Matching users based on genre preferences, using fuzzy inference to calculate user preferences, for a book recommendation system

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[1] 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 conducted a survey amongst a group of 10 students with different backgrounds, and asked them how would they grade our solution. The majority rated the solution with a grade of 6 or higher. The other evaluation criteria based on the genre content of the recommended books, showed 73.5 % success rate. 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 'crawling' 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 35 000 books). From it, we will obtain a data set of about 45 000 users with their preferences. Two third of this set will be used as training data and one third as test data.

The goal and the objective is to achieve at least 50% success rate. This means that on the test data, at least 50% of the recommended books were appropriate. The criteria for appropriate is: the book either contains the genre that the user graded the highest, or the book contains at least one of the genres that the user graded higher than 5.

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].

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

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 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)]
```

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

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. This set contains 33181 books and their genre's as classified by Goodreads users. 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, such as the normalization of the number of votes (rating the most extant genre with a 1 and all subsequent relative to the first), discarding user-profiles for books without genre information and removing unrated books. The final dataset will consist of an ISBN followed by a number of (*genre:grade*)-tuples.

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.

After the Fuzzy inference step a intermediate dataset is created containing all calculated genre preferences per user. Based on the previous available user and genre information this set consists of 46647 entries.

An example of an entry (output fuzzy inference):

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

135300;[("'fantasy'", 3.642717355315818),
    ("'historical-fiction'", 3.642717355315818),
    ("'thriller'", 3.1652603053129242),
    ("'mystery'", 2.8175022851629996),
    ("'fiction'", 2.533228860250104)]
```

For the evaluation of the recommendation part of the system this dataset was divided in a training (31647) and a test (15000) set.

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

B. Design

The Fuzzy Logic Inference System has 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.

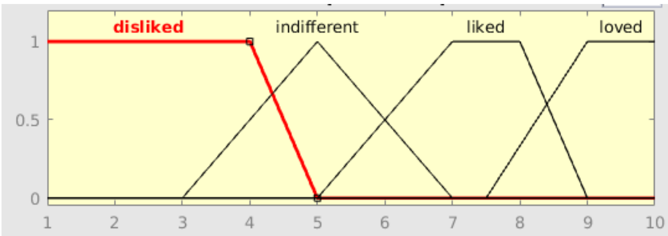

Fig. 1: Membership functions of user ratings.

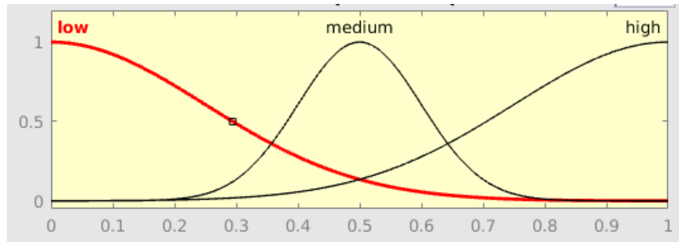

Fig. 2: Membership functions of genre membership.

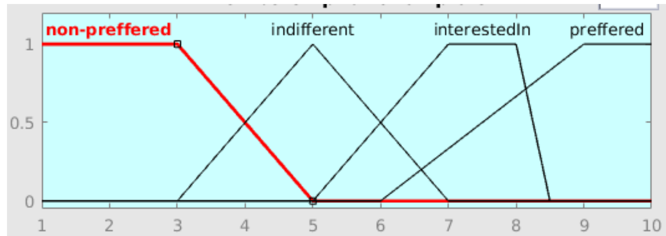

Fig. 3: Output - Membership functions of user preferences.

The 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. 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”, ”interested-in”, ”indifferent”, and ”non-preferred”(Figure 3). Much like the first input, the ”non-preferred”, and ”indifferent” MFs are left shoulder and symmetric triangular. The fuzzy set ”interested-in” is added to have a bigger overlap, and to behave accordingly to the fuzzy sets ”liked” and ”loved”. The ”preferred” MF is right-shoulder with a core at [9, 10].

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.

For the actual recommendation step we calculate an error function (the absolute distance vector) between the target user and every other user that we have data for. First, the intersection of their graded genres is calculated. For this intersection, if we have a match (a vector) that is as big as the target user input vector, and also, if the error is less than 2, a total error of -1 is given. In the case that two users are very close in preferences, the goal is to always choose this user for preferences. Another ’special’ case is when the match is only by one genre, but this genre is the one that the target user gave the highest grade for. Then, so that we make the matched user more preferable, we subtract 2 from the total error. Lastly, if there was no match in the genres, we give the highest error possible, which is a 9. The final value of the error is normalized by the number of matching genres.

