Intelligent Systems

L 7. Recommender Systems (1)

Antonio Bahamonde
Departamento de informática

Universidad de Oviedo en Gijón

Contents

- Definitions, goals and success criteria
- Application examples (Search, Music, e-commerce, …)
- Types of algorithms
  - Collaborative filters
  - Content based and hybrid

RS: Definitions

RS are algorithms that learn the interests and preferences of each consumer and make recommendations that are tailored to their tastes.

RS help connect users and items

- They facilitate the management of the enormous amount of information
- They act as assistants in sales (guide, advice, persuasion, ...)

There are different types depending on the information they handle:

- Implicit or explicit feedback from users (collaborative filters)
- Item characteristics (content-based)

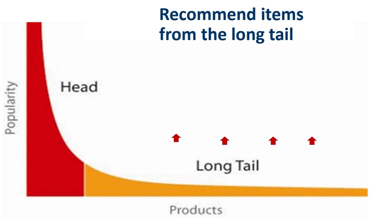
Goals

Information retrieval

- Suggest the correct proposals - (Users know in advance what they want?)

Recommendation diversity

- Serendipity (valuable find that occurs by chance or accidental) - identifies items that users did not even know existed



Success criteria

Precision
- the degree to which each user will like an item
- is the most popular criterion for evaluating RS

Interaction
- give users a good feeling - Educate users about a type of item
- convince or persuade users - explain

Finally, commercial
- increase visits and navigation, the rate of sales per visit
- optimize sales margins and bottom line

Syllabus

synonyms used

Users, consumers, clients
Items, products
Recommend, suggest

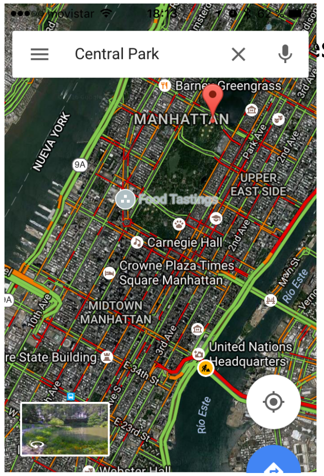
Applications fields

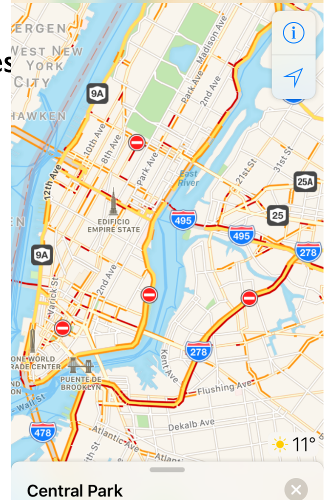
- Google
  - Waze, Mapas
- Musicales
  - Spotify
  - Apple Music
  - last.fm
  - Playlist Prediction for Local Music Discovery
- Electronic Commerce
  - Amazon
  - Alibaba

RS: route indications

Get the best route, every day, with real-time help from other drivers.

Acquired by Google for 966 Million dollars (June 2013), it becomes a Google Mobile Service service (March 2015)





Real time advise about congestions

Music reommendations

- Spotify
- Apple Music
- last.fm
- Playlist Prediction for Local Music Discovery

e-Commerce

- Amazon
- Alibaba

Algorithms that recommend

- Collaborative filters (no contents)
  - Items
  - users
- Content based and hybrid

Collaborative filters

The most used algorithms

- e - commerce, music, cinema, ...
- numerous algorithms with many variants
- the application domain is not important (you do not use it): they can be applied in all

Basic principles

- use the so-called "wisdom of the crowd" to make recommendations (crowdsourcing)
- users rate the items in a catalog implicitly or explicitly
- customers who have shown similar tastes in the past are assumed to have similar tastes in the future

Other collaborative filters in pre-digital era

Popular recommenders in everyday life

* Bestsellers lists
* Music hits lists
* The "recent returns" shelf in libraries
* Paths not labelled but well marked in the woods
* Most read articles in newspapers
* ....

Collaborative filters

They relate 2 different entities: users and items through the evaluations

- explicit (users rate items)
- implicit (purchase, search or browsing history, ...)

Methodologies:

- neighborhood
- latent factors

Neighborhood in collaborative filters

Item-Item user-user

	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

Compute the similarity to make predictions

Neighborhood in collaborative filters

user-user

	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

Neighborhood in collaborative filters

How to estimate the similarity of users?

Correlation $\rho(\boldsymbol{a}, \boldsymbol{b})$

Cosine $\cos(\boldsymbol{a}, \boldsymbol{b}) = \frac{\langle \boldsymbol{a}, \boldsymbol{b} \rangle}{\|\boldsymbol{a}\| \|\boldsymbol{b}\|}$

$$b(u,i) = \overline{r(u,\cdot)} + \frac{\sum_{u' \neq u} sim(u,u')(r(u',i) - \overline{r(u',\cdot)})}{\sum_{u' \neq u} sim(u,u')}$$

There is an analogous expression for item-item

Neighborhood in collaborative filters

User-user with 2 neighbors

$\overline{r(u,\cdot)}$	$\rho(alice,u)$		item1	item2	item3	item4	item5
4,00		alice	5	3	4	4	?
2,25	0,853	user1	3	1	2	3	3
3,50	0,707	user2	4	3	4	3	5
3,00	0	user3	3	3	1	5	4
3,25	-0,792	user4	1	5	5	2	1

$$b(alice, item5) = 4 + \frac{(3 - 2.25) * 0.853 + (5 - 3.5) * 0.707}{0.853 + 0.707} = 5.09$$

Neighborhood in collaborative filters

Item-Item

	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

Neighborhood in collaborative filters

Item-Item
2 neighbors

	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
$\rho(item5, i)$	0.969	-0.478	-0.428	0.582	
$\overline{r(\cdot, i)}$	3.2	3.0	3.2	3.4	3.25

$$b(alice, item5) = 3.25 + \frac{(5-3.2)*0.969 + (4-3.4)*0.582}{0.969+0.582} = 4.6$$

Neighborhood in collaborative filters

Scalability

- user-user is a memory-based method
- if there are millions of users, it cannot be used
- not scalable to most real world scenarios

Neighborhood in collaborative filters

item-item

- is based on a model
- models learned offline are stored to make predictions in real time
- allow explanations
- they don't have the cold start problem

Neighborhood in collaborative filters

However, in all cases

- not all neighborhoods should be considered (threshold of similarity)
- not all items are scored
- these methods do not use a loss function that they try to minimize (its purpose is not clear)

Content based and hybrids

They explicitly use a description of the content of the items to be recommended.

Hybrids use combinations of elements from collaborative and content-based filters