Online Recommender Systems

CS 780/880

Outline

- Content-based offer recommendations
- Collaborative filtering
  - Recommender systems
  - Online travel recommendations
  - Collaborative filtering
  - Matrix factorization and completion

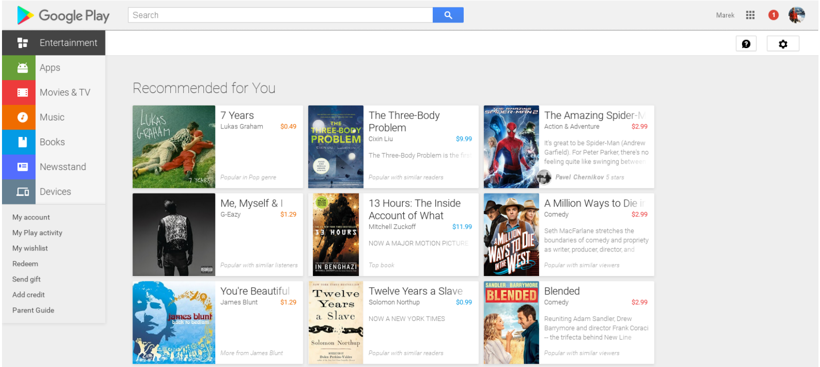
Outline

- Content-based offer recommendations
- Collaborative filtering
  - Recommender systems
  - Online travel recommendations
  - Collaborative filtering
  - Matrix factorization and completion

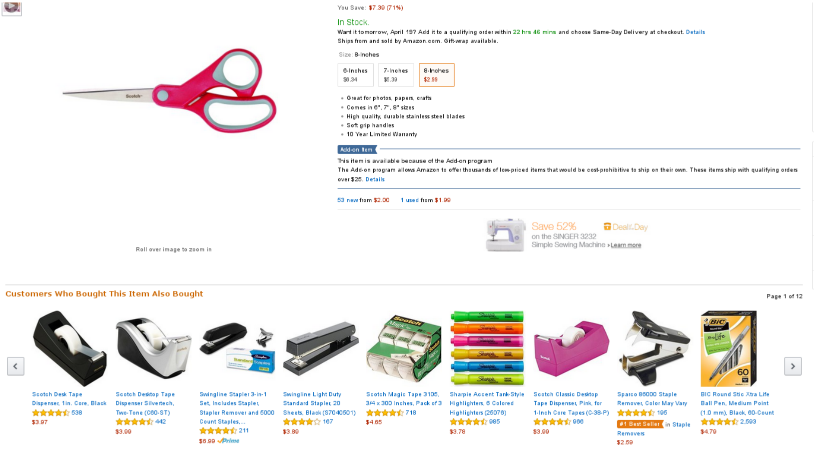
Example recommender systems

- Movie recommendations (Netflix)
- Personalized music (Pandora)
- Relevant product recommendations (Amazon)
- Online advertising (Facebook)
- Online dating (OK Cupid)
- Special offers, coupons, and discounts (Stores)

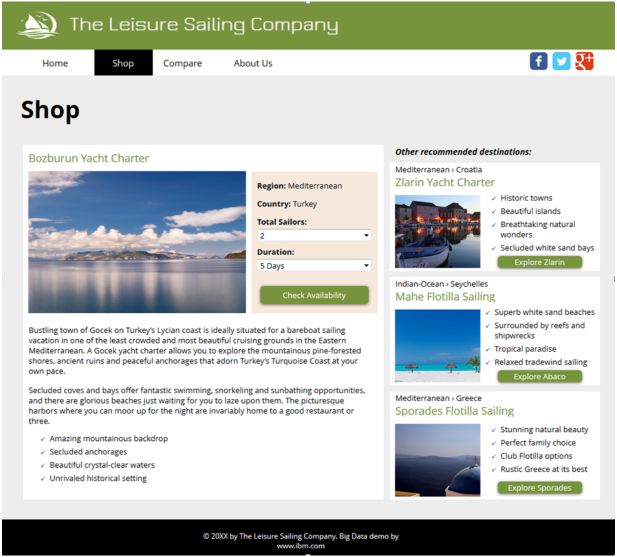
Google Play recommendations



Amazon recommendations



IBM travel recommendations



- Recommend relevant products on website of a tour operator

Travel recommendation steps

Unknown customer

Travel recommendation steps

Unknown customer

Browsing history (trips)

