Last lecture: Information retrieval

- Information retrieval = assign a relevance score to documents in a collection with respect to a query.
- Vector Space Model
  - A dictionary vector assigns to each index in the vector a „word“ (due to pre-processing, can also be stemmed, a key phrase, etc.). **Creating the dictionary vector involves important algorithmic choices**!
  - Document or query vectors contain at each index the value of a function related to the term‘s occurrence and importance in the document or query. **Choosing this function is an important algorithmic choice**!
  - We discussed the TFIDF
- Ranking function: We discussed the cosine similarity – additionally, weights could be used that consider timeliness, source quality etc.
- Natural language processing is crucial to IR, e.g., tokenization, stemming, phrase detection, word sense disambiguation, and synonym matching; Information extraction, and question answering as extension to IR.

Recommende Systems

Viktoria Pammer-Schindler – based on earlier slides by Angela Fessl, and using an example from Jannach et al. – Recommender Systems: An Introduction.

Introduction to Data Science and Artificial Intelligence

Learning Goals

- Define the computational task of recommendation
- Explain what a user model is
- Carry out user-based and item-based collaborative filtering.
- Discuss user-based and item-based collaborative filtering, and compare the two.
- Discuss recommendation in relationship to information retrieval.

Recommendation

Given

- Set of users U and set of items I
    - Computational representation of users and items are an algorithmic choice!
- An item unkown I_{NEW} to User U_0

Do

- Assign a relevance score to the item (used for ranking)

$$r = f(U, I, U_0, I_{NEW})$$

Different recommender systems paradigms

- **Collaborative filtering** – base recommendation on user interactions with items in a system (e.g.: user ratings, clicks, purchase)
- Content-based recommendation – base recommendation on description of users and items in terms of “content”
  - *Example: Recommend new fantasy novel to a fantasy fan – based on metadata (category, keywords) or content analysis.*
  - *~similarity between user‘s interest and knowledge and content-wise description of items.*
- Knowledge-based – explicitly modelled constraints on items
  - *Example: Facets – English books, audio books, new books, price range, breakfast included, WLAN free, …*
  - *~similarity/match between explicit constraints and items.*

In practice: Hybrid approaches – mixing approaches, tweaking to specific use case

Collaborative filtering - Overview

SCIENCE
PASSION
TECHNOLOGY

Collaborative Filtering

Core assumptions:

- Two kinds of entities: Users and items
  - Examples: **books**, music pieces, hotels, airlines, potential partners…
- Interactions between users and items via online platform
  - Examples: **rating**, viewing/clicking, buying, …

➢ For every user-item interaction, there is an entry in a matrix – User-item matrix.

Example

- A database of users and items
- Example:
  - We are interested in Alice, and want to predict how she would like Item5.
  - Given: We know Alice's ratings for Item1, Item2, Item3, Item4; and ratings from User1, User2, User3, User4 for all items

