Recommender Systems

Collective Intelligence, Jaar 1

Maarten de Rijke

April 1, 2019

University of Amsterdam

derijke@uva.nl

Today's menu

1. About 60 minutes on recommender systems
2. About 15 minutes on domestic matters

Based on materials made available by Carlos Castillo, Dietmar Jannach, Alexander Felfernig, Gerhard Friedrich, Markus Zanker

- **Connecting people to information**
- Background
- Collaborative filtering
- Content-based methods
- Loose ends
- Conclusion

Access to information a basic human right

Access to information a basic human right

/. All are en-
n violation of
iscrimination.

emedy by the
fundamental

to a fair and
bunal, in the
any criminal

others and in public or private, to manifest his religion or belief in teaching, practice, worship and observance.

ARTICLE 19 —Everyone has the right to freedom of opinion and expression; this right includes freedom to hold opinions without interference and to seek, receive and impart information and ideas through any media and regardless of frontiers.

ARTICLE 20 —1. Everyone has the right to freedom of peaceful assembly and association.

2. No one may be compelled to belong to an association.

ARTICLE 21 —1. Everyone has the right to take part in the government of his country, directly or through freely chosen

2. Educa
personal
and func
ance and
and shal
tenance

3. Parent
that shal

ARTICLE

cultural
scientific

2. Everyo

Information retrieval

Technology to connect people to the right information in the right way at the right time

- Search engines
  - Google, library search engine, desktop search engine, search engine on your mobile phone, product search...
- Recommender systems
  - Product recommendation, movie recommendation, date recommendation, news recommendation, music recommendation, ...
- Digital assistants
  - Alexa, Siri, Google, bots on web sites, bots for consumer services, ...

Search engines

User enters a query to express their information need

Broad collection of items ("the web") vs narrow collection (scientific articles, your email)

Understand user intent and rank items based on intent analysis, interaction data, behavioral analysis, analysis of document content, popularity, temporal factors, ...

Typically returns a ranked list of items, with the best items ranked at the top

Recommender systems

User logs on

In case there is a broad collection, facets may help to narrow down space of recommended items

Recommender system may use short-term interests (current session) and long-term user interests (previous sessions) and exploit implicit or explicit interaction data

Typically returns a broad range of options for user to select from

Digital assistants

There may or may not be a screen or keyboard

System asks clarification questions

- life is **easier** for the system: can ask for help
- life is **harder** for the system: needs to ask the right question and understand user response

Uses query information, profile information, information gleaned from interactions (implicit and explicit)

Typically returns a single result ("the answer") at a time

Connecting people to information

Background

Collaborative filtering

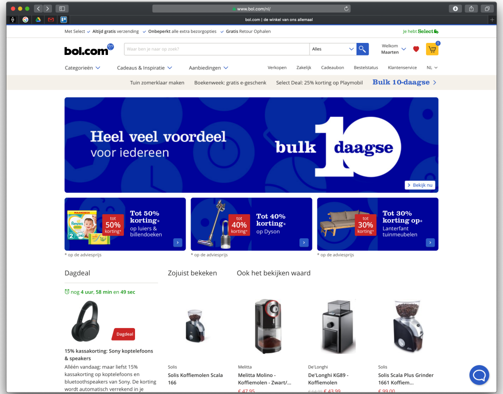
Content-based methods

Loose ends

Conclusion

Why recommender systems?

Recommender systems are software tools and techniques that provide suggestions for items to be of use to a user



Assumptions and users

Assumptions

- Users rely on recommendations
- Users lack sufficient personal expertise
- Number of items is very large
  - $> 16M$ in bol.com
- Recommendations need to be personalized

Who uses recommender systems?

- Retailers and e-commerce in general
  - bol.com, Netflix, etc.
- Service sites, e.g. travel sites
- Media organizations
- Dating apps
- ...

Why and what?

Why?

