Recommender Systems

Electronic Business Technologies

FEUP

Recommender Systems

- In e-markets there is too much information available, and consumers have an overwhelming number of alternatives
- Would be helpful:
  - narrow down users' choices
  - make choices without having sufficient personal experience or information

- Recommender System (RS)

  system that provides recommendations to a user
  - Amazon.com makes use of a recommendation system when you select an item *X*, (Amazon provides a list of items that have been purchased by other users who have also purchased item *X*)

FEUP

Recommender Systems

- Information used by RSs:
  - Purchase data
    purchase history of users
  - Feedback provided by users
    feedback can be provided explicitly or implicitly (ex: playlist)
  - Textual comments
  - Browsing and searching data
    observe user's behaviour and infer their preferences
  - Expert recommendations
  - Demographic data
    - age, gender, education, geographical location, …
  - …

FEUP

Recommendation

- Recommendation problem
  - Problem of estimating *ratings* for the items that have not been seen by a user

$$\forall c \in C, \qquad s'_c = \max_{s \in S} u(c,s)$$

C – set of all users
S – set of all possible items that can be recommended
$u(c,s)$ – utility function (usefulness of item s to user c)

- Recommender systems predict:
  - absolute values of ratings that users would give to items → *rating-based systems*
  - the relative preferences of users → *preference-based filtering*

FEUP

Recommendation

- Each element of the user space *C* can be defined with a profile
  - user characteristics, such as age, gender, income, …
  - user preferences and needs
  - can be elicited explicitly (questionnaires) or implicitly (behavior over time)

user profile

- Each element of the item space *S* can be defined with a set of characteristics:
  - domain-dependent attributes

item profile

FEUP

5

Recommender Systems

- Classified into:
  - *Content-based recommendations*

    recommend items based on the ones the user preferred in the past and user profile
  - *Collaborative recommendations*

    recommend items that people with similar tastes and preferences liked in the past
  - *Hybrid approaches*

    combine collaborative and content-based methods

FEUP

6

Content-based recommendation

u(c,s) estimated based on $u(c,s_i)$

$u(c,s_i)$: utility assigned by user *c* to items s_i

$s_i \in S$ and s_i is similar to item *s*

u(c,s) = score(ContentBasedProfile(*c*), Content(*s*))

ContentBasedProfile(c) : user *c* profile

Content(s) : item *s* profile

ex: a movie recommender will probably recommend a new Woody Allen film to a Woody Allen fan

Property: Do not depend on having other users in the system (do not need a critical mass)

FEUP

Content-based recommendation (1)

u(c,s) can be estimated by the cosine distance between user's and item's profile vectors

Consider a movie recommender system

M1 profile = (1,0,1,0,1)   % actors: Julia, Peter, Mark, Jennifer, Tom

utility matrix (if not boolean, we can normalize it)

	M1	M2	M3	M4	…	avrg
A	3	4	5		…	3
B	2	3		5	…	4

M1, M2, M3, M4 have Julia as actress

UserA profile = (1,…..)   % (likehood of movies with Julia, …)

UserB profile = (-2/3, …)

FEUP

Content-based recommendation (1)

User profile:

- positive values in components related to actors that appear in movies the user likes
- negative values in components related to actors that appear in movies he doesn´t like

Consider a movie Mn

- Mn profile = (1,0,1,1,1) % with many actors the user X likes

  cosine of the angle (UserA, MX) will be a large positive fraction (angle close 0)

- Mn profile = (0,0,1,1,1) % with mostly actors the user X doesn´t like

  cosine of the angle (UserA, MX) will be a large negative fraction (angle around 180)

- Mn profile = (1,0,1,0,1) % with as many actors the user X likes as doesn´t like

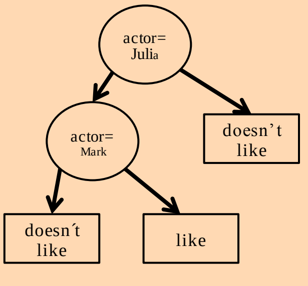
  cosine of the angle (UserA, MX) is around 0 (angle around 90)

Content-based recommendation (2)

u(c,s) can be estimated using machine learning

- the given data is the training set
- for each user, build a classifier that predicts the rating of all items