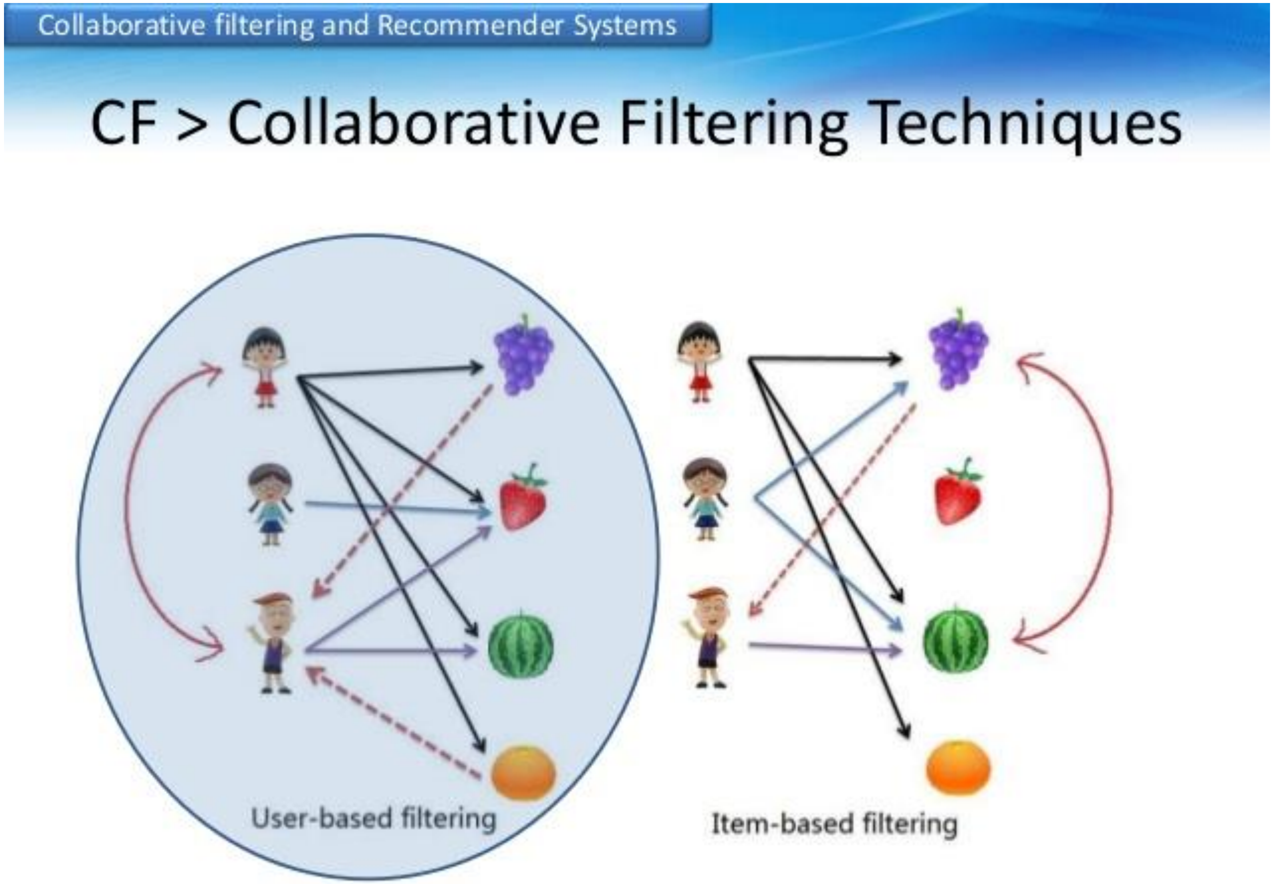
	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Example from Jannach et al. – Recommender Systems An Introduction, see last slide

:

- Which vector describes Alice? Which User1, User2, User3, User4?
- Which vector describes Item5? Which Item1, Item2, Item3, Item4?
  - Note: Alice and Item5 have missing values at index 5 / index 1 respectively!

Collaborative filtering



https://github.com/Scorpi35/Collaborative-Filtering

User-based collaborative filtering

User-based collaborative filtering

	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

Idea:

- If users have rated items similarly in the past, their predications are likely to be similar in the future
- ➢ Find users who are similar to Alice in terms of which items they like
- ➢ Predict Alice's future rating of new item based on ratings of similar users (use a threshold for identifying similar users)

User-based collaborative filtering

	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

Idea: Similar users rate items similarly.
Transfer knowledge on a new item from similar users to U_0

Therefore:

- Find users who are similar to U_0 (Alice) in terms of which items they like
  - TODO: compute pairwise similarities between Alice and all other users
- Predict U_0 's (Alice) future rating of new item based on ratings of similar users (use a threshold for identifying similar users)
  - TODO: predict how U_0 (Alice) will rate the new item.
  - This prediction is used to decide on whether item is recommended or not, in ranking recommender results, or for some other system reaction.

