Recommender system

MENTOR – MENTEE MATCHING

Objective

- Develop a recommender system framework to match mentors and mentees
  - Explain the design choices
  - Extract topics from the titles of authors
  - Scoring mentors
  - Evaluation of the system

- Data : the DBLP Computer Science Bibliography dataset, available here: http://dblp.uni-trier.de/xml/, where mentors will be the authors and topics will be inferred from the titles of their publications.

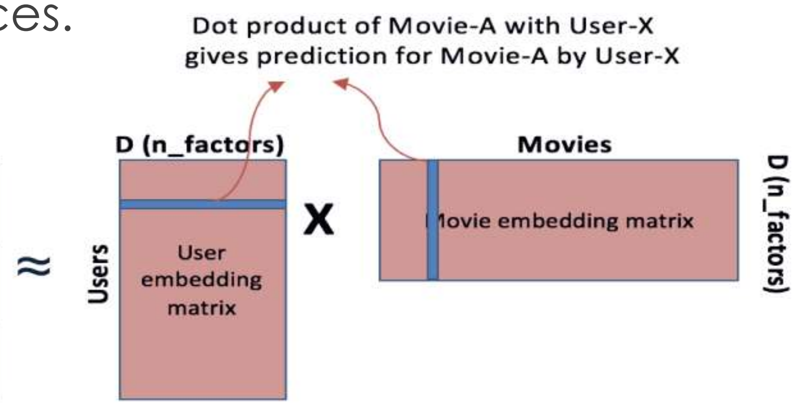
Recommender system design

- **Matrix Factorization (MF):** The idea behind such models is that attitudes or preferences of a user can be determined by a small number of hidden factors. We can call these factors as **Embeddings.**
- **Matrix decomposition can be reformulated as an optimization problem with loss functions and constraints.** Now the constraints are chosen based on property of our model. For e.g. for Non negative matrix decomposition, we want non negative elements in resultant matrices.

Movies

Users

(Sparse matrix)	Waking Life	Boyhood	Before Sunset
Jesse	4.5		4.0
Celine		3.5	5.0



Mentor mentee matrix

- Mentor Matrix: use the Dblp data to extract the main topics from the titles of author's publications and score the author across the various topics generated

- Mentee Matrix:
  - The mentees rate the various topics generated from the topic modelling exercise

- These two matrices can then be used as representing the embeddings for both mentors and mentees

Data processing

- Xml file containing the information about authors and their publications, urls, articles, phd thesis etc.
- A single author can have multiple publications
- Lot of junk titles for some articles like – "Home Page"
- Feed this data into a text processor which would clean it, tokenize it, lemmatize it
- The nltk and regex libraries are used to accomplish this task
- Group multiple publications for an author into a single document
- Remove junk and blank titles

Topic extraction - LDA

- Latent Drichlet Allocation: In natural language processing, latent Dirichlet allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar.
- For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics.
- We use the genism implementation of the LDA model as it is much faster
- Generate 20 topics from the corpus that we have

Mentor expertise

- Use the LDA output to score a document for the author (all the publication titles concatenated)
- We can also score individual titles separately for each author and then add/average them to calculate a single score for every author (didn't get the chance to implement this for now)
- Scale all the scores from 0 to 100. For each topic, the mentor with the highest score gets a score of 100 and the worst gets 0

Mentor recommendation

- Matrix multiplication -
  - Mentor matrix **x** Mentee matrix = Scoring matrix

- We also need to decide if we want a 1-N mentor mentee match or 1-1 match
- 1-N match could make sense in scenarios where the number of mentees outnumber the number of mentors which is generally the case
- 1-1 match would be much harder to implement and would require us to use some sort of stable matching algorithm so that all the mentees get some sort of preference match

Evaluation of the system

- We can use a metric like RMSE to calculate the training an validation scores
- Also we can test edge cases:
  - Mentee with interest in only one topic getting the mentor with the highest expertise
  - Mentee without any preference getting a mentor with varied expertise in multiple topics