15-388/688 - Practical Data Science: Recommender systems

Pat Virtue
Carnegie Mellon University
Spring 2022

Slide credits: CMU AI, Zico Kolter

Outline

Recommender systems

Collaborative filtering

User-user and item-item approaches

Matrix factorization

Outline

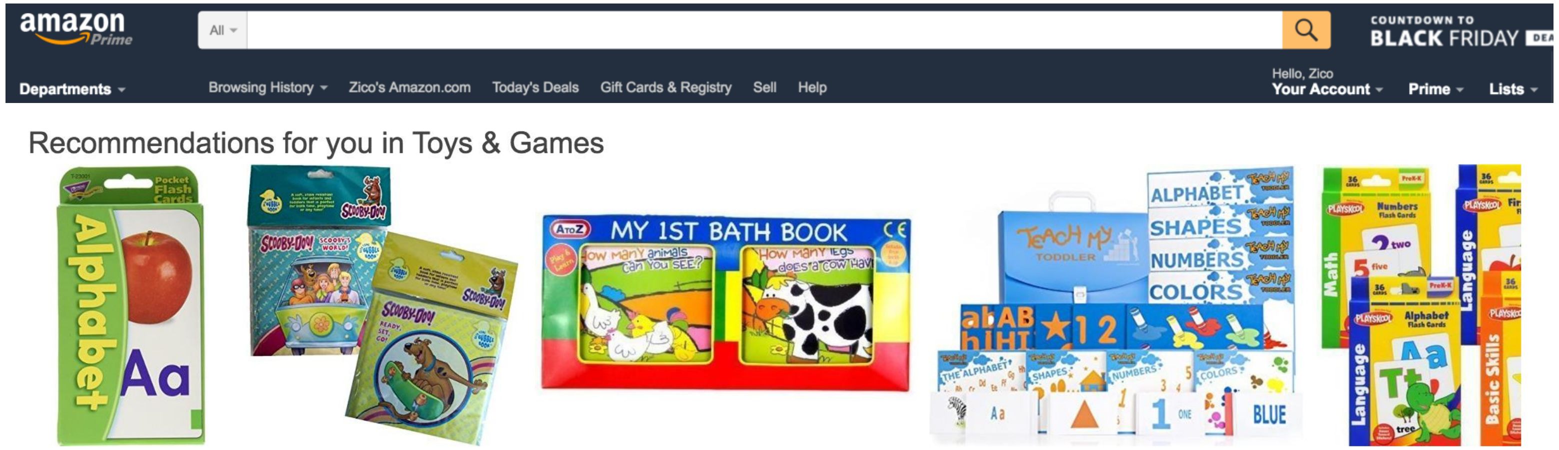
Recommender systems

Collaborative filtering

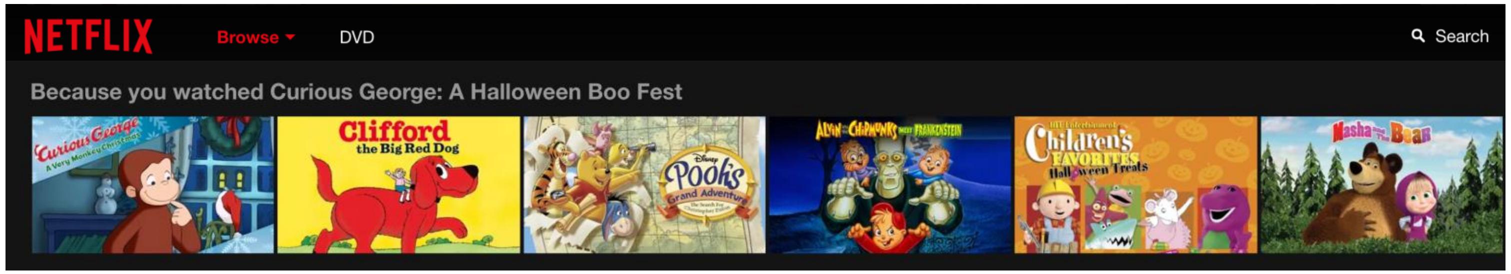
User-user and item-item approaches

Matrix factorization

Recommender systems



Recommendations for you in Toys & Games



Because you watched Curious George: A Halloween Boo Fest

Information we can use to make predictions

"Pure" user information:

- Age
- Location
- Profession

"Pure" item information:

- Movie budget
- Main actors
- (Whether it is a Netflix release)

User-item information:

- Which items are most similar to those I have bought before?
- What items have users most similar to me bought?