Travel recommendation steps

Unknown customer

Browsing history (trips)

Best guess

Travel recommendation steps

Unknown customer

Browsing history (trips)




Best guess

Recommended trips



Benefits of online recommendations

- What are the benefits?
- For users/customer:
  1. Find the right product or information
  2. See only relevant advertising
  3. Discover diverse products or information sources
- For businesses/websites:
  1. Increase user satisfaction
  2. Sell more products and right products
  3. Discriminate between customers
  4. Predict and understand customer preferences

Benefits of online recommendations

- What are the benefits?
- For users/customer:
  1. Find the right product or information
  2. See only relevant advertising
  3. Discover diverse products or information sources
- For businesses/websites:
  1. Increase user satisfaction
  2. Sell more products and right products
  3. Discriminate between customers
  4. Predict and understand customer preferences

Benefits of online recommendations

- What are the benefits?
- For users/customer:
  1. Find the right product or information
  2. See only relevant advertising
  3. Discover diverse products or information sources
- For businesses/websites:
  1. Increase user satisfaction
  2. Sell more products and right products
  3. Discriminate between customers
  4. Predict and understand customer preferences

Types of recommender systems

- By delivery:
  - **Item-to-item** recommendations: e.g. Amazon products
  - **User-to-item** recommendations: e.g. Netflix, Google Play
- By information used:
  - **Content-based**: Use item description and user profile
    - Use when rich user profile and content information is available
  - **Collaborative filtering**: Use preferences of other users
    - When rich interaction history is available
  - **Hybrid**: Combination

Types of recommender systems

- By delivery:
  - **Item-to-item** recommendations: e.g. Amazon products
  - **User-to-item** recommendations: e.g. Netflix, Google Play
- By information used:
  - **Content-based**: Use item description and user profile
    - Use when rich user profile and content information is available
  - **Collaborative filtering**: Use preferences of other users
    - When rich interaction history is available
  - **Hybrid**: Combination

Types of recommender systems

- By delivery:
  - **Item-to-item** recommendations: e.g. Amazon products
  - **User-to-item** recommendations: e.g. Netflix, Google Play
- By information used:
  - **Content-based**: Use item description and user profile
    - Use when rich user profile and content information is available
  - **Collaborative filtering**: Use preferences of other users
    - When rich interaction history is available
  - **Hybrid**: Combination

Types of recommender systems

- By delivery:
  - **Item-to-item** recommendations: e.g. Amazon products
  - **User-to-item** recommendations: e.g. Netflix, Google Play
- By information used:
  - **Content-based**: Use item description and user profile
    - Use when rich user profile and content information is available
  - **Collaborative filtering**: Use preferences of other users
    - When rich interaction history is available
  - **Hybrid**: Combination

Types of recommender systems

- By delivery:
  - **Item-to-item** recommendations: e.g. Amazon products
  - **User-to-item** recommendations: e.g. Netflix, Google Play
- By information used:
  - **Content-based**: Use item description and user profile
    - Use when rich user profile and content information is available
  - **Collaborative filtering**: Use preferences of other users
    - When rich interaction history is available
  - **Hybrid**: Combination

Content-based or collaborative filtering?

