GHENT
UNIVERSITY

FACULTY OF ENGINEERING
AND ARCHITECTURE

AANBEVELINGSSYSTEMEN
RECOMMENDER SYSTEMS

Introduction

GHENT
UNIVERSITY

PRACTICAL EXERCISE2: CONTENT-BASED RECOMMENDATIONS

GHENT UNIVERSITY

GOALS

- Develop a content-based recommender
  - Experiment with different ways of processing the dataset
- Implementation in Python or Java (your choice)
  – Clear and documented code
- Run your implementation on the MovieLens Dataset
  – Interpret the results

YOUR RESULTS

- Source code
- Report with answers on the questions
- To be submitted on or before March 23
- On Ufora➔ Assignments

INPUT = MOVIELENS DATA SET

- MovieLens Ratings
- For this practical exercise: 100K Dataset of October 2016
  - Included in the assignment on ufora as attachment
  - Other datasets for other practical exercises!
  - Use the genres of the movies as item vector

PART 1: BASIC CONTENT-BASED RECOMMENDER

- Rescale the ratings to [-2.5, 2]
- User profile = the sum of positive and negative ratings of each genre of rated movies
- Item profile = movie genres as specified by movielens
- Calculate the recommendation score as dot product

$$\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + u_2 v_2 + \ldots + u_n v_n = \sum_{i=1}^{n} u_i v_i$$

PART 2: NORMALIZING THE ITEM VECTORS

- In Part 1: movies tagged with numerous genres could have more influence on the overall profile than one that has only a few
- Normalization in Part 2 of the item vectors:
  - Dividing the values of each feature by the square root of the number of genres

PART 3: IDF

- Problem: The frequency of occurrence can differ greatly between genres
- Calculate the frequencies of each genre (DF) and take the inverse (IDF = 1/DF)
- Rescale the user profiles of **part 2**
- Compute the two way dot product between the rescaled profiles and **the movie vectors of part 2**:

$$U_{idf} * V$$

PART 4: MORE DIVERSE RECOMMENDATIONS

- The content-based recommendations in a list might be very similar to each other
- Goal: obtain a set of recommendations that is diverse enough
- Solution: reranking algorithm
- Maximum marginal relevance (MMR)
  - a combined criterion that takes the similarity with the user profile and already selected items into account

$$MMR \stackrel{\text{def}}{=} Arg \max_{D_i \in R \setminus S} \left[\lambda (Sim_1(D_i, Q) - (1-\lambda) \max_{D_j \in S} Sim_2(D_i, D_j)) \right]$$

- Q = Query (Description of Document category)
- D = Set of documents related to Q
- S = Subset of documents in R already selected
- R\S = set of unselected documents in R
- λ = Constant in range [0–1], for diversification of results
- Sim = cosine similarity

GHENT UNIVERSITY

DELIVERABLES

– INDIVIDUAL !!!

- Source code
  - Zip file with all source files
  - Your name in all source files
    - __author__ =
    - @author
  - Filename: “Pract2_Lastname_Firstname_source.zip”
- Report
  - Pdf file with answers to the questions
  - Your name in pdf document
  - Filename: “Pract2_Lastname_Firstname_report.pdf”

QUESTIONS?

Ufora Discussions

Toon.DePessemier@ugent.be

Bruno.Willems@ugent.be

FACULTY OF ENGINEERING
AND ARCHITECTURE

Prof. Dr. Ir. Luc Martens & Prof. Dr. Ir. Toon De Pessemier

E Luc1.Martens@ugent.be
E Toon.DePessemier@ugent.be

www.ugent.be

GHENT
UNIVERSITY