BYOU: Book for You Recommender System

Yingtao Xu
Department of Electrical Engineering
Columbia University
yx2318@columbia.edu

***Abstract*—Book for You (BYOU), a book recommender system has been designed for the big data application in this project. This book recommender system consists of both front-end and back-end implementations. The core part for the recommender are four recommendation algorithms. Three of them are implemented by Mahout, that is item-based recommendation, user-based recommendation, and single value decomposition (SVD) recommendation. The rest one is implemented from scratch, which is a contend-based algorithm using TF-IDF.**

***Keywords*-Recommender System, Information Retrieval, TF-IDF, Content-based, Item-based**

I. Introduction

In the era of big data , there are many applications developed on demand by using helpful information. Books are always a good topic for design consideration since in this fast changing world, books are still a preferable method for many people to obtain information no matter on paper or on electronics. However, which book is more suitable sometimes consumes a huge amount of time for a reader who are around so much information (information overload)to decide. Therefore, the potential advantage of a book recommender system appears, that is, learning while saving time.

A recommender is a powerful tool. It uses user's information, for example, name, gender, ID etc. as the dataset. Then, different algorithms which can be content-based filtering algorithms, collaborative filtering algorithms, or hybrid filtering algorithms are applied to train the model and finally generate the model. Next time, when a user put his or her query into the system, such as the user ID, the generated model will provides recommendations. There are many areas

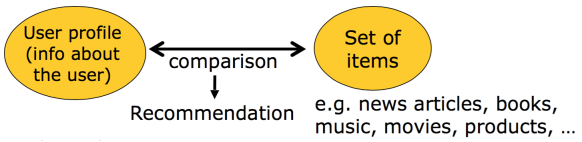

Figure 1. Recommender System

can deploy a recommender for application. For instance, in E-commerce, Amazon use a recommender system to do product recommendation for customers or in enterprise, a recommender system can be used to find domain expert. Or, like in this project, a recommender system can be used in digital libraries to do some paper or book recommendations.

The final delivery of this book recommender system is a fully functioning system. Namely, it has UI, server, and core recommender using four different algorithms

II. Related works

According to this comprehensive survey[1], different algorithms are chosen to be applied in the project. In particular, content-based is the core algorithm that this project is using. Its general idea is to use TF-IDF to generate feature vectors for each document and using cosine metric to calculate similarity (more details in Algorithm section). Also, University of Freiburg has collected and provided a book rating dataset [2] for public to use. This will be described in the following section as well. These two main components gives birth to BYOU - the Book for You Recommender System.

III. Software package description

A. Dependencies

The followings are main tools used in this project.

- Java - Java is used as the main programming language for the development of server, recommender.
- Mahout - Mahout, an open source library, is used to provide API and three of its algorithms to do the recommendation.
- Lucene - Lucene is an open source library which is used to remove stop words (like "the", "a" etc.) and do stemming.
- Jetty - Jetty is a light-weight library providing the server engine.
- Bootstrap - Bootstrap is a front-end template for building webpage.
- Maven - Maven is an application used for managing Java packages.

B. Notable Modules

Here is the list of the modules(classes) included in this project.

- UI - UI is the interface for user to interact with the recommender system and for recommender system to show the recommendation results for the user.

- Servlet - Servlet is used to load the designed jsp file and read the file to render webpage and also used to handle requests from webpage.
- Parser - Parser is designed to parse dataset by reading book information and removing stop words and stemming.
- Title Analyser & Item Profile Vectors Generators - This class is used to generate the feature vectors by using TF-IDF
- Master - Master is used to coordinate node by handling the request from the servlet and passing it to the node and by returning the recommendation results generated from node and passing it back to servlet.
- Node - Node is designed to use the API in Mahout to do item-based, user-based and SVD recommendation and also use the content-based recommendation designed manually to do the recommendation.

IV. System Overview

A. Dataset

The dataset used in this project is a public dataset called Book-Crossing Dataset [2] which is collected by Cal-Nicolas Ziegler in August and September 2004. There are three parts of the dataset including BX-Users which contains users' information like user ID, location, age etc., BX-Books which contains books' information such as ISBN, book title, book author, etc., and BX-Book-Ratings which contains the books' rating information.

The size of the dataset is around 160M, because of the memory limitation, only third of the data are used for this project. Also, not all three parts of the dataset are used. Actually, only BX-Book-Ratings part and BX-Books part are used. More specifically, only book titles are used in BX-Books part as this easy-to-write information can bloat the number of words.

B. Workflow

The process for the recommender system can be illustrated as Figure . Initially, user can click the webpage as a request to the server to generate a model for recommendation. Then, the server pass the request to the recommender. In the recommender, the parser will firstly parse data by removing stop words and stemming. After that, parser will give the parsed data to recommender master and master will coordinates with the node. Then, node will generate the model by using the recommendation algorithm as user specified and give it back to master. Next, master in the recommender will inform the server recommendation model is ready. Finally, when user input ID and click recommend request which, again, through server passed to the recommender, the node in the recommender can generate the recommendation and send it to server so that server can reflect the recommendation results to user on the webpage.

