Book Recommendation System that matches users based on genre preferences using Fuzzy Inference to calculate user preferences over a dataset

Evgeniya Evlogieva
Student @ Universiteit van Amsterdam
11389737

Philip Bouman
Student @ Universiteit van Amsterdam
10668667

***Abstract*—Nowadays, with the rapidly increasing number of e-books out there it is becoming easier to start reading a new book, it is only one click away. With the growing diversity, however, it is becoming harder to find a book that you will like. In this paper, we present a book recommendation system that is using a Fuzzy Logic System (FLS). We will use a dataset of books and user ratings for these books. The FLS will take as input the genre membership of the books and the user ratings, and will give as output to what extent a user likes a certain genre. For measuring the performance of the system, we will conduct a survey amongst a group of about 20 students with different backgrounds, and ask them how would they grade our solution. We are planing to further extend the proposed solution by adding a FLS for matching users, and also by clustering the set of genres and introducing fuzzy membership to a certain cluster.**

***Index Terms*—Fuzzy Logic, Recommender system, Book genre, Book recommendation**

I. Introduction

Fuzzy Logic is one of the pillars of Computational Intelligence and it deals with uncertainties in real world problems. It finds application in a lot of domains such as control [1], decision making [2], [3], image processing [4], etc.

In our project we investigate the application of Fuzzy Logic in the domain of decision making, and, more precisely, in a book recommendation system. There have been several attempts to solve this problem [5], [6], however, none of them using Fuzzy Logic. The aim of recommendation systems is recommending items that the user would like. Whether a user would like something or not, and to what extent, is a matter of personal preference and uncertainty.

The purpose of the project is to recommend a list of books (and an indication of the extent to which the user will like a certain book), based on an input in the form of a list of book genres the user has rated. The significance of the problem is not in its nature, but in the particular Fuzzy Logic approach, as it hasn't been applied yet to this particular area.

Our approach is to use an existing data set with user preferences and books. For the books for these datasets, we need to obtain their genres and to which extent a book belongs to a certain genre. This puts a certain limitation on the number of books we can use, because the 'scraping' of the genres is time-consuming for a rather big dataset.

Because of the above mentioned limitation we will use only a part of the data set (approximately 50 000 books). From it, we will use 80% as training data and 20% as test data.

The goal and our objective is to achieve at least 50% success rate. This means that on our test data, at least 50% of our "users" must actually have "liked" or "loved" any of the books our system recommends based on the part of their ratings we use as input.

Another measurement of success would be to have a survey and ask people to try the system. Considering the existing limitations, we will evaluate which approach will be more suitable for our case.

The rest of the paper is organised as follows. Section 2 gives a brief outline of the current approaches for finding user preferences. Section 3 presents in detail our approach of the problem. In the different subsections we discuss the data that we use for testing, the design and the implementation of the FLS. Section 4 and 5 explain what experiments we held and what results we obtained. In Section 6 we give our personal critical opinion on the results. Finally, concluding remarks and future plans for extending the project are included in Section 7.

II. Literature Review

Understanding user preferences is an important part of recommendation systems. Most approaches in obtaining preferences rely on explicit feedback from users such as ratings or lists of interest. However this kind of feedback tends to be affected by user inconsistencies (*natural noise*) [7].

Another way to get information about user preferences is using implicit elicitation methods, which can automatically learn preferences from item features, users demographic information and past behavior such as web and purchase history. Different implicit methods can be distinguished: collaborative filtering, content-based, and hybrid approaches [8].

Collaborative filtering methods approximate user preferences based on items rated by users, by finding similarities between users and recommend items that similar users liked but the target user had not come across yet. A downfall of the collaborative approach is that it does not take item features (genre, author) in to account and can be faced with a number of computational problems (scalability). Other limitations of collaborative filtering are that it cannot be applied to new items that have not been rated, and to unassigned users (*cold start/first rater problem, sparsity*) [9].

In a content-based approach user preferences are modelled (using machine learning) based on item features of user rated items. Automatically extracting item features can be hard for certain items, since it requires careful feature selection, representation, and inference. Content-based methods also cannot predict users new behavior like having new interest (limited by users history) [8], [9].

A hybrid approach tries to combine both implicit feedback methods, using both item content and user rating behavior. Since this approach tries to combine the strengths of the two other methods, it seemed the most promising for our model. By using the explicit rating for some books by a user and the implicit genre information for those particular books, a ”genre profile” can be created. This ”genre profile” will reflect the attitude of the user towards all genres. By comparing similarities between user's ”genre profiles” a recommendation for new titles can be made.

