Content-based Recommender Systems

Recuperação de Informação
Doutoramento em Engenharia Informática e Computadores

Paula Cristina Vaz – 60620/D

Instituto Superior Técnico
Universidade Técnica de Lisboa

Bibliography

Pasquale Lops, Marco de Gemmis, Giovanni Semeraro: *Content-based Recommender Systems: State of the Art and Trends*. Recommender Systems Handbook: 73-105 (2011) Springer

Outline

Introduction

The large amounts of information available in the Internet.

Introduction

The large amounts of information available in the Internet.

The information in the Internet is dynamic and heterogeneous.

Introduction

The large amounts of information available in the Internet.

The information in the Internet is dynamic and heterogeneous.

Personalizing the access to the available information is important.

Introduction

The large amounts of information available in the Internet.

The information in the Internet is dynamic and heterogeneous.

Personalizing the access to the available information is important.

Recommendation systems

Research in recommendation emerged from the information retrieval research in the mid-90s.

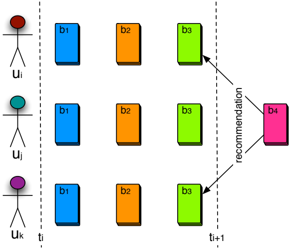
Recommendation paradigms

- **Collaborative filters**
  Collaborative filters identify users with similar preferences and use this information to generate recommendations.

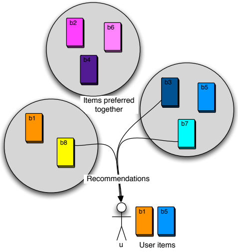
- **Content-based filters**
  Content-based filters try to recommend items similar to those a given user has liked in the past.

Collaborative filters

User-based collaborative filtering

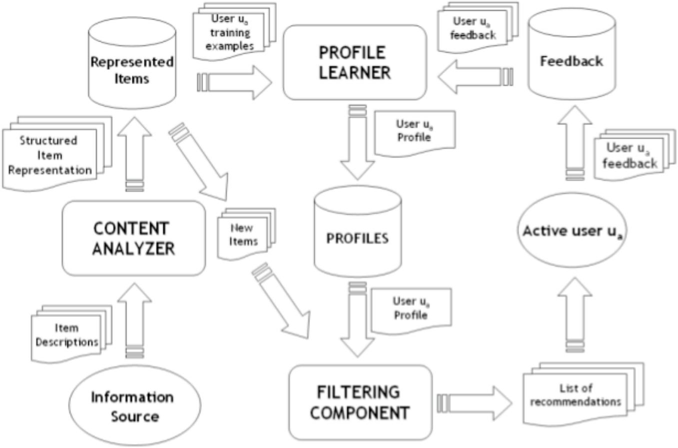


Item-based collaborative filtering

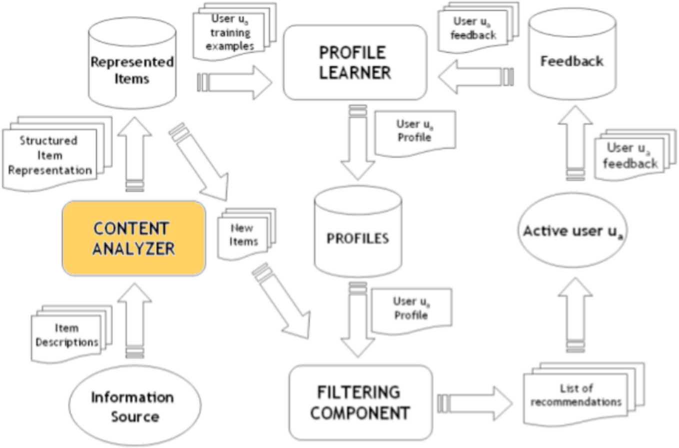


1. Introduction
2. Content-based Recommendation System
   - High Level Architecture
   - Content analyzer
   - Profile learner
     - Probabilistic models and Naïve Bayes
     - Relevance feedback and Rocchio's algorithm
   - Filtering component
   - User Feedback
3. Advantages and drawbacks
4. Over-specialization

High Level Architecture



Content-analyzer



Content-analyzer

The content analyzer generates structured representations of the original items, e.g., documents, web pages news articles, product descriptions, etc.

Item representation have two types:

- the keyword vector space model
- representations that include semantic knowledge

Item representation: Keyword-based vector space model

Documents are represented by a n-dimensional vector, where each dimension corresponds to a term.

- Let
  - $D = \{d_1, d_2, ..., d_N\}$ the document set
  - $T = \{t_1, t_2, ..., t_n\}$ the term set
- document d_j is represented by $d_j = \{w_{1,j}, w_{2,j}, ..., w_{n,j}\}$ where, $w_{k,j}$ is the weight of term k in document j
- typically, the weight is

$$TF - IDF(t_k, d_j) = \frac{f_{k,j}}{max_z f_{z,j}} \cdot \log \frac{N}{n_k}$$

- typically, similarity is measured using the cosine

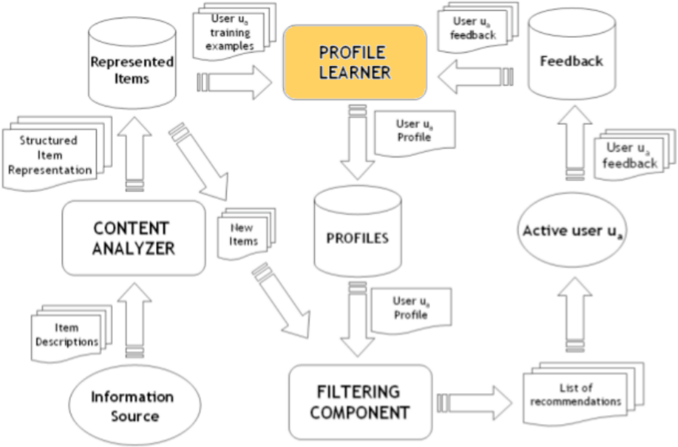
$$sim(d_i, d_j) = \frac{\sum_k w_{k,i} \cdot w_{k,j}}{\sqrt{\sum w_{k,i}^2}\sqrt{\sum w_{k,j}^2}}$$

