Tutorial : Recommendation & Data Preparation

Q1

User based CF of 3 most similar users.

Cosine similarity of [u1, u3..u12] with u2:

[0.372, 0.29, 0.217, 0.527, 0.0, 0.325, 0.198, 0.475, 0.667, 0.487, 0.0]

Rankings of users based on similarity:

[10, 5, 11, 9, 1, 7, 3, 4, 8, 12, 6]

Top 3 users who rated movie 1:

u11, u9, u1 (because u10 and u5 didn’t rate movie 1)

Similarity-weighted recommendation:

$$\frac{0.487 * 4 + 0.475 * 5 + 0.372 * 1}{0.487 + 0.475 + 0.372} = 3.519$$

Unweighted recommendation:

$$\frac{4 + 5 + 1}{3} = 3.33$$

Item based CF of 3 most similar items.

Step 1:

Cosine similarity of [i2…i12] with i1:

[0.528, 0.526, 0.285, 0.302, 0.239, 0.470, 0.913, 0.681, 0.533, 0.257, 0.465]

Rankings of items based on similarity:

[8, 9, 10, 2, 3, 7, 12, 5, 4, 11, 6]

Top 3 items that interact with u2:

i10, i3, i4

Similarity weighted recommendation:

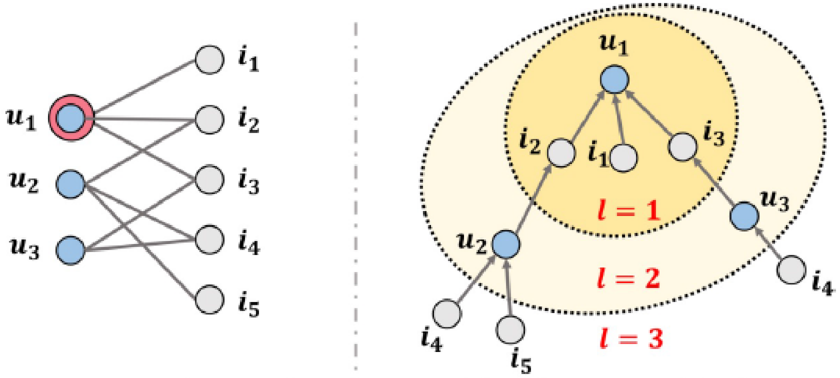
$$\frac{0.533 * 2 + 0.526 * 4 + 0.285 * 2}{0.533 + 0.526 + 0.285} = 2.78$$

Unweighted recommendation:

$$\frac{2 + 4 + 2}{3} = 2.67$$

Q2

- High-order connectivity
  - Recommender systems rely on capturing similarity
    - User-user (User-CF), item-item (Item-CF), user-item (Model-CF)
  - GNN extends similarity to high-orders
    - Connectivity among high-order neighbors
  - Besides, data sparsity issue is well addressed



User-Item Interaction Graph

High-order Connectivity for u_1

Q3

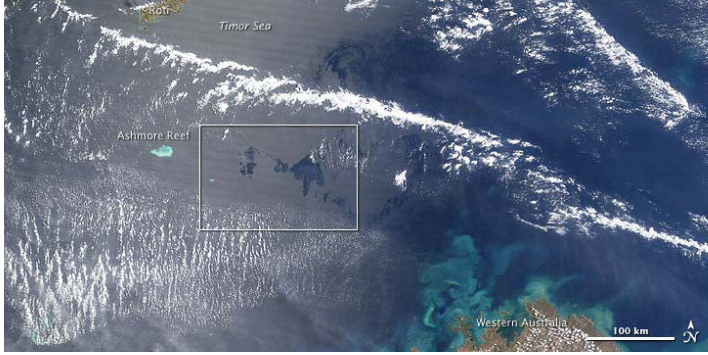
In order to perform knowledge discovery what important steps do you need to perform in the following cases:

a) Trying to detect oil slicks from satellite images to give early warning of ecological disasters and deter illegal dumping

b) A bank is trying to reduce customer attrition

c) A supermarket is planning store layouts

(1) Answers



Trying to detect oil slicks from satellite images to give early warning of ecological disasters and deter illegal dumping

This is essentially a classification task.

- Step 1: data cleaning, preprocessing
- Step 2: manually assign labels to images:
  1 for oil slick and **0** for “ no oil slick”
- Step 3: Build a classifier, Train it using labeled data, test it over unlabeled data

(2) Answers

A bank is trying to reduce customer attrition

- Step 1: Clean and preprocess data and select customers profile data, their transaction history
- Step 2: Perform classification on customers into “loyal” and “nonloyal”
- Step 3: Build a classifier, Train it using labeled data, test it over unlabeled data

(3) Answers

A supermarket is planning store layouts

It can be solved by associate rule mining.

- Step 1: Clean, preprocess and select transactional data
- Step 2: Perform Association Rule Mining (ARM)
- Step 3: Re-arrange the store based on ARM results in order to facilitate the customers

Example:

{diaper, milk} --> {beer}

Q4

- Suppose a group of 12 sales price records has been sorted as follows:

  5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215

  Partition them into 3 bins by each of the following methods:

  - equal-frequency (equidepth) partitioning
  - equal-width partitioning

  Use an appropriate method to smooth.

  (let’s use the mean for this tutorial)

Answer

a) Equal frequency: #bins = 3, Frequency of each bin = 4

Bin#1:	**Bin#2:**	**Bin#3**
[5, 10, 11, 13]	[15, 35, 50, 55]	[72, 92, 204, 215]
Mean=10	Mean=39	Mean=146
[10, 10, 10, 10]	[39, 39, 39, 39]	[146, 146, 146, 146]

b) Equal Width = (215-5)/3 = 70

Bin#1: 5-75	**Bin#2: 75-145**	**Bin#3: 145-215**
[5, 10, 11, 13, 15, 35, 50, 55, 72]	[92]	[204, 215]
Mean = 30	Mean = 92	Mean = 210
[30, 30, 30, 30, 30, 30, 30, 30, 30]	[92]	[210, 210]

Q5

- Given two sets of restaurant objects: one set is collected from Google Maps, and the other set is collected from Yelp. Each restaurant object is associated with a set of attributes, such as name, address, coordinates, phone numbers, etc. Please design an algorithm for entity matching to find all the pairs of the same restaurant objects from the two sets. Analyze the algorithm complexity.

Basic solution

- For each pair of entities (g, y), one from G and the other from Y
    - Compute the similarity of g and y
        - How?
    - Simple threshold method: If the distance below some number, same
    - Define some rule
- Complexity?
    - There are $O(N^2)$ possible matches
    - Each match take m
    - $O(m N^2)$

More advanced solution

- Better efficiency?
- Reduce the search space
  - For an entity g in G, we only find those entities within a radius of 1km in Y for matching.

- Better effectiveness?
- Design better similarity function.