Fuzzy theory can be used to model the overlapping between genres within a book aswell as the uncertainty of the attitude of a user towards a particular (set of) genre(s).

III. Approach

A. Data

There are two sources of data used. The first one is an existing dataset from 2004 with data from bookcrossing.com[1] containing three different files: a list of user accounts, book information and user ratings for those books. The second source is a web crawl of Goodreads[2] consisting of a list of books and their genres as classified by users from Goodreads.

The first dataset, created by combining the user account and user rating files from the first source, will consist of: *user-ID*, *user age*, and a list of (*book:rating*)-tuples as rated by this user. The books will be referenced by their International Standard Book Number (ISBN) and the associated rating will be on a scale of 1 to 10. The original files contain information about 278858 user accounts and 1149779 rated books. The number of user accounts is reduced considerably since about two-thirds of the users did not rate any books and are therefore discarded from the final dataset.

An example of an entry (first dataset):

```
[user-ID, user age, (ISBN, rating), ... ]

[114, 57, ('0312953453', 7),
    ('0446608653', 9), ('0446612545', 9),
    ('0446612618', 8), ('0451208080', 8),
    ('0553584383', 9), ('0671027360', 10),
    ('0812575954', 5)]
```

The second dataset that will be used consists of genre information from Goodreads. By using the ISBN's from the book information data, a webcrawl was utilized to gather genre information about those books. As of now this set contains 33181 books and their genre's as classified by Goodreads users. The aim is to get genre information for approximately 50000 books. The genre information consists of a list of genre names (with at least one vote) and the number of times people rated it as such. Some pre-processing steps are applied, the most important one being the normalization of the number of votes. The final dataset will consist of an ISBN followed by a number of (*genre:grade*)-tuples.

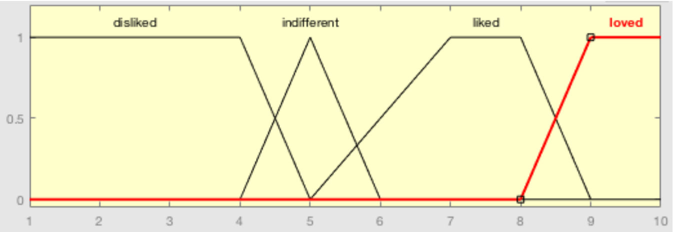


Fig. 1: Membership functions of user ratings.

An example of an entry (second dataset):

```
ISBN;[(genre, grade), ... ]

8445071408;[("'fantasy'", 1),
    ("'classics'", 0.48),
    ("'fiction'", 0.43),
    ("'adventure'", 0.1),
    ("'science-fiction-fantasy'", 0.08)]
```

The second dataset can be used as a lookup table for the entries in the first dataset. E.g.: find the appropriate genre information for a book rated by a user.

The output data of the system will consist of a list of ISBN recommendations in order of ”best-match”. The user will be shown this list along with accompanying book title and author.

B. Design

The Fuzzy Logic Inference System will have the following role: for each user, we will obtain a list of tuples of the type *(genre:grade)*, describing how does the given user like a certain book genre. The recommendation part will happen later, and it will be based on similarities between users, calculated with an error function.

Our system works with input consisting of sets of book ratings for each user. Items are considered ”loved”, ”liked”, ”indifferent” or ”disliked” based on the given rating. This is where fuzzification is applied. On Figure 1 you see the membership function of the ratings to each of the fuzzy sets ”loved”, ”liked”, ”indifferent” and ”disliked”. The range is from 1 to 10, because the ratings in the data set are in this interval.

For the ”disliked” fuzzy set we have a trapezoidal Membership Function (MF), starting from 1, and ending at 5. This decision is taken, based on our reasoning, that every rating from 1 to 4 you give, you more or less disliked the book.

[1] http://www2.informatik.uni-freiburg.de/ cziegler/BX/

[2] http://www.goodreads.com

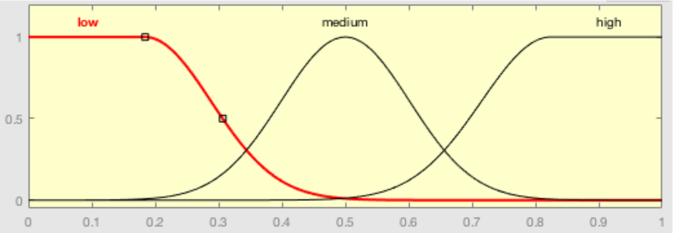


Fig. 2: Membership functions of genre membership.

