqrt{\sum_{k \in I_u \cap I_v} (r_{uk} - \mu_u)^2} \cdot \sqrt{\sum_{k \in I_u \cap I_v} (r_{vk} - \mu_v)^2}}$$

Users u, v

Adjust ratings wrt average to take into account different rating profiles

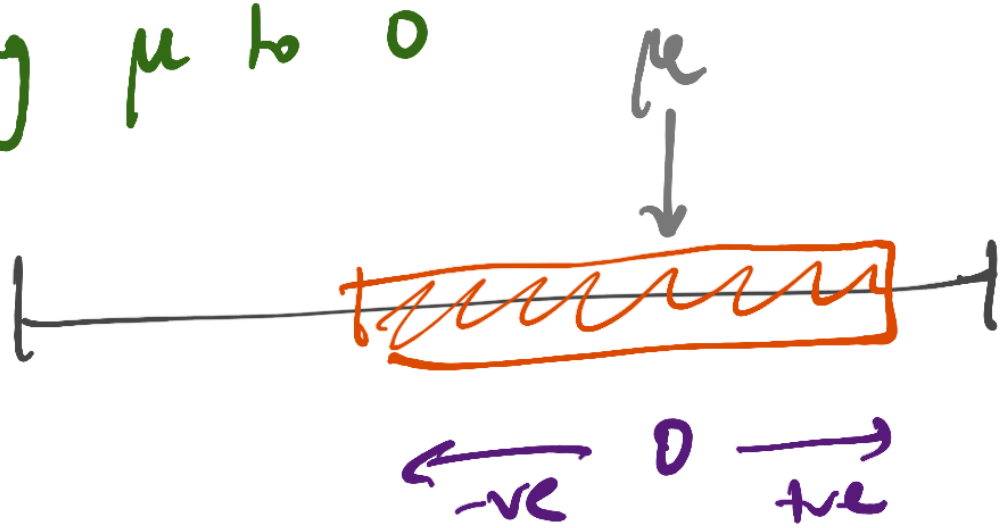
What are μ_u, μ_v?

Ideally wrt $I_u \cap I_v$ - n^2 means!

For simplicity, μ_u is across all of I_u

Can also do adjustments beforehand

- For a user, shift all values by setting μ to 0



1. Rating matrix has missing entries
   - Recommendation ⇒ fill in missing entry
2. Sparse

Fill in missing entry:

1. User-user

   - Identify similar users who have rated this item
   - Extrapolate their rating
   - Simpler problem - rank options

Compute similar users by comparing columns (Pearson coefficient --)

Offline - maintain a static list of top-k similar users for each user, update periodically

2. Item-item

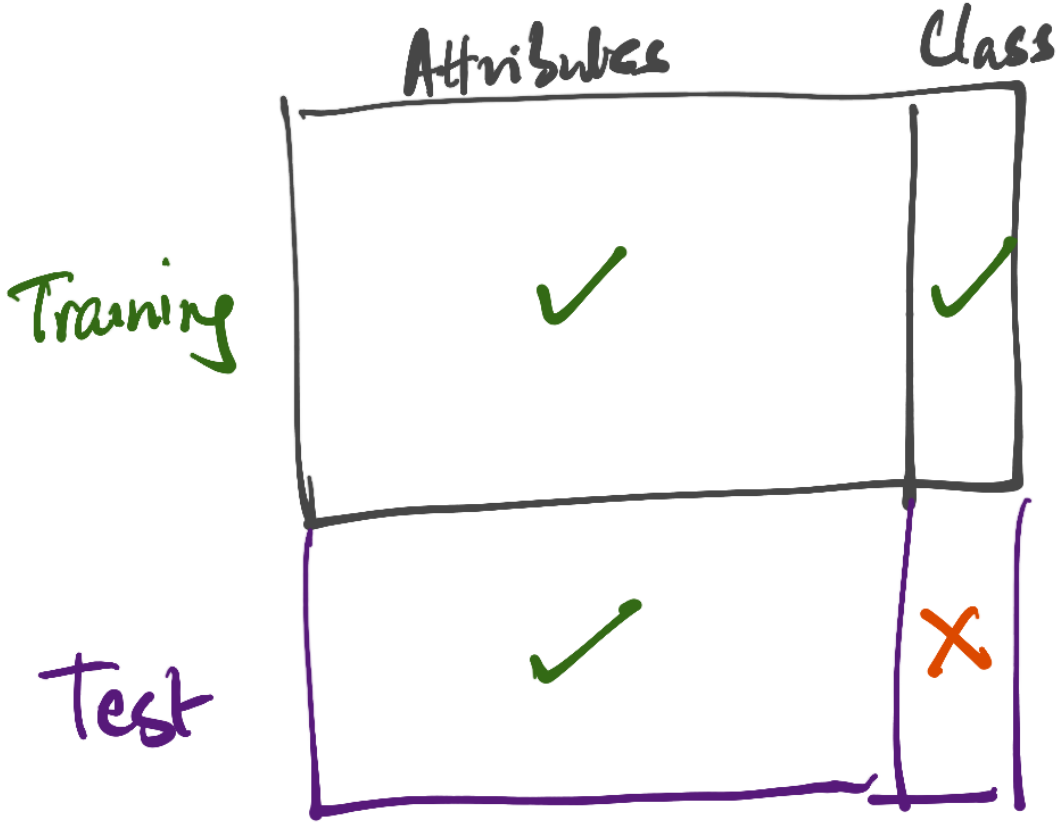
Want to predict $R(i, u)$

Similar items (similar rows)

Extrapolate user u's rating from similar ratings

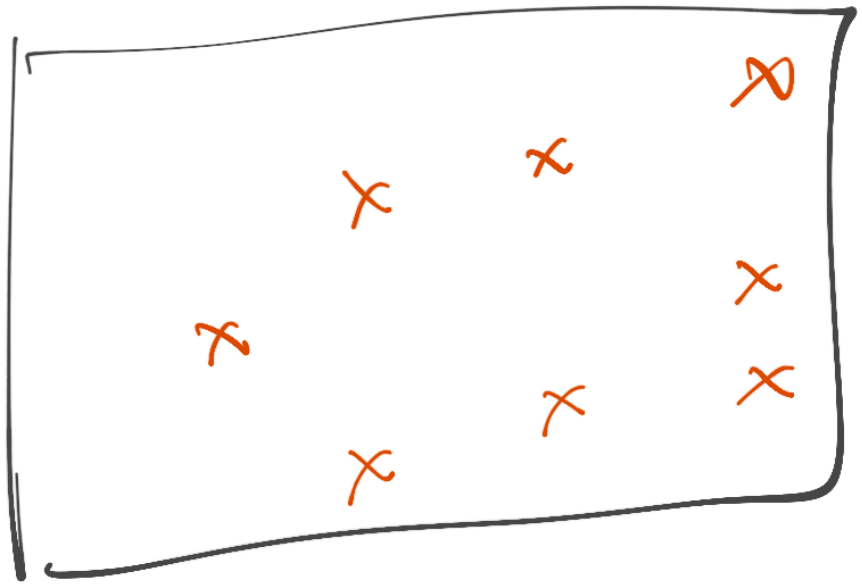
Note: Are ratings time-insensitive?

Connections to classification



Fill in missing values

Recommender



Generic way to apply a classification model.

1. Somehow make a guess for missing values
2. Run classification model for each missing entry
3. Repeat until convergence

Complexity of the problem — rows/columns

Apply SVD or equivalent to merge items/users and "rotate" axes

Latent factor analysis

State of the art

Netflix challenge