COMP 214 Project Report

Movie Recommender System

Cong Bao, Hongchi Chen, Jinke He, Xiang Li, Zekun Wang, Yuan Zhu
Artificial Intelligence Group Project
Team 1

1 TEAM INTRODUCTION

Name	ID	Mail	Role
Cong Bao	201218770	C.Bao2@student.liverpool.ac.uk	• Leader • System architect • Web server developer • Algorithm server developer • API designer • Database designer and developer • Designer and developer of schedule system • Designer and developer of index • Tester of schedule system
Hongchi Chen	201218808	H.Chen40@student.liverpool.ac.uk	• UI designer • Algorithm developer
Jinke He	201219022	J.He26@student.liverpool.ac.uk	• Collector of the data set of user ratings • Designer and developer of Recommendation Algorithm • Designer and developer of Movie Preview Page and Movie Search Page • Tester of algorithm and schedule system
Xiang Li	201219195	X.Li126@student.liverpool.ac.uk	• Server developer • System tester
Zekun Wang	201219686	Z.Wang120@student.liverpool.ac.uk	• Designer of menu style and setting pages • UI Designer of the manager page • Author of “About us” page
Yuan Zhu	201220136	Y.Zhu58@student.liverpool.ac.uk	• Designer of icon • Designer of UI pages • Stylist of System • UI developer • Contributor to server development

2 APPLICATION OVERVIEW

2.1 PROJECT DESCRIPTION

In contemporary society, data is playing a significant role in people's daily life. It is becoming a trend that people use data to help with decision making. Recommender system, which provides recommendations to users, is a typical case of data utilizing. Nevertheless, the algorithms used in common recommender systems are usually based on categories and tags, which require considerable costs in categorization and tag management. The Movie Recommender System (MRSYS), which is developed in our project, uses the Collaborative Filtering (CF) algorithm instead, to provide recommendations based on users' ratings. More specifically, rather than using an average value of ratings as reference, CF try to find users who share similar interests by analysing the previous ratings of a user. Based on the ratings of the similar users CF has found, the system will predict the ratings of a target user on the movies he/she has not watched yet. After that, a list of recommended movies will be generated according to the predicted ratings and provide to users. In addition, MRSYS provides a search function for user to find movies by key words. Favourite list is also provided for user to add movies they are interested in. All these functions provide user an easier and faster approach to search for a movie, get familiar with a movie, collect favourite movies, and receive recommendations according to their preferences.

2.2 TARGET USER

There are three kinds of customers that will make use of MRSYS.

- People who love watching movies but have no idea what kind of movies they may like.
- People who have interests in current most popular or top rated movies.
- People who would like to know basic information, including rating, of a specific movie.

3 REQUIREMENT SATISFACTION

3.1 ARCHIVED REQUIREMENTS

For non-registered user:

- Register for the system by entering account name, email and password.
- View the list of randomly recommended movies shown on the homepage of system.
  The system has achieved this requirement by accessing the movie database and displaying posters of movies on the homepage.
- View the brief introduction, rating and poster of a movie on the movie's introduction page.

For registered user:

- Sign in by entering the user account and password.
  Password verification has been developed to achieve this requirement.
- Sign out by clicking the "Sign out" button.
- Rate a movie by selecting number of stars shown on the movie's introduction page.
- View the recommended movie list on the profile page of a registered user.

The recommendation function is based on the Collaborative Filtering (CF) algorithm performed by the system every morning (6:00 GMT by default). According to the ratings made by the user, the system will predict the ratings of a user on the movies he or she has not seen yet, and then provide the user with the predicted results.

- Search a movie by entering the movie name into the search box.
  The search function has been achieved by using a movie searching API.
- Add a movie to the favourites list by clicking the heart label on the movie's introduction page.
- Remove a movie from the favourites list by clicking the heart label which has been clicked before on the movie's introduction page.
- Edit personal profile including changing email, changing password and deleting an account.

For administrator:

- Update the movie database by using the administrator page.
  The administrator can add a movie to the database as well as delete a movie from the database.
- Upgrade users' permission.
  The system will allow the administrator to upgrade a normal registered user to the administrator of the system.
- Change schedule time.
  The administrator can change the schedule time between website server and algorithm server, and has the option to fetch results immediately for testing or demonstration.

General requirements:

- An account with 10 or more days not conducting any rate will be set to inactive.
  This function is realized with a local-stored file which records user's activity.
- A user judgement should be conducted at 3 a.m. everyday.
- The website should provide registered users with new recommended movies lists each day if they are not judged inactive user.
- The website should NOT provide registered users with new recommended movies lists each day if they are judged inactive user.
- An inactive user can be re-acted by rate arbitrary movie.
- The website should be compatible with popular desktop browsers such as Chrome, Edge, Firefox, IE10+, Opera, Safari, etc.
- A friendly error page with help messages should be given if the website runs into problems.
- The time used to load each web page should not exceed 1 second in any condition.
  This requirement is realized with Asynchronous JavaScript and XML (AJAX) technique.
- The size of each picture should not exceed 1MB.
- The website should bear a peak of 1,000 page views.
- The website should be delivered with a user manual.
- The website should be delivered with a license file.
- The project is licensed under Apache License, Version 2.0.
- Using Java as the main server-side programming language.

