IT492: Recommendation Systems

Lecture - 09

Content-based Methods

Arpit Rana

21st Feb 2022

Limitations of Collaborative Methods

Collaborative Methods have the following disadvantages:

- ***Sparsity***: The number of observed ratings is usually very small compared to the number of user-item pairs. Therefore, it is challenging to find similar users, similar items, or other patterns that are non-spurious.

- ***Cold-start items and users***: These systems would not be able to recommend the new item (not substantially rated) or recommend to the new user (who has not rated substantial number of items).

- **Popularity bias**: These methods recommend items based on ratings and hence they tend not to recommend products with limited historical data.

- **Shilling attacks:** In collaborative settings, malicious users and/or competing vendors may insert fake profiles in an effort to affect the rating predictions for their own advantages.

Definition

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.

Item Representation

- **Structured:** a finite and typically small set of attributes
  - e.g. For products: size, weight, manufacturer, etc.
    for movies: director, length, language, guidance certificate, etc.
    for songs: artist, producer, record label, etc.
- **Unstructured:** no explicit structure, ***often processed to obtain meaningful information***
  - e.g. Keywords extracted from a movie description or user reviews; user assigned tags to an item;
- **Semi-structured:** mixture of *structured* and *unstructured* information
  - e.g. movie genres (comedy, thriller, romance, …) with movie keywords

Item Representation

Source of Information for representing items -

- **Attribute-value pairs:** in terms of values for predefined attributes.
- **Item content:** content (textual descriptions) can be mined for *keywords*
- **User reviews:** user experiences or opinions about a product or service in the form of reviews, can be mined for *aspects, context,* etc.
- **User assigned tags:** can be characterized as ***objective*** (where they convey factual information about an item) or ***subjective*** (where they express the user's opinion about an item).
- **Linked data:** is inter-connected data that is published in a way that complies with the Linked Data principles. e.g., DBpedia

User Profile

A user's profile represents her ***preferences***.

User preferences may be -

- ***Persistent***, indicating a *user's long-term tastes and interests*, or
- ***Ephemeral***, reflecting her *transient (or short-term) requirements*

A user profile is most typically a representation of her ***persistent*** preferences.

User Profile

A user profile simply consists of a history of the user's interactions with the recommender system, for example -

- ***items*** she has viewed or purchased, or ratings that she has given to the items;
- ***features*** that describe the user's tastes and interests (obtained from a sign-up form or aggregated from the items the user interacts with)

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

Keyword-based Vector Space Model

VSM is a spatial representation of text documents wherein -

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

Keyword-based 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 $\boldsymbol{Q_i}$

	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

Keyword-based 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*** | ***x 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

Keyword-based 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

Linguistic Preprocessing (briefly)

In case of unstructured (text) representation, pre-processing is required.

- **Tokenization**: Turning documents into a list of tokens, e.g.,

  Marie was born in Paris.
  "Marie", "was", "born", "in", "Paris", "."

- **Drop stop words:** Removing extremely common words with very low value, e.g.,

  ".", “the”, “a”, “to”, “of”

- **Normalization:** Map text and query term to same form, e.g.,

  “U.S.A” to “USA”; “anti-social” to “antisocial”

- **Stemming/Lemmatization:** Turning tokens to its base form, e.g.,

  “was”, “am”, “are” to “be”; “car”, “cars”, “car’s” to “car”

Item Representation

Representing items using unstructured descriptions (e.g. keywords, tags) raises two issues:

- weighting the terms, and
- measuring the similarity between two items.

Term weighting

The most commonly used term weighting scheme, TF-IDF (Term Frequency–Inverse Document Frequency) which relies on the following assumptions -

- rare terms are not less relevant than frequent terms (IDF assumption);
- multiple occurrences of a term in a document are not less relevant than single occurrences (TF assumption);
- long documents are not preferred to short documents (normalization assumption).

Term weighting

TF-IDF (Term Frequency– Inverse Document Frequency)

$$\mathrm{TF\text{-}IDF}(t_k, d_j) = \underbrace{\mathrm{TF}(t_k, d_j)}_{\mathrm{TF}} \cdot \underbrace{log \frac{N}{n_k}}_{\mathrm{IDF}}$$

$$\mathrm{TF}(t_k, d_j) = \frac{f_{k,j}}{max_z f_{z,j}}$$

Term weighting

Normalized TF-IDF

$$w_{k,j} = \frac{\text{TF-IDF}(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} \text{TF-IDF}(t_s, d_j)^2}}$$

Measuring Item Similarity

Similarity between two items represented by the constituent terms -

$$sim(d_i, d_j) = \frac{\sum_k w_{ki} \cdot w_{kj}}{\sqrt{\sum_k w_{ki}^2} \cdot \sqrt{\sum_k w_{kj}^2}}$$

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.

i.e. no surprising recommendations

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

Next lecture -

Content-based Recommendations contd…