Possible classifiers: *decision tree, neural network, …*



Content-based recommendation

Limitations:

- Items profiles should contain a sufficient set of features
  - IR techniques work well in extracting features from text documents, but other domains are much harder (multimedia data, …)
- Over-specialization
  - limited to recommend items that are similar to those already rated

  res: often addressed by introducing some randomness
- New items
  - The user has to rate a sufficient number of items before the system can really understand its preferences

Collaborative recommendation

u(c,s) estimated based on *u(c',s)*

u(c',s): utility assigned by user *c'* to items *s*

$c' \in C$, and *c'* is similar to *c*

Two general classes:

- *Memory-based (heuristic-based)*:

  heuristics that make rating predictions based on previously rated items by the users
- *Model-based*

  use previous ratings to learn a *model*, which is then used to make rating predictions

Collaborative recommendation (memory-based)

Let be $\mathbf{r_{c,s}}$ the unknown rating for user *c* and item *s*

$\mathbf{r_{c,s}}$ computed as an <u>aggregation</u> of the ratings of some other users (usually, the N most similar) for item *s*

$r_{c,s} = aggr_{c' \in Cs}\, r_{c',s}$

Cs: set of *N* <u>most similar users</u> to *c* who have rated *s*

$\boldsymbol{sim(c,c')}$: simmilarity between users *c* and *c'*

is a distance measure, and is used as a weight

Collaborative recommendation (memory-based)

– Some examples of aggregation functions:

i) $r_{c,s} = \frac{1}{N} \sum_{c' \in Cs} r_{c',s}$

ii) $r_{c,s} = k \sum_{c' \in Cs} sim(c,c') \times r_{c',s}$

k is a normalizing factor, $k = 1/\sum_{c' \in Cs} |sim(c,c')|$

iii) $r_{c,s} = \overline{r_c} + k \sum_{c' \in Cs} sim(c,c') \times (r_{c',s} - \overline{r_{c'}})$

$\overline{r_c} = (1/|S_c|) \sum_{s \in S_c} r_{c,s}$, $S_c = \{s \in S | r_{c,s} \neq \emptyset\}$

Colaborative recommendation (memory-based)

- Similarity (some approaches):
  - *Correlation:* uses the Pearson correlation coefficient

$$sim(c,c') = \frac{\sum_{s\in S_{cc'}}(r_{c,s}-\bar{r}_c)(r_{c',s}-\bar{r}_{c'})}{\sqrt{\sum_{s\in S_{cc'}}(r_{c,s}-\bar{r}_c)^2 \sum_{s\in S_{cc'}}(r_{c',s}-\bar{r}_{c'})^2}}$$

can also be used to compute simmilarity between items

$S_{cc'}$: set of all items corated by *c* and *c'*

  - *Cosine-based:* cosine of the angle between vectors that represent the 2 users in m-dimensional space ($m = |Scc'|$)

$$sim(c,c') = \cos(\vec{c},\vec{c'}) = \frac{\sum_{s\in S_{cc'}} r_{c,s}\ r_{c',s}}{\sqrt{\sum_{s\in S_{cc'}} r_{c,s}^2}\sqrt{\sum_{s\in S_{cc'}} r_{c',s}^2}}$$

  - *Clustering*

FEUP

Colaborative recommendation (memory-based)

	Item1	Item2	Item3	Item4	Item5
User A	4	4	1	4	3
User B	2	1	4	2	5
User C	3	1	3	2	1
User D	5	4	2		3

$\overline{r_D} = 3.5$
$\overline{r_A} = 3$
$\overline{r_B} = 3$
$\overline{r_C} = 2$

using Pearson correlation:

sim(D,A) = 0.9 ; sim(D,B) = -0.7 ; sim(D,C) = 0

using aggregate funtion: $r_{c,s} = \overline{r_c} + k\sum_{c'\in C_S} sim(c,c') \times (r_{c',s} - \overline{r_{c'}})$

$$\overline{r_{D,4}} = 3.5 + 0.625 * \left(0.9 * (4-3) - 0.7 * (2-3)\right) = 4.5$$

$k = 1/(0.9 + 0.7 + 0)$ = 0.625 (using all neighbours)