Moreover, Python is used as algorithm developing language.

- Using JavaScript as the main web page programming language, and jQuery [1] as the main library.
- Using JSP as the dynamic page technique.
- Using Bootstrap 3 [2] as the web page UI framework.
- Using Spring 4 [3] and Apache Struts 2 [4] as the MVC framework.
- Using Hibernate 4 [5] as the ORM framework.
- Using Apache Tomcat 8.0 [6] as the web container.
  Moreover, Django 1.11 [7] is used as algorithm server's web container.
- Using MySQL 5.7 [8] as the database server.
- Using UTF-8 as the page encoding format.
- Using MD5 + SALT as the encryption standard.
- Using MQTT [9] as schedule protocol.
- All user data should be encrypted before writing into database.
- The system shall not disclose any personal information about customers apart from their username and e-mail address and those that are essential to generate recommendation lists.

3.2 ABANDONED REQUIREMENTS

- Function of the e-mail authentication

  The e-mail authentication is complex and it is also a less important function. The time used to realize the system is limited. We had to lay emphasis on algorithm part, which is the key of MRSYS. Therefore, this function has been abandoned.

- Social functions:

  Requirements of social functions for registered users including searching other registered users, viewing recommended friends, following registered user and unfollowing registered user have been abandoned due to the limited realization time. The realization of social functions would be very complex. These requirements can also be considered as the extension part of the system. They are also unnecessary functions for our recommender system. Therefore, these requirements have been abandoned.

4 EVALUATION

4.1 ADVANTAGES

One of the system's advantages is the movie recommendation algorithm. Different from the algorithms used in common recommender systems, which are usually based on categories and tags that require considerable costs in categorization and tag management, the algorithm used in our system is based on CF to provide recommendations according to users' ratings. More specifically, rather than using an average value of ratings as a reference, CF try to find users who share similar interests by analysing the previous ratings of a user. Based on the ratings of the similar users CF has found, the system will predict the ratings of a target user on the movies he/she has not seen yet and collect a list of movies with highest estimated ratings as a

recommendation. This algorithm is user-based, which means less requirements in the storage of user preference, such as tags and categories.

Another significant advantage of the project is the enormous number of movies that the system provides. All movies that are published after 2000 can be found using the searching function. Data including title, year, poster and brief description of all movies are retrieved from the Open Movie Database (OMDB) [10]. With the information provided in movie's preview page, user can learn more about this movie and check whether the movie is attractive for him/her.

Additionally, the system provides a friendly user interface for the user with fast response time. AJAX is used to realize the fast response in websites and asynchronous loading is achieved. It allows all web pages especially movies' preview pages respond to user`s request faster as the user does not need to wait for all elements in a web page to be loaded.

The system also provides a friendly interface for administrators to maintain the system. Manipulations on the database including adding a movie, deleting a movie and upgrading a user can be done through the console in user settings page. Furthermore, an administrator can update the execution time of recommendation algorithm. This console is only provided when the user signed in as administrator. Therefore, there is less potential that a normal user can access the server layer or the database layer of the system.

4.2 DRAWBACKS

The main weakness of MRSYS is in the recommendation algorithm. Since the algorithm used in our system is based on user ratings, a large amount of user's rating data is required. In our project, the data is based on an open source rating data library [11] which is large enough to be used to recommend movies to user. Nevertheless, without these existing rating data, this algorithm will not work. Therefore, this recommendation algorithm has some limitation in practical using that may not suitable in all cases.

Cold start issue is a general problem exists in recommender systems using collaborative filtering approach. In this movie recommendation system, a new user would get inaccurate recommendation list since the number of movies he/she has rated is small. It could lead to the loss of users because of the inaccurate recommendation, which may fall into a vicious circle. Additionally, new movies added to the database will require some time to be recommended to users since the lack of rating.

Another weakness is in the UI as there is an arrangement problem exists and the using of a third-party plugin [12] makes it hard to fix. The first picture on the home page for a user to browse always overlaps with the picture below it. This problem occurs every time the user enter the home page although when the alignment is re-arranged this problem will be fixed. This problem does not effect on major functionalities such as searching and movie recommendation.

There also exist some problems in search function. For example, if enter a key word which is very short, the web page may be stuck in loading status and take a long time to return results. A user who tries to search a movie with short key words will gain unpleasant experience due to this problem.

5 FUTURE WORKS

There are numerous potential development possibilities of MRSYS that can be implemented in the future.

