Visualization of MovieLens Rating

Process Book

Zhao Chang u0931132@utah.edu u0931132
Junwei Shi u0943296@utah.edu u0943296
Tengda Shi tengda.shi@utah.edu u1015168

Project Repository: `https://github.com/VaultB0Y/dataviscourse-pr-visualization`

1 Overview and Motivation

The objective of our project is to perform an interactive visualization analysis on the relationship between different kinds of movies and people who give ratings on the movies. Our project deals with a really interesting problem. Note that thousands of movies are released over the world each year. We can easily find the corresponding information in some online web sites (eg. IMDB[1]). However, in fact, we are only interested in a small part of them. We have no time to scan all the introductions to these movies. Thus, we hope we can perform an interactive visualization analysis and tell people what movies they may like to see based on people's attribute information. These attribute information regarding different people can also be added into existing recommendation systems and improve the accuracy of recommendation.

2 Related Work

Existing works regarding recommendation systems inspire us to finish such a project. There are three major approaches used in prior works. The first one is collaborative filtering. It has been used in a lot of highly cited papers [1, 2]. It does not rely on profile information of users and items. Also, it does not need any knowledge engineering effort. However, it requires some form of rating feedback. Also, it cannot solve cold start for new users and new items [3]. The second one is content-based recommendation. It has already been used in some early works regarding recommendation algorithms [4, 5]. It only makes comparisons between items possible and does not require any community information. However, it needs some necessary content descriptions. Also,

[1] `http://www.imdb.com/`

it cannot solve cold start for new users. The third one is knowledge-based recommendation. It is also used in a few research works [6, 7]. It makes deterministic recommendations and ensures the quality of recommended items. Also, it can solve the cold start problem. However, it only works for static recommendations well and does not react to short-term trends. Recently, there are also some new approaches proposed by researchers. For example, some works improve traditional recommendation systems by adding social network analysis [8, 9]. Some works extract entities in users' tweets or weibos and add such information into existing frameworks for achieving advanced recommendation systems [10, 11].

However, all these works focus on how to recommend products for users accurately. These recommendation algorithms are usually performed based on some complicated mathematical deductions. However, it seems difficult for such algorithms to explain how they can generate reasonable results *in an intuitive way*. Therefore, we need to utilize the visualization tools. We perform an interactive visualization analysis and tell people what movies they may like to see based on peoples attribute information.

The visualization in this web site[2] inspires us to design Figure 12. Figure 12 can easily reveal the dependencies of users with different attributes and different kinds of movies.

3 Questions

The primary questions we are trying to answer with our visualization are:

- Tell people which movie they may like to see. If you plan to watch a movie but you are not sure if you will like this movie, then you can see the ratings of people who are in your ages through our visualization and decide whether to go to see the movie.
- For people in different ages, check what their favorite movies are and the ratings of each certain movie. The basic function of our visualization is to provide the services for users to search the information they want to check about the movies.
- Help the movie makers to investigate who like their movies best. Our visualization will also benefit the movie makers and help them to make movies that more people like to watch.

What we would like to learn is if any trends or patterns exist in our data set. In particular, the things we would like to accomplish include:

[2] `http://benfry.com/isometricblocks/`

- Show all the ratings of each movie, including the information of people who generate these ratings.
- Show different ratings of different movies generated by each person. People can search the information according to the audience's gender, occupation, postal zone and so on.
- Movies can be searched by category, theme, year and so on.

How do these questions evolve over the course of the project?

TBD

What new questions do you consider in the course of your analysis?

TBD

4 Data

We use MovieLens 1M Data Set[3]. This data set contains 1,000,209 ratings applied to 3,952 movies by 6,040 users of the online movie recommender service MovieLens. All ratings are contained in the file "ratings.dat" and are in the following format: UserID::MovieID::Rating::Timestamp. Ratings are made on a 5-star scale. Each user has at least 20 ratings. User information is in the file "users.dat" and is in the following format: UserID::Gender::Age::Occupation::Zip-code. Movie information is in the file "movies.dat" and is in the following format: MovieID::Title::Genres. By performing visualization analysis on the data set, we can answer some interesting questions, such as "what kinds of movies are liked by people with some specific attributes".

We pre-process the data set by clustering the data according to the different kinds of movies and different attributes of people. After generating such aggregation information, we use the transformed data set to perform interactive visualization analysis on it.

5 Exploratory Data Analysis

TBD

[3] `http://datahub.io/dataset/movielens`

6 Design Evolution

6.1 Our Design in Proposal

In our design, we divide the whole framework into five parts:

1) Present the rating information given a specific user.