Document representation using semantic analysis

Documents are represented as,

- Wordnet synset networks that are matched to user profile, also a synset network
- Wordnet synset vector space model using the concept of bag-of-synsets (BOS)
- high-dimensional space of concepts derived from Wikipedia

Profile learner



Profile learner

Collects user documents, from “Represented items” repository, and user feedback. Then tries to generalize the collected data in order to build a profile.

Methods for learning user profile:

- probabilistic methods, e.g., Naïve Bayes
- relevance feedback, Rocchio's algorithm
- decision trees
- nearest neighbor
- clustering

Probabilistic models

Generates a probabilistic model based on previously observed data.

Naïve Bayes

- observes the documents preferred by the user and calculates the parameters of the observed data, typically using
  - multivariate Bernoulli event model
  - multinomial event model
- estimates the *a posteriori* probability, $P(c|d)$ using

$$P(c|d) = \frac{P(c)P(c|d)}{P(d)}$$

- empirical results have shown that the multinomial event model outperforms the mutivariate Bernoulli

Relevance feedback

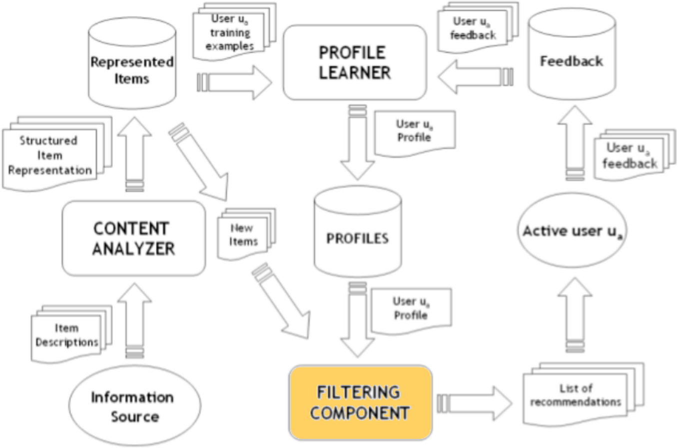
Relevance feedback is a technique that consists of users feeding back into the system on the relevance of retrieved documents with respect to their information needs.

Rocchio's algorithm

- the algorithm calculates the class vector $\vec{c_i} = < \omega_{1,i}, \omega_{2,i}, ..., \omega_{|T|,i} >$, where
  - $\omega_{k,i}$ is the weight o term k in class i
  - T is the vocabulary
- weights are calculated using

$$\omega_{k,i} = \beta \cdot \sum_{\{d_j \in POS_i\}} \frac{\omega_{k,j}}{|POS_i|} - \gamma \cdot \sum_{\{d_j \in NEG_i\}} \frac{\omega_{k,j}}{|NEG_i|}$$

Filtering component



Filtering component

The filtering component matches user profile against document representation to generate a recommendation list of items for the active user.

To find new documents, the filtering component

- searches for the documents that maximize $d = argmax_{d_j} \frac{P(c)P(c|d_j)}{P(d_j)}$ when using Naïve Bayes classification
- compares documents that are similar to $\vec{c_i} =< \omega_{1,i}, \omega_{2,i}, ..., \omega_{|T|,i} >$ when using Rocchio's algorithm

and generates the **list of recommendations**.

User feedback

The active user looks at the recommendation list and gives feedback to recommendation system.

Implicit feedback

- preferences are collected without user explicit intervention
- user activities monitorized and analyzed
  - documents bought/downloaded
  - documents visualized
  - documents bookmarked

Explicit feedback

- user explicit feeds the systems with ratings
- three main approaches
  - binary: like/deslike
  - numeric ratings: 0-5 or totally dislike, moderate dislike, neutral, moderate like, totally like
  - text comments

Advantages

Advantages

- User independence
  Content-based recommenders do not use ratings from other users.
- Transparency
  Explanations can be provided based on features.
- New Item
  New items do not need ratings to be recommended.

Drawbacks

Drawbacks

- Limited content analysis
  If documents are extremely short, e.g., jokes, content many not be enough to classify items.
- New user problem
  In order to get accurate recommendations, the user must have a enough ratings.
- Over-specialization
  The user is going to be recommended documents similar to those already rated by the user.

Novelty vs Serendipity

Novelty

Novelty occurs when the system suggests to the user an unknown item that he might have autonomously discovered.

Serendipity

Serendipitous recommendation helps the user to find a surprisingly interesting item that the user might not have otherwise discovered

Beyond over-specialization

Solutions to surpass over-specialization

- Introduction of some randomness with randomness measures
- Genetic algorithms
- Elimination of items to similar

Conclusion

A high-level architecture content-based recommendation was presented.

Content-based recommenders are not effectively used in real case scenarios due to the over-specialization problem.

Further research in generating serendipitous recommendation is needed.

Content-based recommenders can benefit from further research in NLP, e.g., the use of semantic analysis.