Similarity Idea 1: Cosine similarity of user vectors

	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

- Disadvantage: Users have different tendencies in rating
- ➢ Similarity Idea 2: Normalize user ratings by each user's average rating value (centered user vectors = mean value of vector elements is 0)

Similarity Idea 2: Cosine similarity of centered user vectors

a, b : users

$r_{a,p}$: rating of user a for item p

$\bar{r}_a$: average rating of user a across P

P : set of items, rated both by a and b

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

- Possible similarity values between -1 and 1
- Interpretation of sim(a,b)
  - **Pearson correlation** - Correlation of two variables a,b
  - Cosine of angle between two centered vectors a,b

Prediction

- Common prediction function for user-based collaborative filtering

$$\boldsymbol{pred(a,p) = \overline{r_a} + \frac{\sum_{b \in N} sim(a,b) * (r_{b,p} - \overline{r_b})}{\sum_{b \in N} sim(a,b)}}$$

Idea:

- Set of most similar users (neighbours) N
- Combine their deviation of ratings for the new item in comparison to their average ratings
- … with the their similarity to user a
- … and add/subtract this value from user a's average rating.

Exercise 12

User-based collaborative filtering for Example from Slide 15

1. Compute the pairwise similarities between Alice and Users 1-4 (see slide 16)
2. Choose the two most similar users
3. … and predict a rating for Item 5 for Alice based on the prediction function from slide 17.
4. Decide: Do you recommend the item to Alice?

Tweaking the recommender – two examples

- Similarity: Agreement on controversial items weighs more than agreement on commonly liked/disliked items.
  - Identify controversial items (high variance in ratings), and increase weight of those items in similarity formula
- Prediction: Give more weight to ratings of very similar neighbours (close to 1)

Item-based collaborative filtering

SCIENCE
PASSION
TECHNOLOGY

Item-based collaborative filtering

	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

Idea:

- If users have liked (~rated highly) items in the past, they will like similar items in the future
- ➢ Find items similar to the unknown item
- ➢ … and recommend if these have been liked by the active user in the past.

Item-based collaborative filtering

	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

Idea:

- Users continue to like items they have liked in the past.
- ➢ How similar is a new item to items the user has liked in the past?
  - ➢ TODO: compute pairwise similarities between the unknown item and all other items
- ➢ … recommend new item if it is sufficiently similar
  - ➢ TODO: compute prediction for active user's rating (as measure of relevance)
  - Used to decide on whether item is recommended or not, in ranking recommender results, or for some other reaction

Similarity Idea 1: Cosine similarity of item vectors

	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

- Disadvantage: Users have different tendencies in rating
- ➢ Similarity Idea 2: Normalize user ratings by each user's average rating

Similarity Idea 2: Cosine similarity of normalized item vectors

a, b : items

$r_{u,p}$: rating of user $u \in U$ for item p

$\bar{r}_u$: average rating of user u across P

U : set of users who have rated all items

P : set of items rated by all users

$$\boldsymbol{sim(a,b)} = \frac{\sum_{\boldsymbol{u} \in \boldsymbol{U}} (\boldsymbol{r_{u,a}} - \boldsymbol{\bar{r}_u})(\boldsymbol{r}_{u,b} - \boldsymbol{\bar{r}_u})}{\sqrt{\sum_{u \in U} (\boldsymbol{r}_{u,a} - \boldsymbol{\bar{r}}_u)^2} \sqrt{\sum_{u \in U} (\boldsymbol{r}_{u,b} - \boldsymbol{\bar{r}_u})^2}}$$

- Sometimes called **adjusted cosine similarity**

Prediction

- Common prediction function for item-based collaborative filtering:

$$pred(u,p) = \frac{\sum_{i \in N} sim(i,p) * r_{u,i}}{\sum_{i \in N} sim(i,p)}$$

Idea:

- Set of most similar items (neighbours) N
- Combine ratings of user *u*
- … weighted with the similarity of *i* to the unkown item *p*

Extra question: Why are here the non-normalized $r_{u,i}$ taken?