2) Present the rating information given a specific movie.

3) Process the statistic given a specific user group using filtering.

4) Process the statistic given a specific movie group using filtering.

5) Find the dependencies of users with different attributes and different kinds of movies.

6.1.1 Part 1 Design

We can search the rating information given a specific user. Also we could filter out some types of movies we do not need, such as released year and genres (as shown in Figure 1). After that, we can present the information grouped by genre, or just a scatter plot along the year x-axis. (see Figure 2 and 3). Also we can rank the movies as shown in Figure 11.

6.1.2 Part 2 Design

We can search the rating information given a specific movie. We could present some interesting statistics, such as number of ratings or scores, categorized by ages. (Figure 5 and 6), or categorized by some interesting features such as locations. Please note that Figure 6 is a distribution figure, and we use different colors to denote the distribution percentage.

6.1.3 Part 3 Design

First we need to filter out the users we want, by selecting some features as in Figure 7. Then we show the statistics as Figure 5 and 6.

6.1.4 Part 4 Design

We can filter out the movies by selecting features as shown in Figure 4. We can show the statistics as shown in Figure 9 and Figure 10.

6.1.5 Part 5 Design

After some preprocessing of the data, such as clustering or collaborative filtering, we can show the dependencies of different groups of users and the movies. The groups are based on the features we have selected.

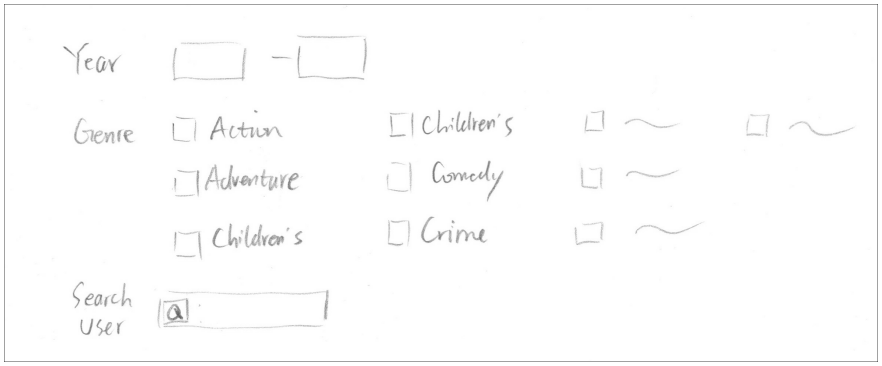

Figure 1: Search for user and filter the movie information

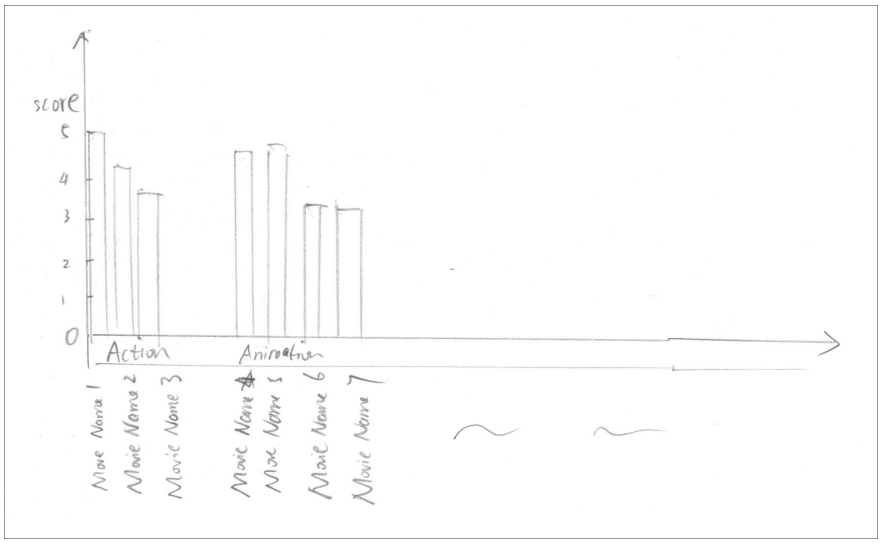

Figure 2: Histogram of ratings grouped by genre

6.1.6 Design ideas after the proposal

It has been mentioned in the project that we wanted to build a project to analyze the movie data. However, we have some good ideas after that. Why can't we make a more flexible visualization tool? It is very cool if we can import any data we want. (weather, population, business data...) Only we need to do is to select different attributes as row and col, then render the data into different kinds