Supervised or unsupervised?

Do recommender systems fit more within the “supervised” or “unsupervised” setting?

Like supervised learning, there are known outputs (items that the uses purchases), but like unsupervised learning, we want to find structure/similarity between users/items

We won’t worry about classifying this as just one or the other, but we will again formulate the problem within the three elements of a machine learning algorithm: 1) hypothesis function, 2) loss function, 3) optimization

Challenges in recommender systems

There are many challenges beyond what we will consider here in recommender systems:

1. Lack of user ratings / only “presence” data

2. Balancing personalization with generic “good” items

3. Privacy concerns

Historical note: Netflix Prize

Public competition ran from 2006 to 2009, goal was to produce a recommender system with 10% improvement in RMSE over existing Netflix system (based upon item-item Pearson correlation plus linear regression), $1M prize

Sparked a great deal of research in collaborative filtering, especially matrix factorization techniques

Larger impacts: put “data science competitions” in the public eye, emphasized practical importance of ensemble methods (though winning solution was never fielded)

Outline

Collaborative filtering

Collaborative filtering refers to recommender systems that make recommendations *based solely upon the preferences that other users have indicated for these item (e.g., past ratings)*

The mathematical setting to have in mind in that of a matrix with mostly unknown entries

$$X = \underbrace{\begin{bmatrix} 1 & & & 3 \\ & 2 & 5 & \\ & 3 & & 5 \\ 4 & & 4 & \end{bmatrix}}_{\text{columns correspond to different items}} \left.\vphantom{\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}}\right\} \text{rows correspond to different users}$$

entries correspond to *known* (given by user) scores for that user, for that items

Matrix view of collaborative filtering

Collaborative filtering X matrix is *sparse*, but unknown entries do not correspond to zero, are just missing

Goal is to “fill in” the missing entries of the matrix

$$X = \begin{bmatrix} 1 & ? & ? & 3 \\ ? & 2 & 5 & ? \\ ? & 3 & ? & 5 \\ 4 & ? & 4 & ? \end{bmatrix}$$

Approaches to collaborative filtering

User – user approaches: find the users that are most similar to myself (based upon only those items that are rated for *both* of us), and predict scores for other items based upon the average

Item – item approaches: find the items most similar to a given item (based upon all users rated both items), and predict scores for other users based upon the average

Matrix factorization approaches: find some low-rank decomposition of the X matrix that agrees at observed values

Outline

Recommender systems

Collaborative filtering

User-user and item-item approaches

Matrix factorization

User-user and item-item approaches

Basic intuition of user-user approach: find other users who are similar to me, e.g. by correlation coefficient or cosine similarity, look at how they ranked other items that I did not rank

One difference: correlation coefficient, etc, are only defined for vectors of the same size, so we only typically compute correlation across items that both users ranked

$$X = \begin{bmatrix} 1 & ? & ? & 3 \\ ? & 2 & 5 & ? \\ ? & 3 & ? & 5 \\ 4 & ? & 1 & 2 \\ ? & 3 & ? & 3 \\ 2 & ? & 4 & 3 \end{bmatrix}$$