- Increase number of items sold
  - 2/3 of Netflix watched are recommendations
  - 1/3 of Amazon sales are from recommendations
- Sell more diverse items
- Increase user satisfaction
  - users enjoy the recommendations
- Increase user fidelity
  - users feel recognized (but not creeped out)
- Understand users (see below)

Recommendations generate by-products

- Recommending requires understanding users and items, which is valuable by itself
- Some recommender systems are very good at this (e.g. factorization methods)
- Automatically identify marketing profiles
- Describe users to better understand them

The core problem

Estimate the **utility** for **a user** of **an item** for which the user has **not expressed** utility

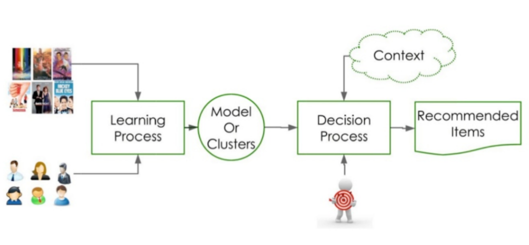
What does this mean?

- Find some good items (most common)
- Find all good items
- Annotate in context (why I would like this)
- Recommend a sequence (e.g., tour of a city)
- Recommend a bundle (camera+lens+bag)
- Support browsing (seek longer session)
- . . .

Source and process

Data soruces

- Items, Users
  - structured attributes, semi-structured or unstructured descriptions
- Transactions
  - appraisals
    - numerical ratings (e.g., 1-5)
    - binary ratings (like/dislike)
    - unary ratings (like/don't know)
  - sales
  - tags/descriptions/reviews



The recommendation process

Aspects

- Data preparation
  - normalization, removal of outliers, feature selection, dimensionality reduction, ...
- Data mining
  - clustering, classification, rule generation, ...
- Post-processing
  - visualization, interpretation, meta-mining, ...

Desiderata

- Must inspire trust
- Must convince users to try the items
- Must offer a good combination of novelty, coverage, and precision
- Must have a somewhat transparent logic
- Must be user-tunable

So many approaches . . .

- Collaborative filtering
- Content-based (item features)
- Knowledge-based (expert system)
- Personalized learning to rank
- Estimate ranking function Demographic
- Social/community based
- Based on connections
- Hybrid (combination of some of the above)

Connecting people to information

Background

Collaborative filtering

Content-based methods

Loose ends

Conclusion

Collaborative Filtering (CF)

- The most prominent approach to generate recommendations
  - used by large, commercial e-commerce sites
  - well-understood, various algorithms and variations exist
  - applicable in many domains (book, movies, songs, . . .)
- Approach
  - use the "wisdom of the crowd" to recommend items
- Basic assumption and idea
  - users give ratings to catalog items (implicitly or explicitly)
  - customers who had similar tastes in the past, will have similar tastes in the future

Pure CF Approaches

- Input
  - only a matrix of given user-item ratings
- Output types
  - a (numerical) prediction indicating to what degree the current user will like or dislike a certain item
  - a top-N list of recommended items

User-based nearest-neighbor collaborative filtering (1)

- The basic technique
  - Given an "active user" (Alice) and an item i not yet seen by Alice
    - find a set of users (peers/nearest neighbors) who liked the same items as Alice in the past and who have rated item i
    - use, e.g., the average of their ratings to predict, if Alice will like item i
    - do this for all items Alice has not seen and recommend the best-rated
- Basic assumption and idea
  - If users had similar tastes in the past they will have similar tastes in the future
  - User preferences remain stable and consistent over time

User-based nearest-neighbor collaborative filtering (2)

- Example
  - A database of ratings of the current user, Alice, and some other users is given:

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

  - Determine whether Alice will like or dislike *Item5*, which Alice has not yet rated or seen

User-based nearest-neighbor collaborative filtering (3)

- Some first questions
  - How do we measure similarity?
  - How many neighbors should we consider?
  - How do we generate a prediction from the neighbors' ratings?