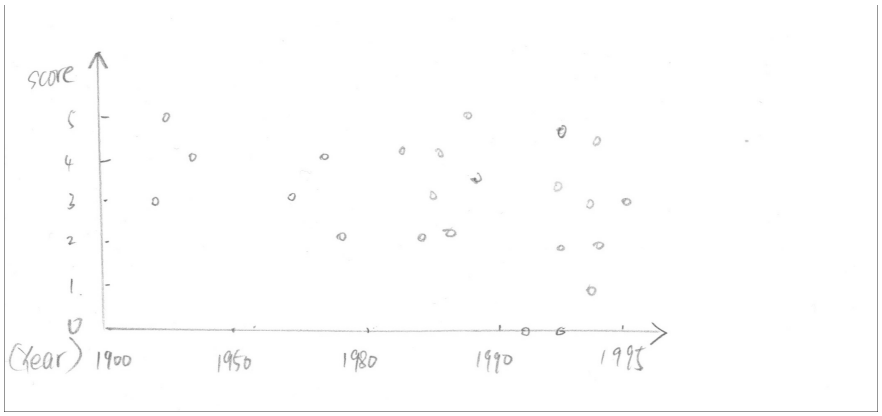

Figure 3: Scatter plot of ratings

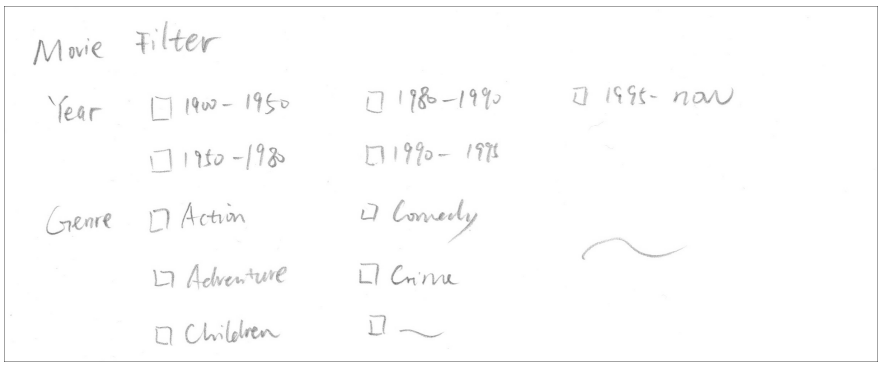

Figure 4: Filter of movie information

of chart. We can change the rows and cols for different views, or add more views in the webpage. We can filter out any feature and data we are interested. This very interesting, although it's much more complicated to build a flexible website than a website which focuses on only one dataset. We will try to implement this in the later process.

6.1.7 Project Peer Feedback

1) Its really a good idea of our visualization. It will give movie lovers lots of useful information.

2) Our dataset is a little big. Thats a problem how to deal with the big dataset. We may need use some machine learning algorithms to re-process our dataset. Whats more, its not a good idea to show the data in bar chart due to the big dataset.

3) Need to add more interactive innovation. We use lots of static picture to show the movie information. They suggest that we can create an interaction which to show the TOP 20 movies for different genders, ages and jobs.

