Recommender Systems

Introduction

- Web enables users to provide feedback about their likes or dislikes; Netflix, ratings are submitted by users

- Other forms of implicit feedback collected in the Web-centric paradigm; Amazon.com

- The basic idea of recommender systems is to utilize various sources of data to **infer customer interests**

Introduction

- **User:** the entity to which the recommendation is provided
- **Item:** the product that is recommended

- Most recommender systems: based on the previous interaction between users and items

- **Knowledge-based recommender systems:** in which the recommendations are suggested on the basis of user-specified requirements rather than the past history of the user

Basic Principle

- Significant **dependencies** exist between user and item-centric activity:
  - Various categories of items may show significant correlations
  - Dependencies can be learned in a data-driven manner from the ratings matrix

Problems of Recommender Systems

1. Prediction:
   - User preferences for items as training data is available
   - For m users and n items, this corresponds to an incomplete $m \times n$ matrix, where the specified (or observed) values are used for training. The missing (or unobserved) values are predicted using this training model

Problems of Recommender Systems

2. Ranking:
   - recommend the top-k items for a particular user (more common)
   - determine the top-k users to target for a particular item

Popular Examples

- **GroupLens Recommender System**
  - recommendation of Usenet news

Popular Examples

- **GroupLens Recommender System**
  - recommendation of Usenet news
- **Amazon.com Recommender System**
- **Netflix Movie Recommender System**
- **Google News Personalization System**
- **Facebook Friend Recommendations**
  - link prediction

Popular Examples

- Movie Recommendation
    - Netflix uses **Spark** to make large-scale movie recommendation to its users.
    - It does this by studying what movie users watch and do not watch in the Netflix.

Basic Models

- Two types of data:
    1. the user-item interactions, such as ratings or buying behavior
    2. the attribute information about the users and items, such as textual profiles or relevant keywords

Basic Models

- Two types of data:
    1. the user-item interactions, such as ratings or buying behavior (suitable for **collaborative filtering** methods)
    2. the attribute information about the users and items, such as textual profiles or relevant keywords (**content-based** recommender methods)

Basic Models

- Two types of data:
  1. the user-item interactions, such as ratings or buying behavior (suitable for **collaborative filtering** methods)
  2. the attribute information about the users and items, such as textual profiles or relevant keywords (**content-based** recommender methods)
- **Knowledge-based** recommender systems:
  - external knowledge bases and constraints are used to create the recommendation, not historical rating or buying data

Basic Models

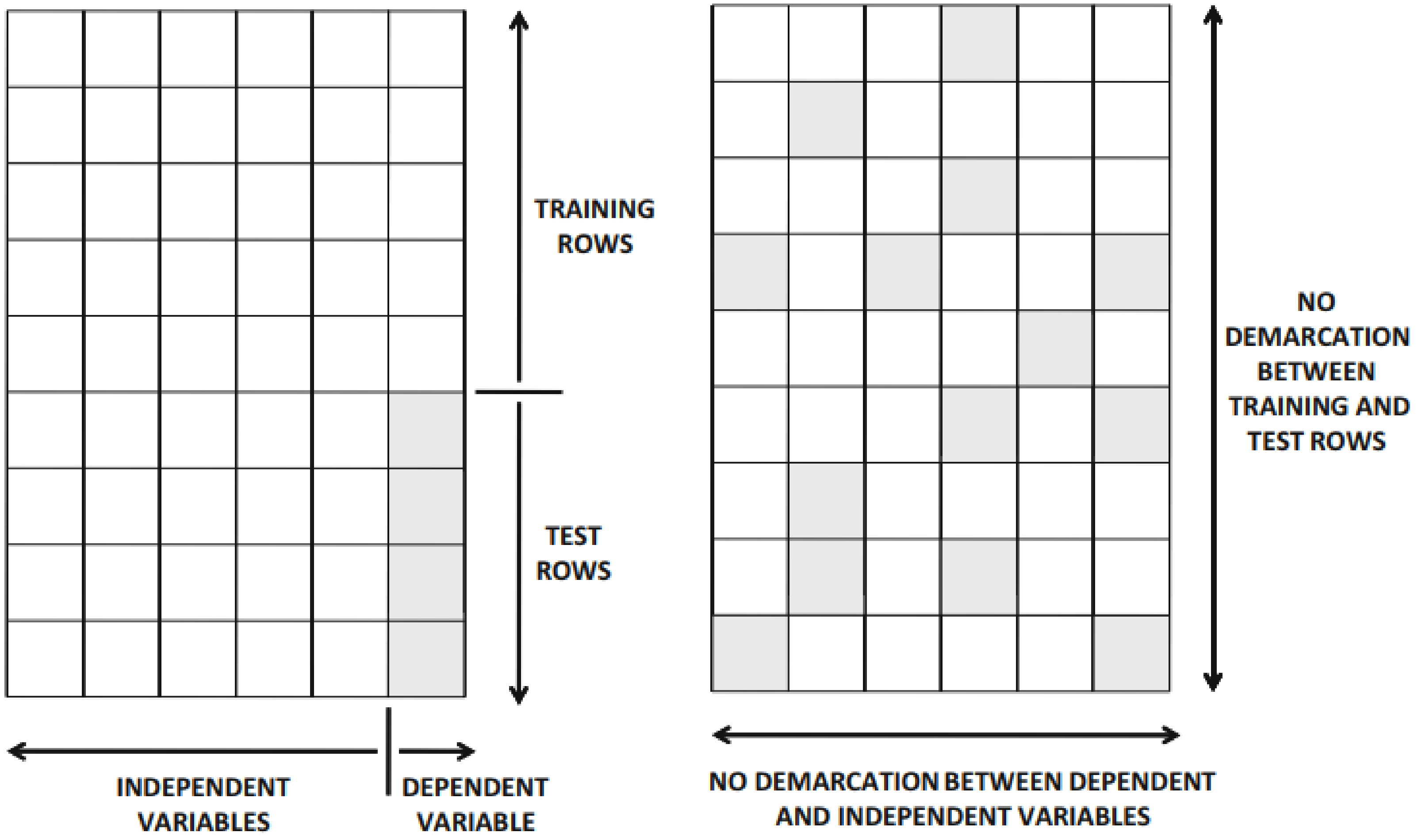
- Two types of data:
  1. the user-item interactions, such as ratings or buying behavior (suitable for **collaborative filtering** methods)
  2. the attribute information about the users and items, such as textual profiles or relevant keywords (**content-based** recommender methods)
- **Knowledge-based** recommender systems:
  – external knowledge bases and constraints are used to create the recommendation, not historical rating or buying data
- **Hybrid** systems
  – combine various types of recommender systems

