Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies

Ákos Lévárdy

Odporúčacie systémy založené na AI

Bachelor thesis

September 23, 2024

Degree course: Informatics
Field of study: 9.2.1 Informatics
Place: FIIT STU, Bratislava
Supervisor: PaedDr. Pavol Baťalík

ANNOTATION

500 words

ANOTÁCIA

500 slov

DECLARATION OF OATH

I hereby declare upon my honour that I wrote this thesis single-handed with usage of quoted literature and based on my knowledge and professional supervision of my supervisor.

ACKNOWLEDGMENT

First and foremost, I would like to thank my supervisor for their invaluable guidance and support throughout the duration of this project.

Contents

List of Figures

List of Tables

List of Abbreviations

1 Introduction

As Internet and Web technologies continue to evolve rapidly, the amount of information available online has expanded excessively across sections such as e-commerce, e-government, and e-learning. To help users navigate this vast sea of content, Recommender Systems have become fundamental. They are very effective tools for filtering out the most appropriate information any user would like to find. The primary focus of these recommendations is to predict if a specific user will be interested in the distinct items.

The main target of this project is to create a recommendation system that uses ..... (text, materials).

2 Analysis

Making decisions is not always easy. People are frequently presented with an overwhelming number of options when picking a product, a movie, or a destination to travel to, and each option comes with different levels of information and trustworthiness.
The main purpose of recommendation systems is to predict useful items, select some of them and after comparing them, the system recommends the most accurate ones.
Because of this amount of detail from all of the items, recommendation systems are becoming increasingly important. They help reduce options and offer better suggestions for the user so that they will have a personalized list to select their favourite. Fast and efficient access to information is essential in any field of study. The Recommendation systems provide users with well chosen options for products that fit their requirements and interests which are mostly supplied by inputs [1], sometimes even matching their tastes.
When a user is trying to find a movie to watch, it would be hard for them to start searching without any starting options. After all a blank page and no suggestions to choose from might even make the user decide not to pick anything.

Different types of recommendation systems exist, and their methods of operation vary. These recommendation types are divided into 3 different categories, which are Content-Based Filtering approaches (CB), Collaborative Filtering approaches (CF) and Hybrid approaches which are the combinations of the two.

Content-Based Filtering works in a way that it creates user profiles and suggests the individual items or products based on the users past choices with similar items. The items have various features and characteristics which connect them. Collaborative Filtering relies more on preferences of other users and their behaviour. The point is that users who had similar interests before will have them again in the future for new items.

Both CB and CF approaches encounter significant challenges such as the Cold-Start Problem, Data Sparsity or Scalability. The Cold-Start Problem arises when making recommendations to new users and/or items for which the available information is limited. As a result, the recommendations offered in such cases tend to be of poor quality and lack usefulness. [2]

2.1 Collaborative Filtering

One of the most popular methods used for personalized recommendations is collaborative filtering. This method filters information from users, which means it compares users behaviour, interactions with items and data, item correlation and ratings from users.

It can perform in domains where there is not much content associated with items, or where the content is difficult for a computer to analyze - ideas, opinions etc. [3] Collaborative filtering can be divided into 2 methods which are "Memory-based" and "Model-Based" collaborative filtering. The first one relies on historical preferences, whereas the second method is based on machine learning models to predict the best options.

There are also 2 basic types of memory-based collaborative filtering which are:

- User-Based Collaborative Filtering
  - The main idea is that 2 completely distinct users who have an interest in a specific item and they rate this item similarly will probably be drawn to a new item the same way.
- Item-Based Collaborative Filtering
  - Calculates similarity between items, rather than users. The user will probably like a new item which is similar to another item they were interested in before.

It is important to mention that the effectiveness depends on the ratio of users and items. For example when trying

to recommend songs, there are usually way more users than songs and generally, many users listened to the same songs or same genres. Which means like-minded users are found easily and the recommendations will be effective. On the other hand, in a different field, when it comes to recommending books or articles the systems deals with millions of articles but a lot less users. This leads to less ratings on papers or no ratings at all, so it is harder to find like-minded individuals [4].

2.2 Content-Based Filtering

Recommender Systems which are using content-based filtering, review a variety of items, documents and their details. Each product has their own description which is collected to make a model for each item. The model of an item is so composed by a set of features representing its content. The main benefit of content-based recommendation methods is that they use obvious item features, making it easy to quickly explain why a particular item is being recommended. [5]
These profiles for items are different representations of information and users interest about the specific item.
The recommendation process basically consists in matching up the attributes of the user profile against the attributes of a content object. [5]

There can also be side information about items, where this side information predominantly contains additional knowledge about the recommendable items, e.g., in terms of their features, metadata, category assignments, relations to other items, user-provided tags and comments, or related textual content. [6]

The process for recommending items using content-based filtering has 3 different phases.