Unknown customer

Browsing history (trips)




Best guess

Recommended trips



Content-based recommender systems

- Learn user preference model based on attributes
- Use historical user preference data

User		Movie		
Gender	Age	Genre	Year	Rating
Male	16	Comedy	1998	1
Male	98	Horror	1912	5
Female	43	Action	2016	3

- Use supervised learning methods

Collaborative-filtering recommender systems

- Very flexible: no need to know content or user profiles
- Simple and powerful methods
- **Training data**: Partial user-item preferences

$$\begin{array}{cl} & \text{item} \\ \text{user} & \begin{pmatrix} 3 & ? & ? & 1 & 3 & 1 & 1 & 5 & ? & 5 \\ ? & 2 & 3 & ? & 3 & ? & 2 & ? & ? & ? \\ ? & 5 & ? & ? & 1 & ? & ? & 2 & ? & 4 \\ 4 & 3 & ? & 3 & ? & ? & ? & ? & ? & ? \\ 2 & 3 & 5 & 2 & ? & ? & ? & 5 & ? & 4 \\ ? & ? & 1 & 2 & 5 & 3 & 4 & ? & ? & ? \end{pmatrix} \end{array}$$

- **Making recommendations**: Fill in the blanks (?)

Nearest neighbors: Use similar users

- Infer preferences from similar users
- **Other users**:

$$\begin{array}{cc} & \text{item} \\ \text{user} & \begin{pmatrix} 3 & ? & ? & 1 & 3 & 1 & 1 & 5 & ? & 5 \\ 4 & 3 & ? & 3 & ? & ? & ? & ? & 5 & ? \end{pmatrix} \end{array}$$

- **Current user**:

$$\begin{array}{cc} \text{user} & \begin{pmatrix} ? & 3 & 3 & 3 & ? & ? & 1 & 5 & ? & ? \end{pmatrix} \end{array}$$

- Basic algorithm:
  1. Find similar user (e.g. 2 ratings same)
  2. Infer unknown preferences
- Why does this fail?

Nearest neighbors: Use similar users

- Infer preferences from similar users
- **Other users**:

$$\begin{array}{cc} & \text{item} \\ \text{user} & \begin{pmatrix} 3 & ? & ? & 1 & 3 & 1 & 1 & 5 & ? & 5 \\ 4 & \color{red}{3} & ? & \color{red}{3} & ? & ? & ? & ? & 5 & ? \end{pmatrix} \end{array}$$

- **Current user**:

$$\begin{array}{cc} \text{user} & \begin{pmatrix} ? & \color{red}{3} & 3 & \color{red}{3} & ? & ? & 1 & 5 & ? & ? \end{pmatrix} \end{array}$$

- Basic algorithm:
  1. Find similar user (e.g. 2 ratings same)
  2. Infer unknown preferences
- Why does this fail?

Nearest neighbors: Use similar users

- Infer preferences from similar users
- **Other users**:

$$\begin{array}{cc} & \text{item} \\ \text{user} & \begin{pmatrix} 3 & ? & ? & 1 & 3 & 1 & 1 & 5 & ? & 5 \\ 4 & 3 & ? & 3 & ? & ? & ? & ? & 5 & ? \end{pmatrix} \end{array}$$

- **Current user**:

$$\begin{array}{cc} \text{user} & \begin{pmatrix} ? & 3 & 3 & 3 & ? & ? & 1 & 5 & ? & ? \end{pmatrix} \end{array}$$

- Basic algorithm:
  1. Find similar user (e.g. 2 ratings same)
  2. Infer unknown preferences
- Why does this fail?

Nearest neighbors: Use similar users

- Infer preferences from similar users
- **Other users**:

$$\begin{array}{cc} & \text{item} \\ \text{user} & \begin{pmatrix} 3 & ? & ? & 1 & 3 & 1 & 1 & 5 & ? & 5 \\ {\color{red}4} & 3 & ? & 3 & ? & ? & ? & ? & {\color{red}5} & ? \end{pmatrix} \end{array}$$

- **Current user**:

$$\begin{array}{cc} \text{user} & \begin{pmatrix} {\color{red}4} & 3 & 3 & 3 & ? & ? & 1 & 5 & {\color{red}5} & ? \end{pmatrix} \end{array}$$

- Basic algorithm:
  1. Find similar user (e.g. 2 ratings same)
  2. Infer unknown preferences
- Why does this fail?