Collaborative Filtering Models

1. Memory-based methods (neighborhood-based collaborative filtering algorithms)
   - +simple to implement
   - +easy to explain
   - -do **not** work very well with **sparse** ratings matrices

2. Model-based methods
   - Machine learning and data mining methods are used:
     - decision trees, rule-based models, Bayesian methods and latent factor models

Collaborative Filtering Models



(a) Classification (b) Collaborative filtering

Neighborhood-based methods

1. **User**-based collaborative filtering:
   - determine users, who are similar to the target user A, and recommend ratings for the **unobserved** ratings of A by computing weighted averages of the ratings of this “peer group”
   - the ***k*** most similar users to A can be used to make rating predictions for A
   - Similarity functions are computed between the **rows** of the ratings matrix to discover similar users

Neighborhood-based methods

2. **Item**-based collaborative filtering:
   - rating predictions for target **item** B by **user** A
   - determine a set *S* of **items** that are most similar to target **item** B
   - ratings in S, which are specified by **users** A, are used
   - Similarity functions are computed between the **columns** of the ratings matrix to discover similar items

Neighborhood-based methods

Ratings are predicted in

- **user-based** by the ratings of **neighboring users**
- **item-based** by the user’s **own** ratings on neighboring (i.e., closely related) items

Neighborhood-based methods

Assumption

- an incomplete m $\times$ n matrix R = $[r_{uj}]$ is the user-item ratings matrix containing **m** users and **n** items
- only a small subset of the ratings matrix is specified or observed (training data)
- the unspecified entries (test data)

Neighborhood-based methods

Assumption

- an incomplete m × n matrix R = $[r_{uj}]$ is the user-item ratings matrix containing **m** users and **n** items
- only a small subset of the ratings matrix is specified or observed (training data)
- the unspecified entries (test data)

Formulation:

1. Predicting the rating value of a user-item combination
2. Determining the top-k items or top-k users

Neighborhood-based methods

Assumption

- an incomplete m × n matrix R = $[r_{uj}]$ is the user-item ratings matrix containing **m** users and **n** items
- only a small subset of the ratings matrix is specified or observed (training data)
- the unspecified entries (test data)

Formulation:

1. Predicting the rating value of a user-item combination
   - missing rating r_{uj} of the user *u* for item *j* is predicted
2. Determining the top-k items or top-k users

Neighborhood-based methods

Assumption

- an incomplete m × n matrix R = $[r_{uj}]$ is the user-item ratings matrix containing **m** users and **n** items
- only a small subset of the ratings matrix is specified or observed (training data)
- the unspecified entries (test data)