Answer: Because here we sum over all items, but just one user; and therefore don't need to normalize for this user's bias.

Exercise 13

Item-based collaborative filtering for Example from Slide 15

1. Compute the pairwise similarities between Item 5 and Items 1-4 (see slide 29 – adjust vectors by rating average for each user)
2. Choose the two most similar items
3. … and predict a rating for Item 5 for Alice based on the prediction function from slide 30.
4. Decide: Do you recommend the item to Alice?

Discussion

SCIENCE
PASSION
TECHNOLOGY

Challenges in recommender systems

- Sparsity of user-item matrix if explicit interactions (e.g., ratings) are taken into account
  - Use implicit measures of interest/preference (clicking, buying, …)
  - Spreading activation
- Cold-start problem: What to do with new users or items?
  - Use metadata, content analysis, explicitly stated preferences, „test“ user with high-variance selection of items
- Scale
  - Offline pre-computation; limited size of neighbourhood, thresholds for keeping neighbourhoods small

Core idea in terms of vectors as knowledge representation

Recommender systems as systems that represent

- Users
- Items
- As vectors
- And use vector-based similarity or correlation-measures as basis for identifying the relevance of a particular item to a user (basis for recommendation)

Discussion 1: Relevance of an item I_{NEW} for a user U_0

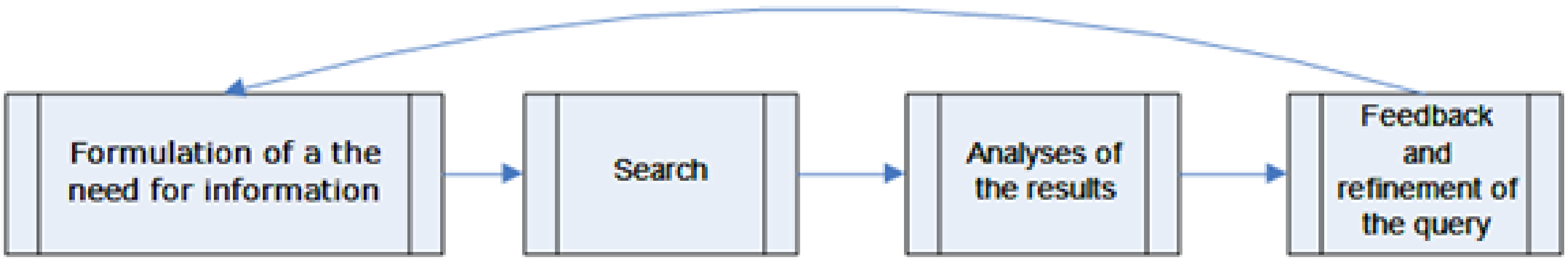
Depending on context of recommender system, **relevance** can mean different things; and hence implementations differ.

Relevance can mean whether a

- User will like an item
- User needs this item (e.g., in educational domain – what does learner need to know? – learning materials and explanations)
- User will click on an item
- User will buy the item

Relevance relates to goals for a socio-technical system that are OUTSIDE the technical system!

Discussion 2 – Relation to Information Retrieval



We don't have, in this sense a query

… but we still want to identify in (SEARCH) a set of items

… RELEVANT items

… and expect to get some feedback & iteration

Discussion 2 – Relation to Information Retrieval

Information Retrieval: Explicit query, retrieve most similar/relevant items from a collection of items

Recommender system: No query, proactive “retrieval” (=recommendation) of most relevant items from a collection of items

Recommended Reading

- Jannach, D.; Zanker, M.; Felferning, A.; Friedrich, G. Recommender Systems: An Introduction. http://recommenderbook.net/ - Chapters 1 (Intro) and 2 (Collaborative Filtering)