Lecture 07-08

- Content-based Methods
    - Multi-round Recommendations

IT492: Recommendation Systems (AY 2023/24) — **Dr. Arpit Rana**

Content-based Methods Revisited

Content-based methods try to predict the ***utility*** of items for an ***active user*** based on ***item descriptions*** and her ***past preferences***.

In content-based systems, there are ***choices*** on the following

- **Item representation**: how items are represented,
- **User profile**: how user preferences are modeled, and
- **Filtering technique**: how items are matched with the user preferences.

Filtering Technique

A *filtering technique* suggests relevant items from a set of candidate items.

These techniques are also split into the following categories -

- *Memory-based techniques*: employ similarity measures to match the representations of candidate items against the profile
- *Model-based techniques*: learn from the profile a model that can predict item relevance

Vector Space Model

VSM is a ***spatial representation*** of text documents wherein -

- each document is represented by a vector in an ***n-dimensional space***
- each dimension corresponds to a ***term*** from the overall ***vocabulary*** of a given document collection

Vector Space Model

- Imagine each item (e.g. movie) is represented by a binary-valued (column) vector of dimension ***d***,
  - e.g. ***d = 3***, where each element of the vector corresponds to a feature (e.g. movie genre).
- We can gather these vectors into a matrix, which we will refer to as ***Q***
  - So, ***Q*** is a ***d x | I |*** matrix.
  - If we want to refer to the column in ***Q*** that corresponds to item ***i***, we will write Q_i

	i_1	i_2	i_3	i_4	i_5	i_6
comedy	1	0	0	1	1	0
thriller	0	0	0	0	1	1
romance	1	0	1	0	1	0

Vector Space Model

- Imagine each user is represented by a binary-valued row vector of her tastes.
  - These vectors also have dimension ***d***, and the elements correspond to the ones used for items.
- We can gather these vectors into a matrix, which we will refer to as P
  - So, P is a $|U| \times d$ matrix.
  - If we want to refer to the row in P that corresponds to user u, we will write P_u

	comedy	thriller	romance
u_1	0	1	0
u_2	1	1	1
u_3	0	0	0
u_4	1	0	1

Vector Space Model

- The ***score*** that capture the ***relevance*** to user ***u*** of item ***i*** is simply the similarity of vectors Q_i and P_u
- We can use ***cosine similarity*** for this (ignoring normalization). This is simply the product of the two vectors.

$$sim(u,i) = P_u . Q_i$$

	comedy	thriller	romance
u_1	0	1	0
u_2	1	1	1
u_3	0	0	0
u_4	1	0	1

	i_1	i_2	i_3	i_4	i_5	i_6
comedy	1	0	0	1	1	0
thriller	0	0	0	0	1	1
romance	1	0	1	0	1	0

Recommender: Pre-Processing

Data Preparation:

- Model a recommendation problem as a ***ranking problem***
- **Convert user ratings** (non-binary) to "liked" or "disliked" classes.
  - On a likert scale (e.g., 1 – 5), 1–3 as "dislike" and 4 – 5 as "liked".
- **Split** your data to *train*, *validation*, and *test*.
  - Remember, this split will be done user-wise with *stratification.*

K Nearest Neighbour

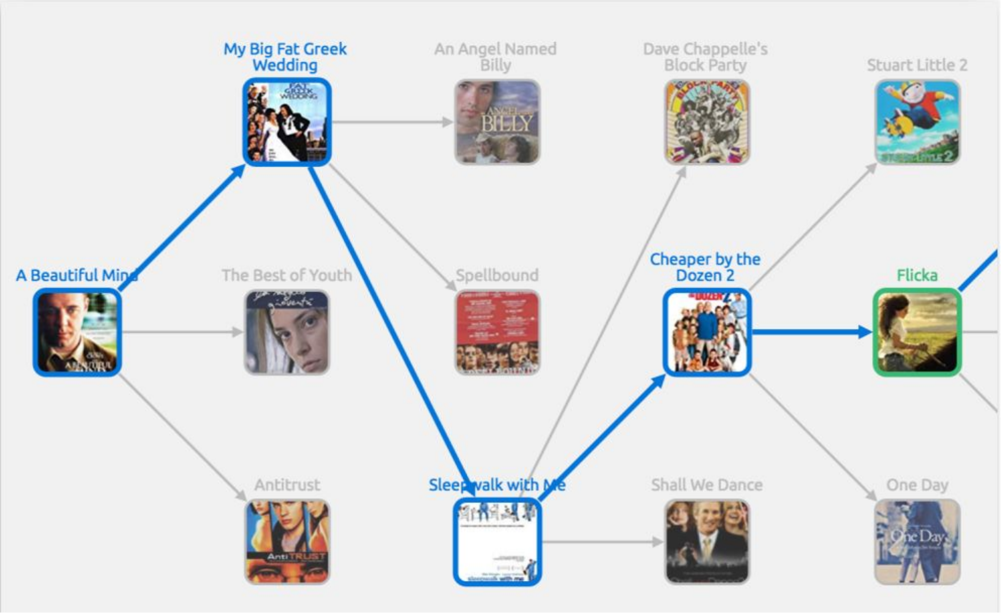
Algorithm:

- Calculate ***item-item similarity*** using their *content-based representation*
  - *Using **Cosine, Jaccard, or (Sørensen's) Dice Coefficient**.*
- For each candidate item:
  - Find top-*k* neighbours in user *u*'s profile
  - Calculate average of similarity scores between the candidate item and the corresponding neighbours in user profile
  - Add the candidate item with the score in the recommendation list
- Sort the recommendation list based on the item score and select top-*N* to recommend to the user *u*.

- Multi-round Recommendations

Conversational Recommendation

Conversational Recommender System: *Recommend* → *Review* → *Refine*



Conversational Recommendation

Conversational Recommender System: *Recommend* → *Review* → *Refine*

- When users are not satisfied with initial top-n recommendations
- When users have ephemeral goals different from their usual tastes
- When user requirements are uncertain or are not fully observable (e.g. context, the user's mood, her companions, etc.) [Pu & Chen 2008]
- Achieves higher level of trust and transparency, and greater acceptance by enabling users to steer the recommendation [He et al. 2016]

User Feedback in GUI-based Conversational Recommendations

Conversational Strategies

Navigation-by-Asking *(direct)*

- **Value elicitation**

  User provides a value for a specific attribute.

  e.g. I want a 2.4GHz Intel core I5 processor laptop.

- **Critiquing**

  User proposes *tweaks* to attribute values.

  e.g. Show me more like option A, but *cheaper*.

Navigation-by-Proposing *(less direct)*

- **Rating-based**

  User *rates* the recommended items without consuming them.

  e.g. Show me more like 20% of option A, 40% of option B, and 70% of option C.

- **Preference-based**

  User simply *selects* one of the recommended items.

  e.g. Show me more like option A.

User Feedback Forms: Facts and Challenges

Value elicitation

- Optimal set of attributes and the logical order
- High level of domain knowledge
- Users' willingness to answer direct and specialized questions

Critiquing

- Handling preference conflicts
- Helping users to perform trade-off analysis

Rating-based

- Level of efforts increases as the number of recommended items increases

Preference-based

- Usually leads to longer dialogs

User Feedback: Influencing Factors

	Value elicitation	*Critiquing*	*Rating -based*	*Preference -based*
Cost (effort required by the user)	XXX	XX	XX	X
Ambiguity (ability to guide the recommender)	X	XX	XX	XXX
Expertise (domain knowledge required)	XXX	XX	XX	X
Interface (type of interface required)	XXX	XX	X	X

Goals of Conversational Recommender Systems

Effectiveness (*maximize*)

Effectiveness is the degree to which the system helps the user to accomplish her task.

e.g. finding a relevant recommendation or some broader measure of user satisfaction

Efficiency cost (*minimize*)

Efficiency cost is a measure of the effort involved in completing the task.

e.g. In terms of total time elapsed, total number of user actions with the system's user interface, number of interaction cycles, or cognitive load

Disadvantages of Content-based Systems

The main advantage of content-based methods is that they are *easy to explain at feature-level*. Their most significant challenges include the following:

- *Degree of content analysis:* Their ability to discriminate between items depends on the granularity of the item representations. If two different items are represented by the same set of features, they are indistinguishable and equally likely to be recommended.

- *Over-specialization:* These methods tend to recommend items that are similar to items the user has liked in the past. Thus, they often provide the least serendipitous recommendations.

- *Cold-start user:* A new user, with an immature profile, is less likely to get accurate recommendations

Next Lecture

- Evaluation of Recommendation Systems