IT492: Recommendation Systems

Lecture - 11

Content-based Methods

Arpit Rana

28th Feb 2022

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.

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 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

Supervised Learning

Given:

- A document $d \in X$
- A fixed set of classes $C = \{c_1, c_2, c_3, \ldots c_n\}$
- A training set of documents D each with a label in C: $\langle d, c \rangle \in X \times C$
  - $\langle d, c \rangle = \langle$*Beijing joins the World Trade Organization, China*$\rangle$

Determine:

- A ***learning method*** or ***learning algorithm*** which will enable us to learn a classifier γ

$\boldsymbol{\gamma}$**:** $\boldsymbol{X \rightarrow C}$

$\boldsymbol{\Gamma(D) = \gamma}$

$\boldsymbol{\Gamma}$ is a learning method that takes training set ***D*** as input and returns a classifier $\boldsymbol{\gamma}$**,** e.g. Naive Bayes

This is called a ***supervised learning*** as a supervisor (who defines the classes and labels training documents) serves as a teacher directing the learning process.

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***.

Naive Bayes Algorithm

The probability of a document ***d*** being in class ***c*** is computed as

$$P(c \mid d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k \mid c)$$

Beijing and Taipei join the WTO

⟨Beijing, Taipei, join, WTO⟩
so, $n_d = 4$

- n_d is the length of the document (number of tokens)
- $P(t_k \mid c)$ is the
  - conditional probability of term t_k occurring in a document of class c
  - measure of how much evidence t_k contributes that c is the correct class
- $P(c)$ is the prior probability of c

Naive Bayes Algorithm

Our goal is to find the best class for document ***d***

- The best class is the most likely or ***maximum a posteriori (MAP)*** class ***c_{map}***

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg\max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k|c).$$

- $\hat{P}$ denotes that probability is estimated from the training set, not a true value.

Naive Bayes Algorithm

Multiplying lots of small probabilities can result in *floating point underflow*

- Since ***log(xy) = log(x) + log(y)***, we can sum ***log*** probabilities instead of multiplying probabilities.
- Since ***log*** is a ***monotonic function***, the class with the highest score does not change.

So what we usually compute in practice is:

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} \left[\log \hat{P}(c) + \sum_{1 \le k \le n_d} \log \hat{P}(t_k|c)\right].$$

Naive Bayes Algorithm

Simple interpretation:

- Each conditional parameter $\log \hat{P}(t_k|c)$ is a weight that indicates how good an indicator t_k is for class c.
- The prior $\log \hat{P}(c)$ is a weight that indicates the relative frequency of class c.

We select the class with the most evidence (weight).

Add-one Smoothing:

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V}(T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B'}$$

Linear Regression

Do It Yourself.

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 -

Hybrid Recommendations