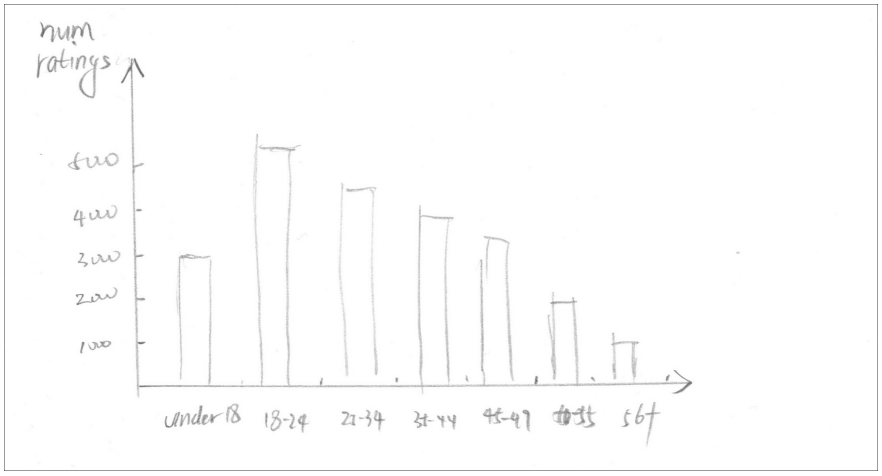

Figure 5: Number of ratings grouped by ages

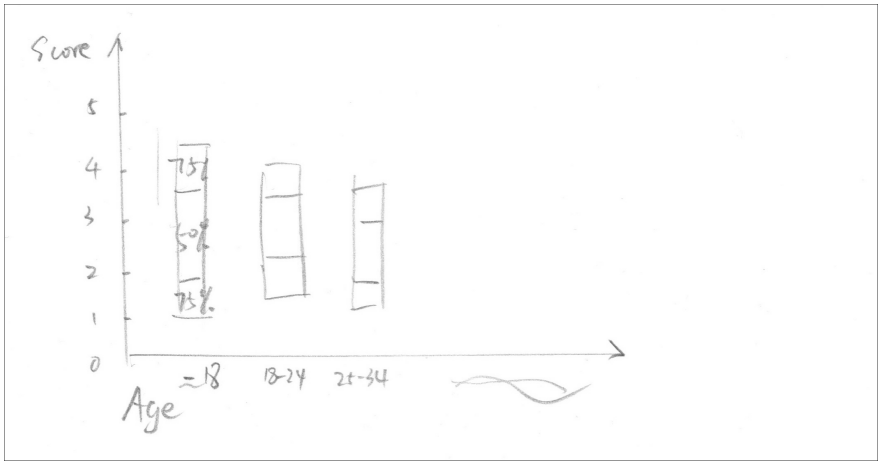

Figure 6: Distributions of ratings grouped by ages

4) Take use of the zip code information, so we can find the location information of each movie and show it on a map. This maybe more intuitive.

5) We dont have enough features in our dataset. If we can find a dataset which has more features of the movie, it will offer more useful information for the movie lovers.

6.2 Our Design in Final Submission

TBD

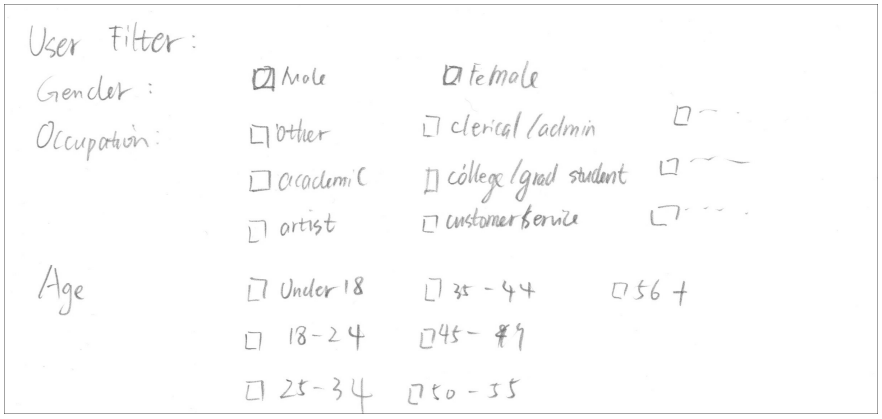

Figure 7: Filter of user information

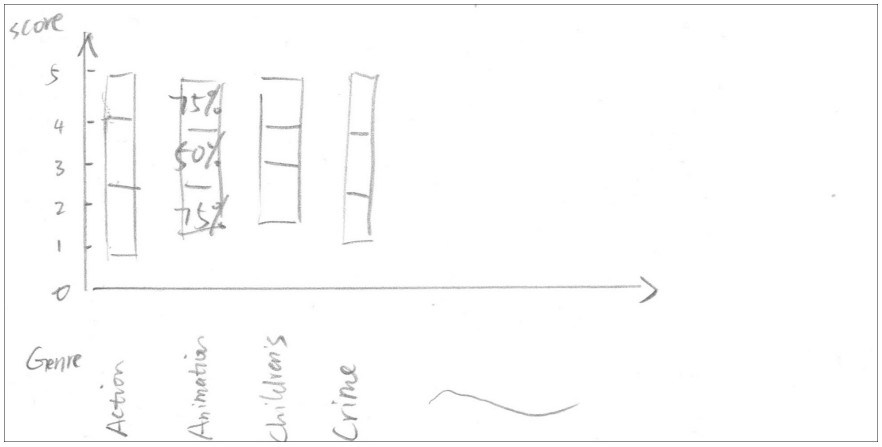

Figure 8: Distributions of scores grouped by genres

7 Implementation

7.1 Our Implementation in Milestone

We have implemented some basic static charts for visualization: bar chart, line charts, pie charts, and box plot in d3.