- Content Analyzer
- Profile Learner
- Filtering Component

The user modeling process has the goal to identify what are the users needs and this can be done 2 ways. Either the system calculates them from the interactions between the user and items or the user can specify these needs directly by giving keywords to the system, providing search queries [4].

2.3 Feedback

When trying to acquire feedback from the user there are 2 separate ways. The first one is the Explicit Feedback where it is necessary for the user to give item evaluation or actively rate products. Most popular options are gathering like/dislike ratings on items or the ratings can be on a scale either from 1 to 5 or 1 to 10. After the ratings the user can also give comments on separate items.

The other way is Implicit Feedback where the information is collected passively from analyzing the users activities. Some alternatives can be clicks on products, time spent on sites or even transaction history.

2.4 Hybrid approach

- uses both the CF and the CB filtering, for more accuracy

When trying to choose which recommendation approach is the best, first it is important to know the use case for the specific system. In the field of libraries, research paper and article recommendation, a study shows that more than half of the recommendation approaches applied content-based filtering [4].

2.5 Difficulties related to recommendation systems

- cold-start problem
- data sparsity
- scalability
- bias and diversity
- privacy
- serendipity
- over-specialization problem can occur - CBF
- ...

- Exploration VS Exploitation

2.6 Performance - measures - Metrics

- recall rate
- root mean square error
- precision
- cumulative gain ?
- accuracy
- overall efficacy
- f1 - measure
- Normalized Discounted Cumulative Gain (NDCG)
- ...

2.7 Semantics ?

2.8 Ontology ?

2.9 Matrix Factorization

Matrix factorization (MF) is a technique utilized in collaborative filtering to decompose a matrix of user-item ratings into lower-rank matrices capturing the latent factors under- lying the data [7].
People prefer to rate just a small percentage of items, therefore the user-item rating matrix, that tracks the ratings people assign to various items, is frequently sparse. In order to deal with this sparsity, matrix factorization (MF) algorithms split the matrix into two lower-rank matrices: one that shows the latent properties of the items and another that reflects the underlying user preferences. These latent representations can be used to predict future ratings or complete the matrix's missing ratings after factorization.

2.10 Search Engines

Search Engines have become crucial for navigating the vast amount of information available online. They make it possible for people to quickly look up solutions, learn new things, and browse the wide variety of resources available on the internet. Search engine optimization is now necessary to guarantee that search engines deliver relevant results, quick search times, and a top-notch user experience given the explosive growth of online information.
A search engine is essentially a software that finds the information the user needs using keywords or phrases. It delivers results rapidly, even with millions of websites available online. The importance of speed in on-

line searches is highlighted by how even minor delays in retrieval can negatively affect users' perception of result quality. [8]

2.11 Concept Drift ?

Concept Drift - Entropy-based

Information systems inevitably experience frequent data changes. This change in the statistical properties of the target variable, caused by unforeseeable variations in the underlying distribution of the data stream, is known as concept drift. [9]

POZNAMKY

Most content-based systems are conceived as text classifiers built from training sets including documents which are either positive or neg- ative examples of user interests Content-Based Book Recommending Using Learning for Text Categorization

THIS CAN PROVE THAT CF IS BAD FOR BOOKS [10]

Items that have not been rated by a sufficient number of users cannot be effectively recom- mended. Unfortunately, statistics on library use indicate that most books are utilized by very few patrons Therefore, collaborative approaches naturally tend to recommend popu- lar titles, perpetuating homogeneity in reading choices. Also, since significant information about other users is required to make recommendations, this approach raises concerns about privacy and access to proprietary customer data. Finally, al- though newly introduced items are frequently of particular interest to users, it is impossible for a collaborative approach to recommend items that no one has yet rated or purchased - analyzes relationships between users and interdependencies among products to identify new user-item associations

- recommend items or content to users by analyzing their interactions and similarities with other users

[11]

- long-tailed users - user groups with relatively uncommon or more diverse interests or preferences

- association rules

[12]

- user confidence
- time context

- personalized -vs- not personalized(recommend based on huge amount of people) recommendations
- recommend specific subject, study material, field of study, examples, lessons
- "At first it is important to describe / to outline / to define ..."
- Material difficulty level - features
- Comments, reviews, ratings, number of views, content, inquiries, and other factors can be used to automatically assign a difficulty level.

- four primary stages: student profiling, material collection, material filtering, and material validation.
- Machine Learning (ML), Decision Making (DM) approaches
- Each stage is explained in detail in the sub-sections below.
- The DM techniques were used to extract keywords from the material provided to develop queries

- MATERIALS: textbooks, lecture notes, additional questions, quizzes, exam samples, reports, articles, and books

I. The content-based module: The module is responsible for analyzing the con- tents of the materials and representing each material with a set of keywords and assigning them to topics and courses. II. The collaborative module: The module used the ratings, reviews, and num-

ber of views of the materials in the student's history. III. The contextual module: The module used the students' marks and level of performance. IV. The serendipity module: The module used the publicity of the materials and their reviews in the material database. [13]

CLASSIFICATION METHODS
- Text classification
automatically categorizing textual data

3 Specification of requirements

4 Implementation

4.1 Dataset

5 Conclusion

References

[1] Simon Philip, P.B. Shola, and Abari Ovye John. Application of content-based approach in research paper recommendation system for a digital library. *International Journal of Advanced Computer Science and Applications*, 5(10), 2014. doi:10.14569/IJACSA.2014.051006.