Nearest neighbors: Use similar users

- Infer preferences from similar users
- **Other users**:

$$\begin{array}{cc} & \text{item} \\ \text{user} & \begin{pmatrix} 3 & ? & ? & 1 & 3 & 1 & 1 & 5 & ? & 5 \\ 4 & 3 & ? & 3 & ? & ? & ? & ? & 5 & ? \end{pmatrix} \end{array}$$

- **Current user**:

$$\begin{array}{cc} \text{user} & \begin{pmatrix} 4 & 3 & 3 & 3 & ? & ? & 1 & 5 & 5 & ? \end{pmatrix} \end{array}$$

- Basic algorithm:
  1. Find similar user (e.g. 2 ratings same)
  2. Infer unknown preferences
- Why does this fail?

Nearest neighbors: Use similar users

- Infer preferences from similar users
- **Other users**:

$$\begin{array}{cc} & \text{item} \\ \text{user} & \begin{pmatrix} 3 & ? & ? & 1 & 3 & 1 & 1 & 5 & ? & 5 \\ 4 & 3 & ? & 3 & ? & ? & ? & ? & 5 & ? \end{pmatrix} \end{array}$$

- **Current user**:

$$\begin{array}{cc} \text{user} & \begin{pmatrix} 4 & 3 & 3 & 3 & ? & ? & 1 & 5 & 5 & ? \end{pmatrix} \end{array}$$

- Basic algorithm:
  1. Find similar user (e.g. 2 ratings same)
  2. Infer unknown preferences
- Why does this fail? Conflicting preferences!

Matrix factorization

- Better model for addressing incompatible preferences
- **Assumption**: Latent (unobserved) customer type, e.g.
  1. Adventure seeking
  2. Luxury oriented
- Low-rank matrix decomposition:

$$\underbrace{\begin{array}{cc} & \text{item} \\ \text{user} & \begin{pmatrix} 2 & 3 & 3 & 1 & 1 \\ 2 & 3 & 3 & 1 & 1 \\ 4 & 3 & 1 & 1 & 3 \\ 4 & 3 & 1 & 1 & 3 \\ 4 & 3 & 1 & 1 & 3 \end{pmatrix} \end{array}}_{} = \underbrace{\begin{array}{cc} & \text{type} \\ \text{user} & \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} \end{array}}_{} \underbrace{\begin{array}{cc} \text{item} & \\ \begin{pmatrix} 4 & 3 & 1 & 1 & 3 \\ 2 & 3 & 3 & 1 & 1 \end{pmatrix} & \text{type} \end{array}}_{}$$

$$D = U \quad Q^\top$$

- $D_{11} = U_1 Q_1^\top$
- Not integral in general!

Matrix factorization

- Better model for addressing incompatible preferences
- **Assumption**: Latent (unobserved) customer type, e.g.
  1. Adventure seeking
  2. Luxury oriented
- Low-rank matrix decomposition:

$$\begin{array}{cc} & \text{item} \\ \text{user} & \underbrace{\begin{pmatrix} 2 & 3 & 3 & 1 & 1 \\ 2 & 3 & 3 & 1 & 1 \\ 4 & 3 & 1 & 1 & 3 \\ 4 & 3 & 1 & 1 & 3 \\ 4 & 3 & 1 & 1 & 3 \end{pmatrix}}_{D} \end{array} = \begin{array}{cc} & \text{type} \\ \text{user} & \underbrace{\begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}}_{U} \end{array} \begin{array}{cc} \text{item} & \\ \underbrace{\begin{pmatrix} 4 & 3 & 1 & 1 & 3 \\ 2 & 3 & 3 & 1 & 1 \end{pmatrix}}_{Q^\top} & \text{type} \end{array}$$

- $D_{11} = U_1 Q_1^\top$
- Not integral in general!