So far we have not combined the frontend and backend together. To build a prototype quickly, we just preprocess the dataset and generate some csv file we need for frontend use.

We need to add more HCI components in this project, such as drag and drop, brushing, data filter, changing row and column, etc.

Besides, we want to design the project in a beautiful way. We will need to separate the project into different layers and in a MVC framework.

Figure 9: Number of ratings grouped by genres

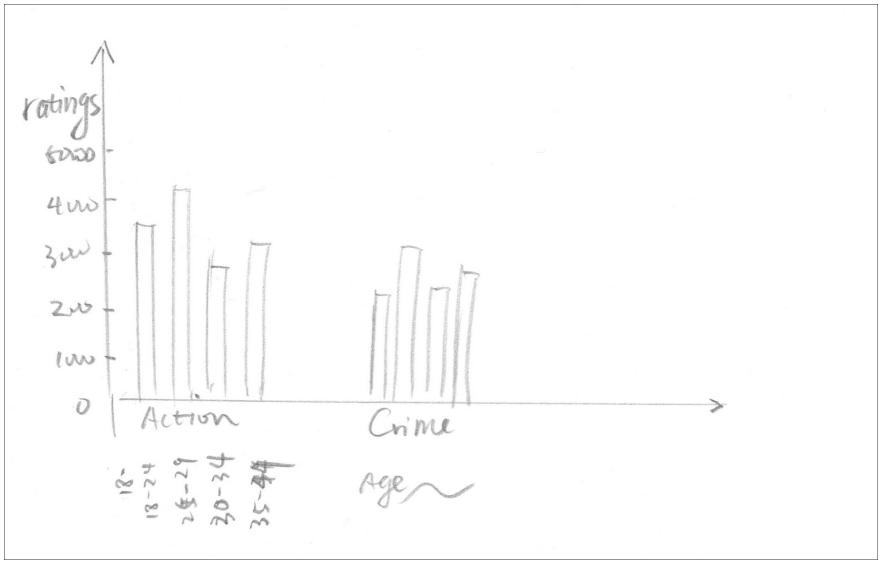

Figure 10: Number of ratings grouped by genres and subdivided by ages

7.1.1 Data Preprocessing

We simply transformed the original data into the format that can be easily processed. Then we imported the transformed data into MySQL. We used three tables to present the transformed data. The SQL statement "Create table if not exists Movie(MovieID int, Title varchar(100), Genres varchar(50))" can create Movie table. The SQL statement "Create table if not exists User(UserID int, Gender int, Age int, Occupation int, ZipCode varchar(15))" can create User table. The SQL statement "Create table if not exists Movie(MovieID int, Title varchar(100), Genres varchar(50))" can create Movie table. The SQL statement "Create table if not exists Rating(UserID int, MovieID int, Rating int, Timestamp varchar(15))" can create Rating table. After inserting the tuples into

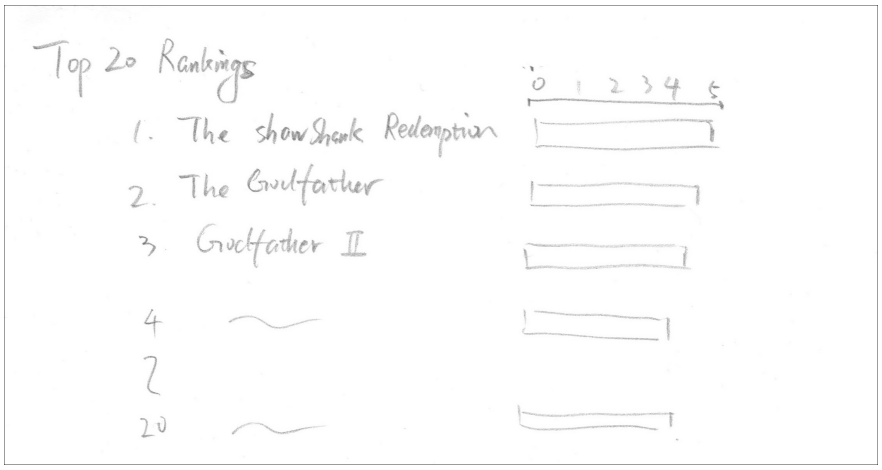

Figure 11: Rankings of movie scores

the tables in the database, we can easily gain some statistics by performing the corresponding aggregate queries.