The ten users with the lowest value of error will be the ones to get the recommendations from. From these ten users, we filter the books that they rated with a 10, and recommend them. Also, so that the list of recommended books is not too long, we recommend only the first 5 books to the user.

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 (**1**).

Second a web crawl was utilized to collect genre data (for rated books) from Goodreads. The appropriate data is then extracted and normalized (**2**).

These two datasets are combined to form the input for the fuzzy inference part of the system (**3**). In the fuzzy inference part a "genre profile" is created for each user (**4**) by repeatedly calling the MATLAB script to calculate the preference for each genre (**5**).

The output of the fuzzy inference step is stored (**6**) and can now be used for the evaluation of the system (separating the FLS output in a training and test set) (**7**) and ultimately the recommendation of books for new users (based on similar user profiles in the FLS output) (**8**).

All the code can be found on Github.[3]

IV. Experiments and Results

We conducted experiments with about 35 000 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 fitted better to the problem definition. The tuned FLS can be seen on Figure 6.

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.

The next step was manually testing the system, and doing some more tuning up, this time, on the recommendation part (calculating the error and matching users).

Another experiment to measure the performance that was conducted was getting one third of the user data (15 000 users) for test data.

The evaluation criteria that was first considered was taking half of the user's real rating, calculating genre preferences

[3]https://github.com/phielp/Fuzzy

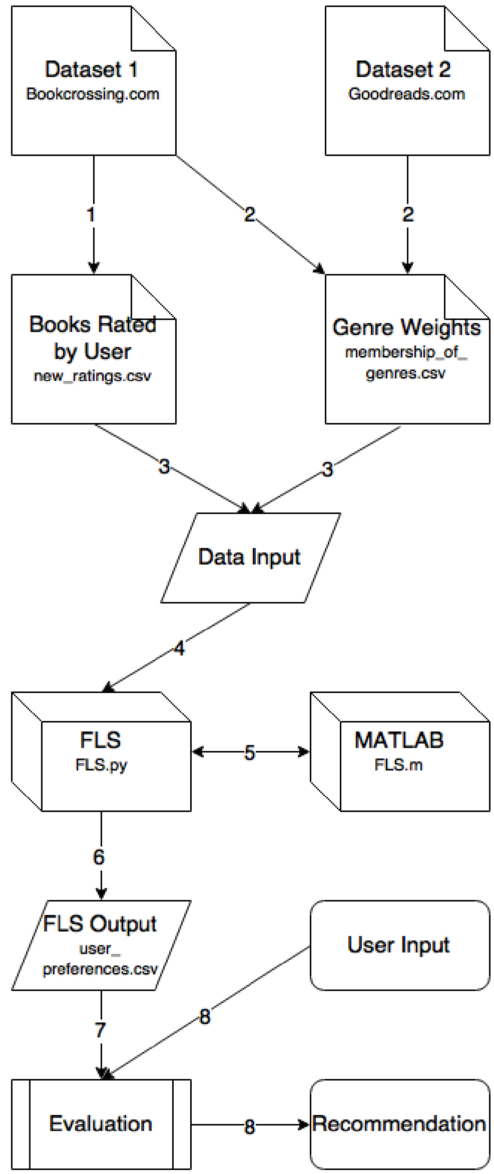

Fig. 4: Flowchart

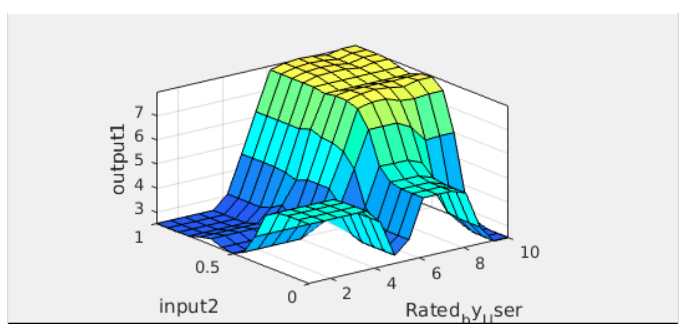

Fig. 5: Surface of the initial FLS

for them, and then check if the system recommended some of the other half of the user's high ratings. After trying this method and looking further into the data, the conclusion that