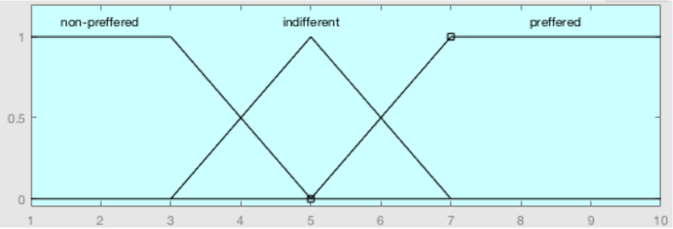


Fig. 3: Output - Membership functions of user preferences.

The ”indifferent” fuzzy set has a triangular MF, because the ”truly” indifferent grade for a book you could give is a 5. The function is also symmetrical, because any value that is equally far from a 5 has to have equal membership degree. For a rating of 7 and 8, we can definitely say that someone liked the book. That’s why the MF of the ”liked” fuzzy set is trapezoidal with a core [7, 8]. It starts from 5 so that we have more overlap between it and the ”indifferent” one, and it ends at 9. Lastly, the ”loved” MF is right shoulder MF, with a core [9, 10].

The second input of the system is, for every book, a list of tuples *(genre:membership_to_the_genre)*. The membership is calculated with a special normalization formula. Originally, the data set was containing how many people ’shelved’ a certain book as a certain genre. But when someone ’shelves’, or grades a book, he/she does not give just one genre. That way, if we want to normalize the data by summing up all the people to 100%, we may have people, who were counted twice or three times. Therefore, after some tuning of the parameters, the following solution was chosen: for a normalization coefficient was chosen the highest number of people, who rated a certain genre. That way, the highest rated genre will always have a membership degree of 1. To obtain the real membership of a book to a genre, the number of people who ’shelved’ is as that genre, is divided to the normalization coefficient. To justify the chosen algorithm, in the real world, the sum of the membership degrees of the different genres do not necessarily add up to 1. After calculating this membership of a book to a genre, it is fuzzified it to be estimated how much it belongs to the fuzzy sets ”low”, ”medium” and ”high” (Figure 2).

For these three fuzzy sets we decided we wanted the membership functions to be smoother, so we chose the Gaussian shaped ones. The ”low” and ”high” are symmetrical to one another. The ”medium” is with a core at the value 0.1. The range is between 0 and 1, because the normalized values that we have obtained are in this interval.

The next step in the FLS is the rules. For this step we decided to come up with the rules using expert knowledge.

The output fuzzy sets are ”preferred”, ”indifferent”, and ”non-preferred”(Figure 3). Much like the second input, the ”preferred”, and ”non-preferred” MFs are symmetrical, and the ”indifferent” is with a core at 5.

The crisp output of the FLS, is, for a certain user, a list of tuples *(genre:grade)*, that presents how much this user likes the certain genre.

*Note: In the future, a process of tuning the parameters will be performed, so the above mentioned MFs might change in shape or range.

For the actual recommendation step we will calculate an error function (the absolute distance vector or cosine similarity) between the target user and every other user that we have data for. The user with the lowest value of error will be the one we will get our recommendations from.

C. Implementation

For the implementation Python and MatLab will be used. The fuzzy inference part of the recommender system will be implemented in Matlab using the Fuzzy Toolbox. The final recommender system (figure 4) will be a Python script, taking user genre preferences as input and giving a list of recommendations as output.

A Matlab script will be used to setup and call the different parts of the fuzzy inference system within the Fuzzy Toolbox. With the Python **matlab.engine** package this Matlab process can then be controlled from Python to get the values for all the users and genres.

To collect and preprocess the data a number of python scripts are used aswell. First a script is used to extract the user account and user rating datasets from the Book-Crossing Dataset[3]*(Books Rated by User)*.

Second a web crawl was utilized to collect genre data (for rated books) from Goodreads[4]. The appropriate data is then extracted and normalized *(Genre Weights)*.

These two datasets are combined to form the input for the fuzzy inference part of the system. In this step a ”genre profile” is created for each user by repeatedly calling the MATLAB script to calculate the preference for each genre.

All the code can be found on https://github.com/phielp/Fuzzy

IV. Experiments

We conducted experiments with about 40 000 books and their genres. The initial FLS needed to be tuned, rules were changed, membersip functions were altered, fuzzy sets were added. The surface of the initial FLS was not smooth, and on certain places had a strange behaviour. It can be seen on Figure 5. After tuning the system, the surface became more smooth, and it fit better to the problem definition. The tuned FLS can be seen on Figure 6.