7.1.2 Basic Visualization Design

Build the basic visual structure/layout using JavaScript.

7.2 Our Implementation in Final Submission

TBD

8 Evaluation

What do we learn about the data by using our visualizations?

TBD

How do we answer our questions?

TBD

How well does our visualization work, and how could we further improve it?

TBD

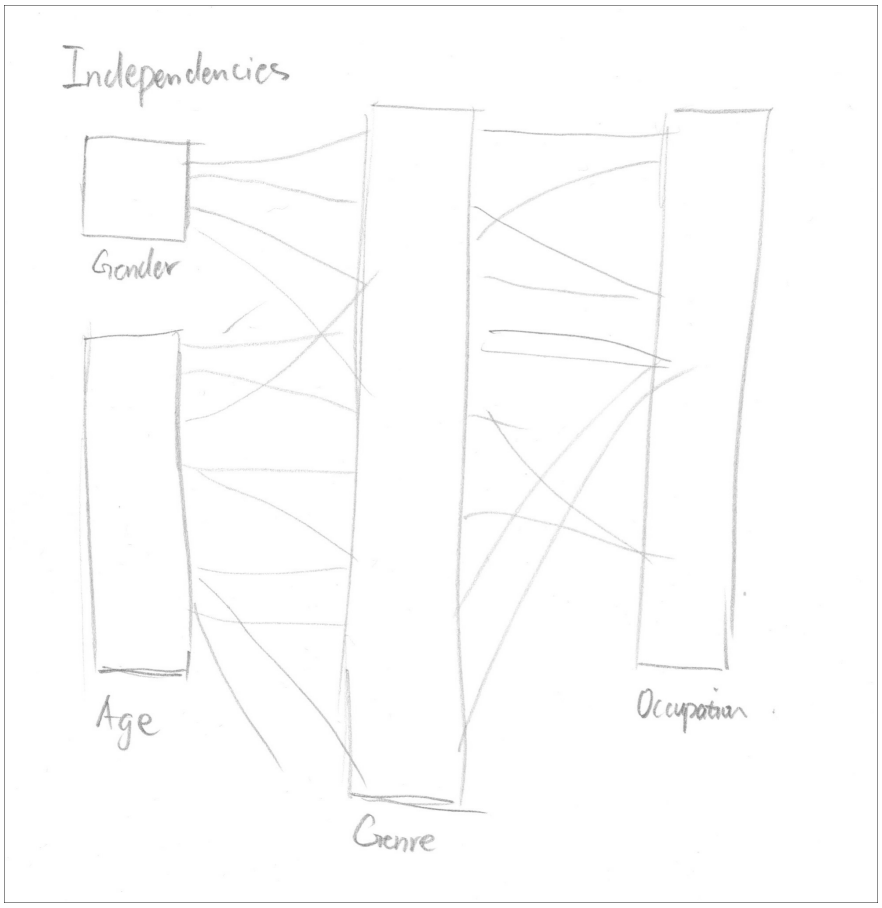

Figure 12: Dependencies of users and movies

References

[1] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285-295, 2001.

[2] G. Linden, B. Smith, and J. York. Amazon. com recommendations: Item-to-item collaborative filtering. In *Internet Computing, IEEE*, 7(1), pages 76-80, 2003.

[3] D. Jannach. Tutorial: Recommender systems. In *IJCAI*, 2013.

[4] C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *AAAI/IAAI*, pages 714-720, 1998.

[5] R. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195-204, 2000.

[6] R. Burke. Knowledge-based recommender systems. In *Encyclopedia of library and information systems* 69, Supplement 32, pages 175-186, 2000.

[7] B. Sigurbjörnsson and R. V. Zwol. Flickr tag recommendation based on collective knowledge. In *WWW*, pages 327-336, 2008.

[8] S. Debnath, N. Ganguly, and P. Mitra. Feature weighting in content based recommendation system using social network analysis. In *WWW*, pages 1041-1042, 2008.

[9] M. Pham, Y. Cao, and R. Klamma. A clustering approach for collaborative filtering recommendation using social network analysis. In *J. UCS*, 17(4), pages 583-604, 2011.

[10] O. Phelan, K. McCarthy, and B. Smyth. Using twitter to recommend real-time topical news. In *Recommender systems*, pages 385-388, 2009.

[11] O. Phelan, K. McCarthy, M. Bennett, and B. Smyth. Terms of a feather: Content-based news recommendation and discovery using twitter. In *Advances in Information Retrieval*, pages 448-459, Springer Berlin Heidelberg, 2011.