Computing matrix factorization

- Given a **small** constant k
- How to compute the best factorization U, Q?

$$D \stackrel{?}{=} UQ^{\mathsf{T}} \text{ s.t. } \operatorname{rank}(U) = \operatorname{rank}(Q) = k$$

- Solve optimization problem:

$$\min_{U,Q} \|D - UQ^{\mathsf{T}}\|_F^2 \text{ s.t. } \operatorname{rank}(U) = \operatorname{rank}(Q) = k$$

- Frobenius norm: $\|A\|_F^2 = \sum_{i,j} A_{ij}^2 = \sum_i \sigma_i^2$
- Solve optimally by SVD when D is complete
- NP-hard when D is incomplete
- Simple, practical, and effective method
  1. Fix U and solve: $\min_Q \|D - UQ^{\mathsf{T}}\|_F^2$ s.t. $\operatorname{rank}(Q) = k$
  2. Fix Q and solve: $\min_U \|D - UQ^{\mathsf{T}}\|_F^2$ s.t. $\operatorname{rank}(U) = k$

Computing matrix factorization

- Given a **small** constant k
- How to compute the best factorization U, Q?

$$D \stackrel{?}{=} UQ^{\mathsf{T}} \text{ s.t. } \operatorname{rank}(U) = \operatorname{rank}(Q) = k$$

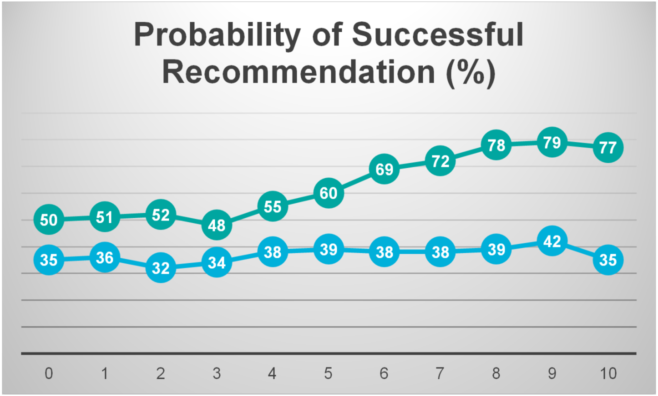
- Solve optimization problem:

$$\min_{U,Q} \|D - UQ^{\mathsf{T}}\|_F^2 \text{ s.t. } \operatorname{rank}(U) = \operatorname{rank}(Q) = k$$

- Frobenius norm: $\|A\|_F^2 = \sum_{i,j} A_{ij}^2 = \sum_i \sigma_i^2$
- Solve optimally by SVD when D is complete
- NP-hard when D is incomplete
- Simple, practical, and effective method
  1. Fix U and solve: $\min_Q \|D - UQ^{\mathsf{T}}\|_F^2$ s.t. $\operatorname{rank}(Q) = k$
  2. Fix Q and solve: $\min_U \|D - UQ^{\mathsf{T}}\|_F^2$ s.t. $\operatorname{rank}(U) = k$

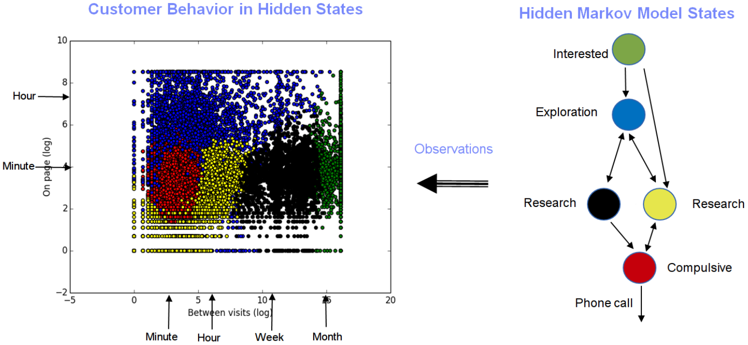
Matrix factorization: Results

- One year worth of website click-stream data
- Dependence on number of previously visited sites
- Matrix factorization (green) vs baseline (blue)



Beyond simple models: Dynamic user behavior

- Hidden Markov Model



- Strategic recommendations: optimal in long run (conversion/satisfaction)

Recommendations on social media (Twitter)

- Challenges
  1. Large volume: (6000 t/s)
  2. Short (cryptic) text: “Cruise with me on this wild ride”
- Approach:
  1. Use language models to identify travel intent
  2. Matrix factorization to match catalog descriptions with tweets
- Positive engagement: 15%