[3] http://www2.informatik.uni-freiburg.de/ cziegler/BX/

[4] http://www.goodreads.com

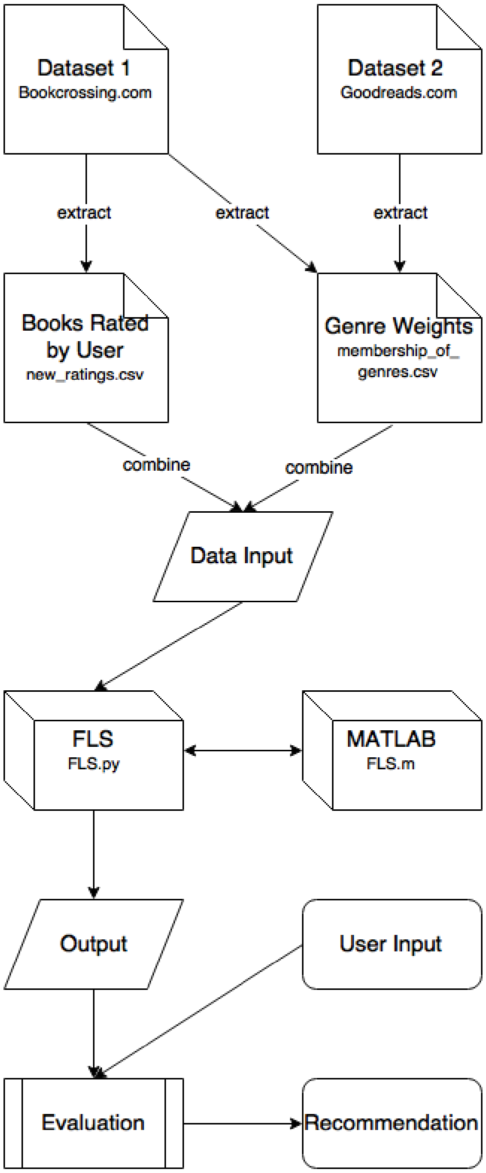
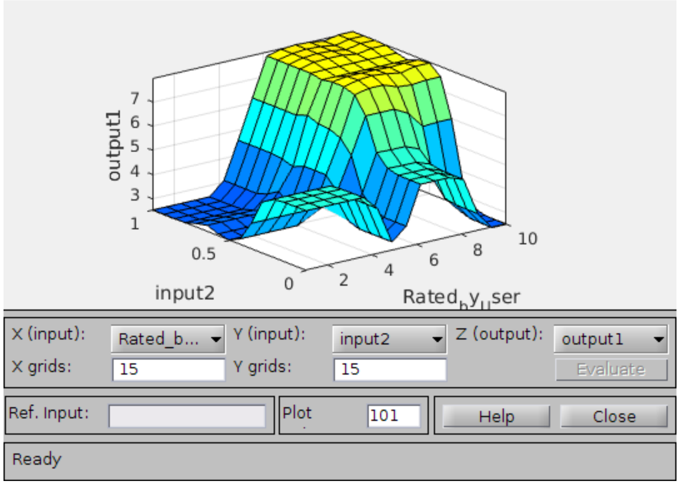


Fig. 4: Flowchart

After the system was tuned, we used it to extract user genre preferences for users in the dataset with size of approximately 45 000 users. It took in total 227 minutes (5 minutes per 1000 users). We used a python script to parse the data and call a Matlab script with the FLS, get the results from it and save them in a file.

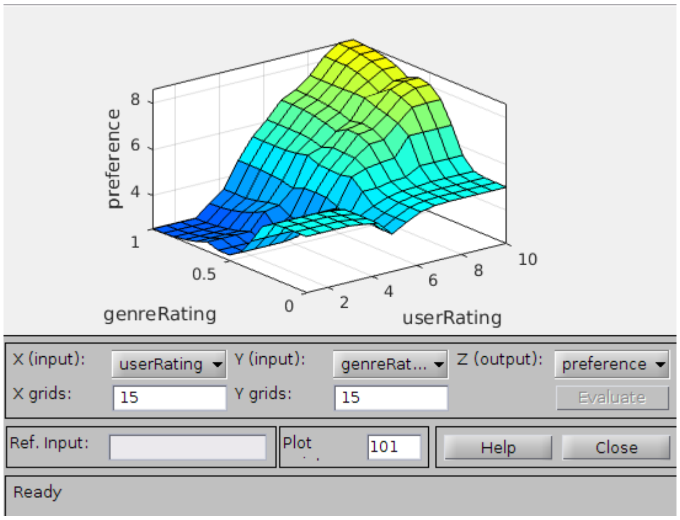
For now we have manually tested the system, and still have some tuning up to do on the evaluation and matching users methods.