FEUP

Colaborative recommendation (model-based)

Uses the collection of ratings to learn a *model*, which is then used to predict ratings

- Probabilistic approach:

$$r_{c,s} = E(r_{c,s}) = \sum_{i=0}^{n} i \times \Pr(r_{c,s} = i | r_{c,s'}, s' \in S_c)$$

  - *Pr* can be estimated by: cluster models, Bayesian networks

- Machine learning approach:
  - Neural networks, Decision trees, …

Collaborative recommendation

Limitations

- New user problem
  - The user has to rate a sufficient number of items before the system can really understand its preferences

  res: use item popularity ; item entropy ; user personalization ; hybrid recommendation

- New item problem
  - New item needs to be rated by a substantial number of users

  res: hybrid recommendation

- Sparsity
  - Recommendation depends on the availability of a critical mass of users

  res: use user profile information for sim(c,c´) ; SVD (Singular Value Decomposition) for sparce ratings matrices

Collaborative recommendation

Limitations

– Reliability
  - Ratings can be artificially inflated by phantom users

– Users' privacy
  - The more information a system has about users' preferences and tastes, the better it can provide recommendation . But users may not want their habits known

res: anonymous participation

FEUP

Hybrid methods

Combine *collaborative* and *content-based* methods:

1) Combining separate recommenders
   - Combine the outputs (ratings) from individual RSs using a linear combination or a voting scheme
   - Use one of the individual RSs, one that is "better" (higher level of confidence, more consistent with past ratings of the user, …)

FEUP

Hybrid methods

Combine *collaborative* and *content-based* methods:

2) Adding content-based characteristics to collaborative methods
   - for instance, include content-based profiles for each user

     -> used to calculate *sim(c,c´)*

     -> recommend item that scores highly against the user's profile

3) Adding collaborative characteristics to content-based methods
   - Create a collaborative view of a collection of content-based profiles (for instance, user profiles)

Hybrid methods

Combine *collaborative* and *content-based* methods:

4) Developing a single unifying recommendation model
   - for instance, use the profile information of users and items in a single statistical model that estimates unknown ratings:

$$r_{cs} = x_{cs}\mu + z_c\gamma_s + \omega_s\varphi_c + e_{cs}$$

random variables for noise, user and item heterogeneity (normal distributions)

vector of item characteristics

vector of user characteristics

matrix of user and item characteristics

   - Use of *case-based reasoning* to improve recommendation accuracy

RSs and e-commerce

Recommender Systems used in e-commerce

- Helping users find products/services
- Cross-selling
  - suggesting additional products for the user to purchase
- Personalization
  - Enables vendors to offer personalized products
- Keeping users informed
  - Keep users up-to-date on current offers and new products
- Retaining user loyalty
  - RSs create a valeu-added relationship between the site and the user.

FEUP

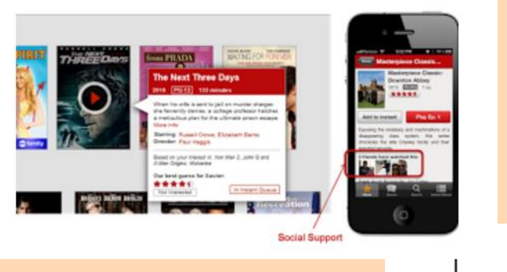
Recommender Systems

Some examples:

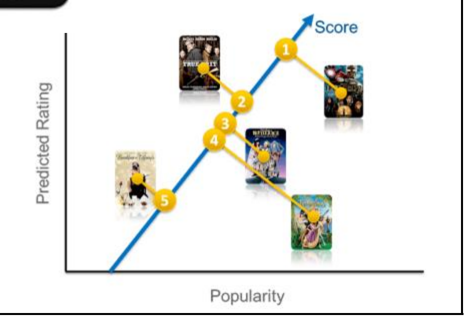
- Linkedin
  - Recommendations for people, jobs, groups or companies (collaborative filtering)
- Amazon
  - Recommends items other users purchased (collaborative filtering)
- Netflix
  - Video rental and streaming service (content-based and collaborative filtering)

FEUP

Recommender System - Netflix



Promotes trust by provide explanations



Ranking = Predicted rating + Popularity

FEUP