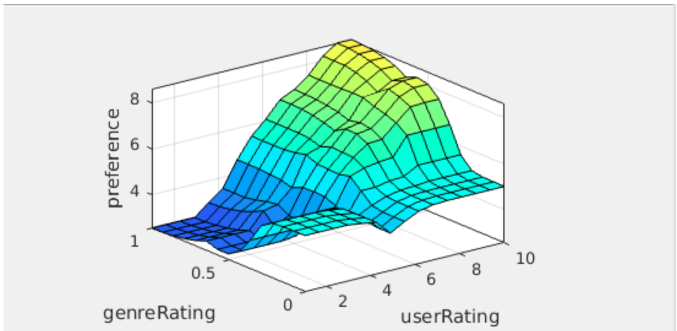

Fig. 6: Surface of the final FLS

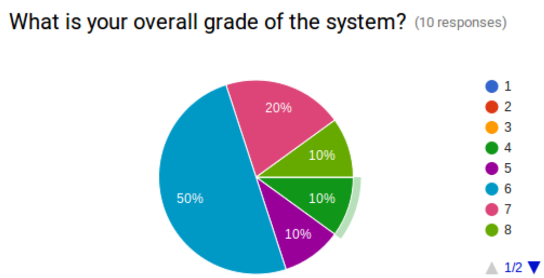

Fig. 7: Resulst for the overall grade

was drawn was, that there was not enough number of users that have rated a sufficient amounts of books.

Therefore the evaluation criteria for success was changed to the following: the top rated genre from the user appears in the book that we have recommended, or, one of the genres that the user rated higher than 5, appears in the book that we have recommended. With this evaluation criteria the system achieved success rate of 73.5%.

One of the experiments that was also considered was comparing the results of the recommendation, to the results obtained without the Fuzzy part of genre preferences. However, because of the time limitation and because of the different setup for this kind of measurement, this comparison could not happen. Nevertheless, this will definitely be one of the aspects of our future work on the matter.

The final experiment that was run (and one of the most valuable ones), was conducting a survey between 10 fellow students, who gave their opinions about the system. On Figure 7 you see the overall grade they gave.

The average grade is 6.1, and the majority (80%) have given a grade 6 or higher. The other question we asked in the survey, was how many of the recommended books do you think you will like. On Figure 8, there are the results of that question.

The majority of the people (80%) would consider reading at least one of the recommended five books.

Lastly, one more method of evaluating the results was based on clustering a user's genre preferences. We tried k-means clustering to classify the training data in a set of clusters (consisting of all the genres), with all users classified by their most preffered genre. The idea was that a target user

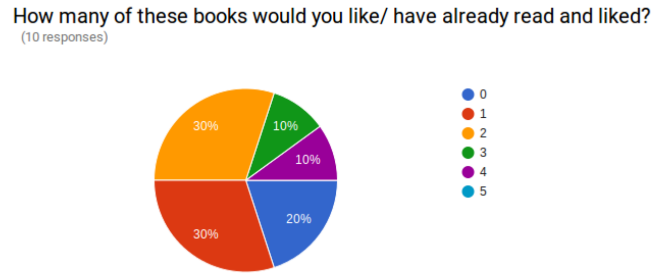

Fig. 8: How many books were "successfully" recommended

is recommended any of the books rated high by users in the same cluster. Because of the time limitation it was not possible to obtain any results in this approach.

V. Discussion

In related research (using fuzzy logic or fuzzy set theory) 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. The results we achieved seem to be in line with related approaches, confirming the initial idea that fuzzy logic is a useful tool in modeling user preferences. The approaches used in this paper might prove to be useful for approaches in different domains where genre classification can be applied.

VI. Conclusion

This paper shows an overview of how a book recommendation system can be created, making use of fuzzy logic to model uncertainties that arise when dealing with user preferences. The proposed approach is also more appropriate for the representation of (memberships of) features in items, then traditional approaches such as crisp or binary methods. Based on collaborative filtering for user ratings and expert knowledge about genre information the system can recommend books to new users to make discovering new books more accessible for readers.

Future work may include addressing limitations of this research and extending the propsed methods. First the number of genres used could be brought down to construct a more general model. Second the number of features of either the items or users could be extended to get a better distinction between them. Increasing the amount of data also is an important part as a recommendation can only be meaningful if the database consists of a representative amount of available choices. Finally the evaluation methods used can be improved or extended, for instance using clustering of genres or genre groups to match and classify similar users. Another direction would be to calculate user similarities by a Fuzzy Inference

System. The idea is the two inputs to be the user preferences and the user age. The Inference System will then calculate a crisp number for the similarity between the two users.

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.