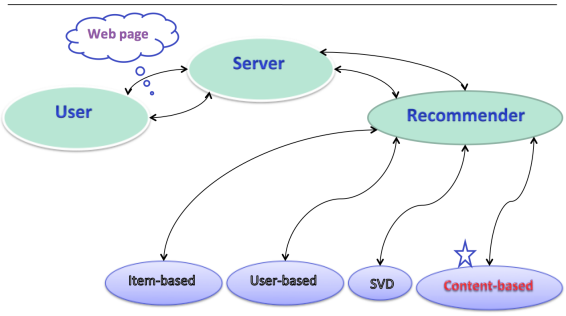

Figure 2. Workflow of the book recommender system

V. Algorithms

This section will give description about the core algorithms used in this project.

A. Item-based, User-based, and SVD recommendation

All three algorithms belong to collaborative filtering (CF) technique. For user-based and item-based filtering, they are memory-based type of CF whose approach is to use user's rating data to compute the similarity between users or items for recommendation. SVD, an abbreviation for singular value decomposition, is a model-based type of CF. It can explore deeper information of the data for recommendation by analysing the 'eigenvalues'(not exactly mathematical eigenvalues but similar idea).

All of the three algorithms are included in Mahout and by using the similarity in Mahout for each algorithm, recommendation can be done.

B. Content-based recommendation (TF-IDF)

A content-based recommendation uses the content, usually textual information such as texts, description etc., to do the recommendation. In this project, book title is the content chosen to use. TF-IDF, the abbreviation for term frequency-inverse document frequency, is the numerical statics reflecting how important an item is among a collection of items, for example how important a word is to a document.

In this project, TF-IDF are used to calculate the feature vectors for the books and after computing the cosine similarity of the feature vectors, Mahout can again to give recommendation.

In detail, term frequency (TF) vectors will be calculated first by counting how many times each word appear in each title. Next, document frequency vectors will be computed by counting how many titles contain each word and by log formula of DF, IDF vectors can be obtained. Then, multiplication of TF and IDF will give the TF-IDF vectors. After the feature vectors calculation, cosine similarity need

to be determined for the recommendation, which can be seen as the formula below (A and B are feature vectors):

$$Similarity = cos(\theta) = \frac{AB}{||A||||B||} \quad (1)$$

Then, Mahout can use the similarity and the feature vectors to do the content-based recommendation.

VI. Experiment Results

Here is a sample recommendation result by building the content-based(TF-IDF) model. The first table gives the top five recommendations for the user and the second table provides a list of user's reading history. More detailed results can be found in the demo.

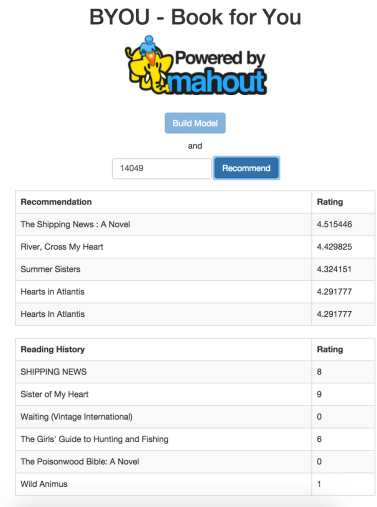

Figure 3. Result sample

VII. Conclusion and Future Work

This is a working project which can be used as a simple useful big data application. From front-end to back-end, the design of a feasible recommender system is very interesting. Also, while applying the algorithms in Mahout is a enhanced practice for the course, implementation of the content-based (TF-IDF) algorithm from scratch is a innovation and it is really fun.

For future work, evaluation of different algorithms used in this project is worth to be done so that the better algorithm for this book dataset and for the book recommendation can be obtained. Also, this project uses only the book title for recommendation, so in the future maybe more book information like genres, author, date etc. can be added into the project. Moreover, as mentioned previously, BX-Users part of this dataset has been discarded so in the future BX-Users information can be used to generate a more tailored model.

Acknowledgment

The author would like to thank Professor Lin for teaching many basic ideas about big data and recommender algorithms and hence inspire author with the idea to do this book recommender system. Also, the author would like to thank all the TAs' assistance in this course for many useful instructions for recommender system development.

APPENDIX

REFERENCES

[1] Adomavicius, Gediminas, and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." Knowledge and Data Engineering, IEEE Transactions on 17, no. 6 (2005): 734-749. Harvard

[2] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, Georg Lausen; *Proceedings of the 14th International World Wide Web Conference (WWW '05)*, May 10-14, 2005, Chiba, Japan. To appear.