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Measuring user similarity (1)

- A popular similarity measure in user-based CF: Pearson correlation
  - a, b: users
  - $r_{a,p}$: rating of user a for item p
  - P: set of items, rated both by a and b
  - $\bar{r}_a$: mean rating provided by a

Possible similarity values between -1 and 1

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2}}$$

Measuring user similarity (2)

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

sim = 0,85

sim = 0,00

sim = 0,70

sim = -0,79

Making predictions

- A common prediction function:

$$pred(a,p) = \frac{\bar{r}_a + \sum_{b \in N} sim(a,b) \cdot (r_{b,p} - \bar{r}_b)}{\sum_{b \in N} sim(a,b)}$$

- Calculate, whether the neighbors' ratings for the unseen item i are higher or lower than their average
- Combine the rating differences – use the similarity with a as a weight
- Add/subtract the neighbors' bias from the active user's average and use this as a prediction

Improving the metrics/prediction function

- Not all neighbor ratings might be equally “valuable”
  - Agreement on commonly liked items is not so informative as agreement on controversial items
  - Possible solution: Give more weight to items that have a higher variance
- Value of number of co-rated items
  - Use “significance weighting,” by e.g., linearly reducing the weight when the number of co-rated items is low
- Case amplification
  - Intuition: Give more weight to “very similar” neighbors, i.e., where the similarity value is close to 1.
- Neighborhood selection
  - Use similarity threshold or fixed number of neighbors

Memory-based and model-based approaches

- User-based CF is said to be "memory-based"
  - the rating matrix is directly used to find neighbors / make predictions
  - does not scale for most real-world scenarios
  - large e-commerce sites have tens of millions of customers and millions of items
- Model-based approaches
  - based on an offline pre-processing or "model-learning" phase
  - at run-time, only the learned model is used to make predictions
  - models are updated / re-trained periodically
  - large variety of techniques used
  - model-building and updating can be computationally expensive
  - **item-based CF** is an example for model-based approaches
    - use similarity between items (not users) to make predictions

Weaknesses

- Assumes standardized products
  - E.g., a touristic destination at any time of the year and under any circumstance is the same item
- Does not take into account context
- Requires a relatively large number of transactions to yield reasonable results
- **Cold-start problem**
  - What to do with a new item?
  - What to do with a new user?

Connecting people to information

Background

Collaborative filtering

Content-based methods

Loose ends

Conclusion

Content-based recommendation

- While CF-methods do not require any information about the items
  - it might be reasonable to exploit such information; and
  - recommend fantasy novels to people who liked fantasy novels in the past
- What do we need:
  - some information about the available items such as the genre ("content")
  - some sort of user profile describing what the user likes (the preferences)
- The task:
  - learn user preferences
  - locate/recommend items that are "similar" to the user preferences

What is the content?

- Most CB-recommendation techniques were applied to recommending text documents
  - Like web pages or newsgroup messages for example
- Content of items can also be represented as text documents.
  - With textual descriptions of their basic characteristics
  - Structured: each item is described by the same set of attributes

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism

  - Unstructured: free-text description

Content representation and item similarities

- Item representation

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism

- User profile

Title	Genre	Author	Type	Price	Keywords
...	Fiction	Brunonia, Barry, Ken Follett	Paperback	25.65	Detective, murder, New York

- Simple approach
  - Compute similarity of unseen item with user profile based on keyword overlap (e.g., Dice coefficient) $$\frac{2 \cdot |keywords(b_i) \cap keywords(b_j)|}{|keywords(b_i) + keywords(b_j)|}$$
  - Or use and combine multiple metrics

Term-Frequency - Inverse Document Frequency (1)

- Simple keyword representation has its problems
    - in particular when automatically extracted as
        - not every word has similar importance
        - longer documents have a higher chance to have an overlap with the user profile
- Standard measure: TF-IDF
    - Encodes text documents in multi-dimensional Euclidian space
        - weighted term vector
    - TF: Measures, how often a term appears (density in a document)
        - assuming that important terms appear more often
        - normalization has to be done in order to take document length into account
    - IDF: Aims to reduce the weight of terms that appear in all documents

Term-Frequency - Inverse Document Frequency (2)