Item-item approaches do the same thing but by column instead of row

Outline

Recommender systems

Collaborative filtering

User-user and item-item approaches

Matrix factorization

Matrix factorization approach

Learn to map users and items to the same k-dimensional space

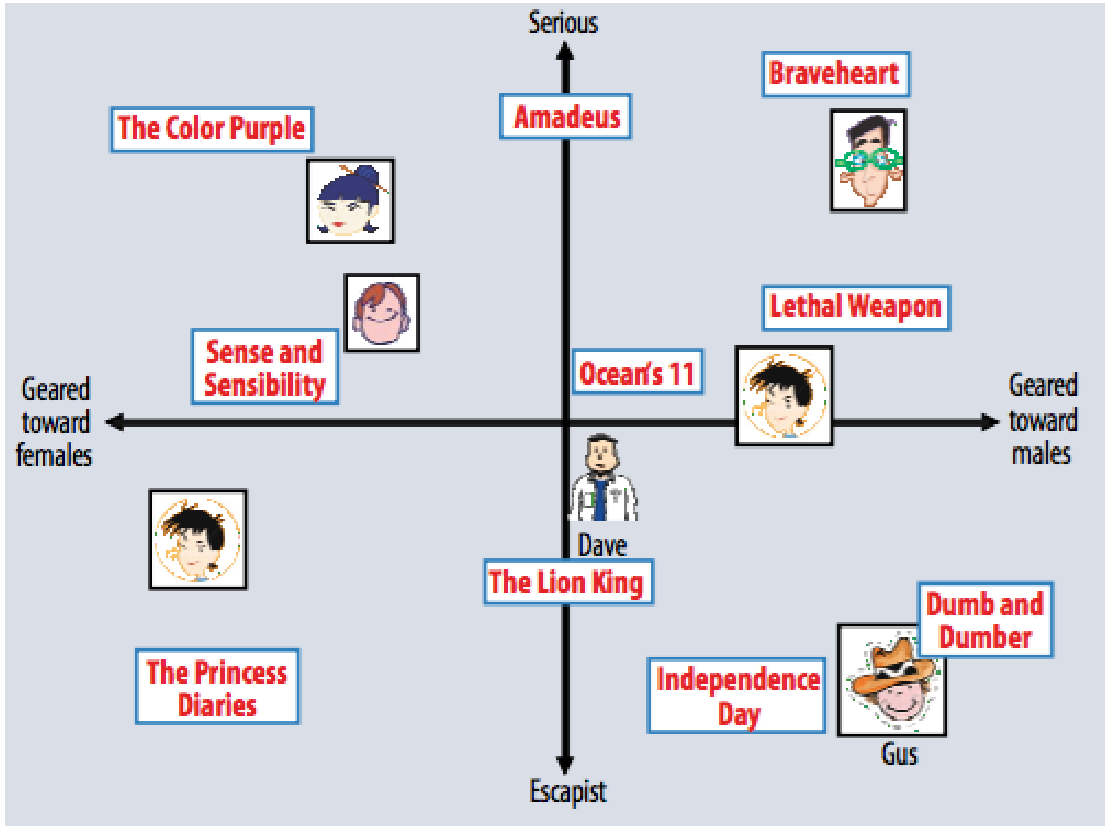


Figure: Koren et al. (2009)

Matrix factorization approach

Approximate the i, j entry of $X \in \mathbb{R}^{m \times n}$ as $\hat{X}_{ij} = u_i^T v_j$ where $u_i \in \mathbb{R}^k$ denotes user-specific weights and $v_j \in \mathbb{R}^k$ denotes item-specific weights

1. Hypothesis function
$$\hat{X}_{ij} = h_\theta(i,j) = u_i^T v_j, \qquad \theta = \{u_{1:m}, v_{1:n}\}$$

2. Loss function: squared error (on observed entries)
$$\ell\left(h_\theta(i,j), X_{ij}\right) = \left(h_\theta(i,j) - X_{ij}\right)^2$$

leads to optimization problem (S denotes set of observed entries)
$$\underset{\theta}{\text{minimize}} \sum_{i,j \in S} \ell\left(h_\theta(i,j), X_{ij}\right)$$

Optimization approaches

3. How do we optimize the matrix factorization objective? (Like k-means, EM, possibility of local optima)

Consider the objective with respect to a *single* u_i term:

$$\underset{u_i}{\text{minimize}} \sum_{j:(i,j)\in S} \left(v_j^T u_i - X_{ij}\right)^2$$

This is just a least-squares problem, can solve analytically:

$$u_i = \left(\sum_{j:(i,j)\in S} v_j v_j^T\right)^{-1} \left(\sum_{j:(i,j)\in S} v_j X_{ij}\right)$$

Alternating minimization algorithm: Repeatedly solve for all u_i for each user, v_j for each item (may not give global optimum)

Matrix factorization interpretation

What we are effectively doing here is factorizing X as a low rank matrix

$$X \approx UV, \qquad U \in \mathbb{R}^{m\times k}, V \in \mathbb{R}^{k\times n}$$

where

$$U = \begin{bmatrix} - \; u_1^T \; - \\ \vdots \\ - \; u_m^T \; - \end{bmatrix}, \qquad V = \begin{bmatrix} | & & | \\ v_1 & \cdots & v_n \\ | & & | \end{bmatrix}$$

However, we are *only* requiring the X match the factorization at the observed entries of X

Relationship to PCA

PCA also performs a factorization of $X \approx UV$ (if you want to follow the precise notation of the PCA slides, it would actually be $X^T = UV$ where V contains the columns $Wx^{(i)}$)

But unlike collaborative filtering, in PCA, *all* the entries of X are observed

Though we won't get into the details: this difference is what lets us solve PCA exactly, while we can only solve matrix factorization for collaborative filtering locally