[2] Malak Al-Hassan, Bilal Abu-Salih, Esra'a Alshdaifat, Ahmad Aloqaily, and Ali Rodan. An improved fusion-based semantic similarity measure for effective collaborative filtering recommendations. 17(1), 2024. doi:10.1007/s44196-024-00429-4.

[3] Prem Melville, Raymond J. Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, pages 187–192, Edmonton, Alberta, 2002.

[4] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. Research-paper recommender systems: a literature survey. 17(4):305 – 338, 2016. doi:10.1007/s00799-015-0156-0.

[5] *Content-based Recommender Systems: State of the Art and Trends*, pages 73–105. 2010. doi:10.1007/978-0-387-85820-3_3.

[6] Pasquale Lops, Dietmar Jannach, Cataldo Musto, Toine Bogers, and Marijn Koolen. Trends in content-based recommendation: Preface to the special issue on recommender systems based on rich item descriptions. 29(2):239 – 249, 2019. doi:10.1007/s11257-019-09231-w.

[7] Srilatha Tokala, Murali Krishna Enduri, T. Jaya Lakshmi, and Hemlata Sharma. Community-based matrix factorization (cbmf) approach for enhancing quality of recommendations. 25(9), 2023. doi:10.3390/e25091360.

[8] Serge Stephane AMAN, Behou Gerard N'GUESSAN, Djama Djoman Alfred AGBO, and KONE Tiemoman. Search engine performance optimization: methods and techniques. 12, 2024. doi:10.12688/f1000research.140393.3.

[9] Yingying Sun, Jusheng Mi, and Chenxia Jin. Entropy-based concept drift detection in information systems. 290, 2024. doi:10.1016/j.knosys.2024.111596.

[10] Raymond J. Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. page 195 – 204, 2000. doi:10.1145/336597.336662.

[11] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. 42(8):30 – 37, 2009. doi:10.1109/MC.2009.263.

[12] Ke Yan. Optimizing an english text reading recommendation model by integrating collaborative filtering algorithm and fasttext classification method. 10(9), 2024. doi:10.1016/j.heliyon.2024.e30413.

[13] Tasnim M. A. Zayet, Maizatul Akmar Ismail, Sara H. S. Almadi, Jamallah Mohammed Hussein Zawia, and Azmawaty Mohamad Nor. What is needed to build a personalized recommender system for k-12 students' e-learning? recommendations for future systems and a conceptual framework. 28(6):7487 – 7508, 2023. doi:10.1007/s10639-022-11489-4.

[14] Mehrbakhsh Nilashi, Othman Ibrahim, and Karamollah Bagherifard. A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. 92:507 – 520, 2018. doi:10.1016/j.eswa.2017.09.058.

[15] Hongbo Wang, Yizhe Wang, and Yu Liu. A sequential recommendation model for balancing long- and short-term benefits. 17(1), 2024. doi:10.1007/s44196-024-00460-5.

[16] Shilpa S. Laddha and Pradip M. Jawandhiya. Semantic search engine. 10(21):1–6, 2017. doi:10.17485/ijst/2017/v10i23/115568.

[17] Dirk Lewandowski. Understanding search engines. page 1 – 296, 2023.

[18] T. R. Mahesh, V. Vinoth Kumar, and Se-Jung Lim. Uscotc: Improved collaborative filtering (cfl) recommendation methodology using user confidence, time context with impact factors for performance enhancement. 18(3):e0282904, 2023. doi:10.1371/journal.pone.0282904.

[19] Ali Taleb Mohammed Aymen and Saidi Imène. Scientific paper recommender systems: A review. 361 LNNS:896 – 906, 2022. doi:10.1007/978-3-030-92038-8_92.

[20] Akhil M. Nair, Oshin Benny, and Jossy George. Content based scientific article recommendation system using deep learning technique. 204 LNNS:965 – 977, 2021. doi:10.1007/978-981-16-1395-1_70.

[21] Cataldo Musto, Pierpaolo Basile, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Introducing linked open data in graph-based recommender systems. 53(2):405 – 435, 2017. doi:10.1016/j.ipm.2016.12.003.

[22] D. De Nart and C. Tasso. A personalized concept-driven recommender system for scientific libraries. volume 38, page 84 – 91, 2014. doi:10.1016/j.procs.2014.10.015.