- Given a keyword i and a document j
- $TF(i,j)$
  - term frequency of keyword i in document j
- $IDF(i)$
  - inverse document frequency calculated as $IDF(i) = \log \frac{N}{n(i)}$
    - N: number of all recommendable documents
    - $n(i)$: number of documents from N in which keyword i appears
- $TF\text{-}IDF(i,j)$
  - $TF\text{-}IDF(i,j) = TF(i,j) \cdot IDF(i)$

Example TF-IDF representation

Term frequency

- Each document is a count vector in $\mathbb{N}^d$

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	1.51	0	3	5	5	1
worser	1.37	0	1	1	1	0

Combined TF-IDF weights

- Each document is a real-valued vector of TF-IDF weights in $\mathbb{R}^d$

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

Improving the vector space model (1)

- Vectors are usually long and sparse
- Remove stop words
  - They will appear in nearly all documents.
  - e.g. “a”, “the”, “on”, ...
- Use stemming
  - Aims to replace variants of words by their common stem
  - e.g., “wen” $\Rightarrow$ “go”, ”stemming” $\Rightarrow$ “stem”, ...
- Size cut-offs
  - only use top n most representative words to remove “noise” from data
  - e.g. use top 100 words

Improving the vector space model (2)

- Use lexical knowledge, use more elaborate methods for feature selection
  - Remove words that are not relevant in the domain
- Detection of phrases as terms
  - More descriptive for a text than single words
  - e.g. “United Nations”
- Limitations
  - meaning remains unknown
  - example: usage of a word in a negative context
    - “there is nothing on the menu that a vegetarian would like...”
    - the word “vegetarian” will receive a higher weight then desired
    - $\Rightarrow$ an unintended match with a user interested in vegetarian restaurants

Limitations of content-based recommendation methods

- Keywords alone may not be sufficient to judge quality/relevance of a document or web page
  - up-to-date-ness, usability, aesthetics, writing style
  - content may also be limited / too short
  - content may not be automatically extractable (multimedia)
- Ramp-up phase required
  - Some training data is still required
  - Social media: Use other sources to learn the user preferences
- Overspecialization
  - Algorithms tend to propose “more of the same”
  - Or: too similar news items

Wrapping up

- In contrast to collaborative approaches, content-based techniques do not require user community in order to work
- Presented approaches aim to learn a model of user's interest preferences based on explicit or implicit feedback
  - Deriving implicit feedback from user behavior can be problematic
- Evaluations show that a good recommendation accuracy can be achieved with help of machine learning techniques
  - These techniques do not require a user community
- Danger exists that recommendation lists contain too many similar items
  - All learning techniques require a certain amount of training data
  - Some learning methods tend to overfit the training data
- Pure content-based systems are rarely found in commercial environments

Connecting people to information

Background

Collaborative filtering

Content-based methods

Loose ends

Conclusion

So many more approaches . . .

- Collaborative filtering
- Content-based
- Knowledge-based (expert system)
- Personalized learning to rank
  - Estimate ranking function Demographic
- Social/community based
  - Based on connections
- Hybrid (combination of some of the above)
- Ensembles
  - Machine learned combinations of large numbers of the above

Human factors

- Advanced systems are conversational
  - Transparency and scrutability
  - Explain users how the system works Allow users to tell the system it is wrong
- Help users make a good decision
- Convince users in a persuasive manner Increase enjoyment to users
- Provide serendipity
  - “An aptitude for making desirable discoveries by accident”
  - Don’t recommend items the user already knows Delight users by expanding their taste
    - But still recommend them something somewhat familiar
- Provide diversity

Evaluation

- User experiments
- Precision @ Cut-off
- Ranking-based metrics
  - E.g. Kendall's Tau Score-based metrics
- Score-based metrics
  - E.g. RMSE

Connecting people to information

Background

Collaborative filtering

Content-based methods

Loose ends

Conclusion

Conclusion

Three main ways of connecting people to information: search, recommendation, digital assistants

Recommender systems widely: collaborative filtering, content-based approaches are dominant, along with ensembles

Recommender systems are increasingly conversational

Likely to become even more widespread as we collect more data

References i

F. Ricci, L. Rokach, and B. Shapira. Introduction to recommender systems handbook. In *Recommender Systems Handbook*. Springer, 2010.