Formulation:

1. Predicting the rating value of a user-item combination
   - missing rating r_{uj} of the user *u* for item *j* is predicted
2. Determining the top-k items or top-k users
   - learn the top-k most relevant items / users for a particular user / item

Utility (Rating) Matrix

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A							
B							
C							
D							

Populating the Utility Matrix

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Measuring Similarity

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Jaccard Distance

- d(x, y) = 1 - SIM(x, y) $= 1 - \frac{x \cap y}{x \cup y}$
- d(x, y) = 0 if x = y, because x ∪ x = x ∩ x = x. if x ≠ y, then the size of x ∩ y is strictly less than the size of x ∪ y, so d(x, y) is strictly positive

Cosine Distance

- Given two vectors x and y, the cosine of the angle between them is the **dot product** x.y divided by the L2-norms of x and y (i.e., their Euclidean distances from the origin).

Cosine Distance

- $x = [1, 2, -1]$ and $y= [2, 1, 1]$.

Cosine Distance

- x = [1, 2, -1] and y= [2, 1, 1].
- The **dot** product x.y is 1 × 2 + 2 × 1 + (-1) × 1 = 3.
- x has L2-norm $\sqrt{1^2 + \ 2^2 + (-1)^2} = \sqrt{6}$.
- Thus, the cosine of the angle between x and y is $\frac{3}{\sqrt{6} \times \sqrt{6}} = \frac{1}{2}$

- The angle whose cosine is ½ is 60 degrees, so that is the cosine distance between x and y.

Jaccard Distance

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Jaccard Distance

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

A and B have an intersection of size 1 and a union of size 5. Thus, their Jaccard similarity is 1/5, and their Jaccard distance is 4/5; i.e., they are very far apart.

A and C have a Jaccard similarity of 2/4, so their Jaccard distance is the same, 1/2. A appears closer to C than to B.

Cosine Distance

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Cosine Distance

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

The cosine of the angle between A and B is

$$\frac{4\times5}{\sqrt{4^2+5^2+1^2}\sqrt{5^2+5^2+4^2}}=0.380$$

Cosine Distance

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

The cosine of the angle between A and B is

$$\frac{4\times5}{\sqrt{4^2+5^2+1^2}\sqrt{5^2+5^2+4^2}}=0.380$$

The cosine of the angle between A and C is

$$\frac{5\times2+4\times1}{\sqrt{4^2+5^2+1^2}\sqrt{2^2+4^2+5^2}}=0.322$$

Rounding the Data

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Rounding the Data

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	1			1			
B	1	1	1				
C					1	1	
D		1					1

Rounding the Data

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	1			1			
B	1	1	1				
C					1	1	
D		1					1

Jaccard distance:
d(A,B)=
d(A,C)=

Rounding the Data

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	1			1			
B	1	1	1				
C					1	1	
D		1					1

Jaccard distance:
d(A,B)= 3/4
d(A,C)= 1

Rounding the Data

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	1			1			
B	1	1	1				
C					1	1	
D		1					1

Jaccard distance:
d(A,B)= 3/4
d(A,C)= 1

C appears further from A than B

Rounding the Data

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	1			1			
B	1	1	1				
C					1	1	
D		1					1

Jaccard distance:
d(A,B)= 3/4
d(A,C)= 1

Cosine distance:
d(A,B)= ?
d(A,C)= ?

C appears further from A than B

Ratings Types

1. Continuous (rarely used)
2. Interval-based
   - Examples: numerical integer values from 1 to 5, from -2 to 2, or from 1 to 7
3. Ordinal (ordered categorical values)
   - “Strongly Disagree”, “Disagree”, “Neutral”, “Agree”, “Strongly Agree”
4. Binary
   - only two options
5. Unary
   - specify a positive preference for an item, but no mechanism to specify a negative preference; a “like” button on Facebook

Ratings Types

	GLADITOR	GODFTHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			5		2
U_2		5			4	
U_3	5	3		1		
U_4			3			4
U_5				3	5	
U_6	5		4			

(a) Ordered ratings

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			1		1
U_2		1			1	
U_3	1	1		1		
U_4			1			1
U_5				1	1	
U_6	1		1			

(b) Unary ratings

Ratings Types

	GLADITOR	GODFTHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			5		2
U_2		5			4	
U_3	5	3		1		
U_4			3			4
U_5				3	5	
U_6	5		4			

(a) Ordered ratings

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			1		1
U_2		1			1	
U_3	1	1		1		
U_4			1			1
U_5				1	1	
U_6	1		1			

(b) Unary ratings

Reference

Aggarwal CC. Recommender systems. Cham: Springer International Publishing; 2016.