Another experiment to measure the performance will be to get some of the other data (for now the plan is to use about 15 000 books), and based on some of the user ratings, to try to predict what the other user ratings are.



X (input): Rated_b... Y (input): input2 Z (output): output1
X grids: 15 Y grids: 15
Ref. Input: Plot 101
Ready

Fig. 5: Surface of the initial FLS



X (input): userRating Y (input): genreRat... Z (output): preference
X grids: 15 Y grids: 15
Ref. Input: Plot 101
Ready

Fig. 6: Surface of the initial FLS

V. Results

So far we have observed the issue of giving recommendations of books, based on only one common genre between the users. (it could also happen that for that genre our target user has given a very low grade) Without fixing the ’matching’ phase we cannot obtain feasable results with which to evaluate our performance and success rate.

Overall, evaluating the results from the FLS alone, looking at the raw input and the output of the system, it seems to produce desirable results. This is the format of the results with the collected user preferences:

```
user-ID;[(genre, rating), ... ]

255328;[("’science-fiction’", 7.5004),
    ("’thriller’", 7.3735),
    ("’fantasy’", 6.2548),
    ("’suspense’", 5.3333)]
255331;[("’fiction’", 7.8390),
("'medical'", 6.7421),
("'thriller'", 6.4035),
("'suspense'", 5.4170),
("'legal-thriller'", 5.2003)]
```

The first method of evalutaion will be using the eucledian distance between the user preferences in the training data and those in the test data.

Another method of evaluating the results will be based on clustering a user's genre preferences. For now k-means clustering will be used to classify the training data in a set of clusters (consisting of all the genres), with all users classified by their most preffered genre. A user can then be recommended any of the books rated high by users in the same cluster.

However, the final results will be clear after having tuned everything and having run experiments on the test data.

VI. Discussion

In related research the existing models are not applied to book reccomendation, but they are applied to different domains, such as movie- and music recommendation. Some related papers also address automatic genre classification or recommendation based on genre preferences [8], [10], [11]. For movie recommendation (in Zenebe et al.) [8] with a 3:1 split of the data the mean precision, mean recall, and mean F-measure were 62.41%, 56.52%, and 59.68%, respectively. For music classification (in Scaringella et al.) [11] a mean accuracy of around 70% is reported. So it seems reasonable to aim for a result between 60% and 70% accuracy.

VII. Conclusion

Appendix A
Some formulas here

Acknowledgment

References

[1] Chuen-Chien Lee. Fuzzy logic in control systems: fuzzy logic controller. ii. *IEEE Transactions on systems, man, and cybernetics*, 20(2):419–435, 1990.

[2] Hung-Tso Lin. Fuzzy application in service quality analysis: An empirical study. *Expert systems with Applications*, 37(1):517–526, 2010.

[3] Mohammad Hossein Fazel Zarandi, Neda Mohammadhasan, and Susan Bastani. A fuzzy rule-based expert system for evaluating intellectual capital. *Advances in Fuzzy Systems*, 2012:7, 2012.

[4] Gopala Sainarayanan, R Nagarajan, and Sazali Yaacob. Fuzzy image processing scheme for autonomous navigation of human blind. *Applied Soft Computing*, 7(1):257–264, 2007.

[5] Zan Huang, Wingyan Chung, Thian-Huat Ong, and Hsinchun Chen. A graph-based recommender system for digital library. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 65–73. ACM, 2002.

[6] Dong Kun. Research of personalized book recommender system of university library based on collaborative filter [j]. *New technology of library and information service*, 11:44–47, 2011.

[7] Denis Parra-Santander and Xavier Amatriain. Walk the talk: Analyzing the relation between implicit and explicit feedback for preference elicitation. 2011.

[8] Azene Zenebe, Lina Zhou, and Anthony F Norcio. User preferences discovery using fuzzy models. *Fuzzy Sets and Systems*, 161(23):3044–3063, 2010.

[9] Anthony F. Norcio Azene Zenebe. Fuzzy modeling for item recommender systems or a fuzzy theoretic method for recommender systems. 2008.

[10] Masaru Sugano, Roger Isaksson, Yasuyuki Nakajima, and Hiromasa Yanagihara. Shot genre classification using compressed audio-visual features. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages II–17. IEEE, 2003.

[11] Nicolas Scaringella, Giorgio Zoia, and Daniel Mlynek. Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine*, 23(2):133–141, 2006.