- In current version of the program, user can discover a movie by entering related key words. The searching engine will then capture and present all movies which have corresponded name pattern. However, the consideration of different user groups is not very impeccable. On account of the entertainment property the program has, it will have more potential users from different race and age in future version. At present, the program is lack of a classification method and clearly system that can protect special group of users from restricted movie. For example, a twelve-year-old child should not be allowed to have authority to access pornographic movie.
  In order to achieve this feature, two attributes need to be added into the system. Firstly, during the register procedure, except basic information, user should also be asked to input their age and nationality. The additional information will then be submitted and stored in database. Moreover, for each movie, there should have a single attribute to support movie classification system. A movie with eroticism and violence will be marked. When a user processing a searching, specific movie will be banned and not shown to user. The related classification standard can reference to IMDB Movie Rating Website.
- Another point that can be altered is the recommendation method. In current version, except recommendation list, user can only be able to search movie by enter key words, which is lack of accuracy. In future works, a set of restriction conditions could be added through drop-down lists. Regarding movie type as an example, user could be provided several labels including “fiction” or “year before 1980” to submit, after that the corresponding movie list will be generated. The idea of the method is designed mainly for new registered user who would like to browse and be recommended specific type of movie but does not recall the accurate name of a single movie.
- The social user experience can be considered to expand. For now, an account can only proceed interaction with the system, including searching, rating, favourite and being offering recommendation list. In the next update, user-to-user function will be active. In this function, the system is expected to also update the backstage database structure to store more complex relationship between users. For the registered user group, except the existing rating function, they can also write comment under the movie page and reply other users’ view.
  Similar to the existing favourite movie function, registered user can also be able to favourite another user. After the favourite action, user can browse the recommendation list of the target user. If two users favourite each other, they become friends and can browse personal profile of each other and conduct private chat.
- The movie database should be updated at regular interval. New released movie should be included into database and therefore keep the timeliness of the system. Meanwhile, an old movie with zero number of rate or browsing record should be marked “unpopular”. After a period, system should delete unpopular movie. If there occurs a new comment or rate, the ‘unpopular’ label should be removed.

- The current movie database may not be comprehensive. If user would like to search a movie but could not find the pointed movie, he/she should be given a chance to request the movie.
  In order to achieve the function, a wish list textbox should be added to user profile. By submitting the accurate name of a movie which has not yet existed in movie database, user can request the website to include the movie. After a period, administrator will check the submitted list and add the movie to movie database if it has been requested for several times, and send notifications to the requesting users.
- The potential commercial value of website should be considered into future development. After a mutual version of website has been completed, the developers of program could properly accept some of sponsors from normal enterprise and build up cooperation relationship. The advertisement of corresponded enterprise could then be posted on a corner or below the website page in order to not disturb user experience. The advertisement will be evaluated first to make sure not to offence related group or the law.

6 BCS CODES OF PRACTICE AND CONDUCT SATISFACTION

According to files "Code of Conduct" and "Code of Practice" which outline the primary standard of information IT, during the specification, design, implement and demonstration link of the development of MRSYS Project, the most of practice and conduct disciplines are complied.

Public Interest

The project values and treats seriously the interest of the customer. The privacy of users are protected well by the program via password verification setting. The password entered by user is encrypted before stored into database to prevent unnecessary divulgence of privacy.

Most of original data comes from IMDB movie website, which is granted permission before, in order to obey the copyright of Third Party.

MRSYS does not have prejudice to any group of minorities. The user is not forced to give any real personal information about themselves to the maximum extent except email address, nor will the webpage and developer to judge user by the information. Every registered user has equal right on MRS. In addition, the database will be examined before presenting to user, in order to avoid the appearance of unhealthy content that has potential harm to user group.

Professional Competence and Integrity

During the entire development procedure, team members used multiple ways to improve their own computation and communication skills, consulting supervisor, trying to contact related department leader and searching for online sources. Every piece of work only comes from the effort the group member paid, without filching any unauthorized resource, such as movie database source and web structure. In current stage, the webpage does not accept commercial purpose, and does not have any intension to accept illegal cooperation in the future, which complies with principles of "BCS Code of Conduct".

Duty to Relevant Authority

For the project, there is no specific relevant authority group. However, the final demonstration of finished webpage is critically follow the design phase of the project. Moreover, if regarding potential and existing user as relevant authority, the design and implement of webpage is friendly and clearly, with idea of easy for user to get familiar of. There are individual user manual and notice for user from different scopes to understand functions of webpage and how to access them. There is no intension to deceive and make profit from user. If user chooses to delete account, the action will be realized immediately and thoroughly and all copies of user information will be destroyed. The related features are corresponded to principles of "BCS Code of Conduct" to not neither disclose nor take advantage of the relevant authority.

Non-complied Discipline

However, due to the restriction of resource and copyright in related area, the developer may not be able to contact and keep in touch with the top developers and specialists of movie or computer domain. In addition, the website is programmed and designed only by university year two students by advices of university supervisor. Some of resources and development skills are found on Internet without authoritative references, therefore may have plenty of drawbacks and insufficiencies. As a result, according to "Code of Practice", the design and implement do not comply to discipline "Act Professionally as a Specialist" of "Code of Practice".

7 BIBLIOGRAPHY

[1] "jQuery," [Online]. Available: https://jquery.com/.

[2] "Bootstrap - The world's most popular mobile-first and responsive front-end framework.," [Online]. Available: http://getbootstrap.com/.

[3] "spring," [Online]. Available: https://spring.io/.

[4] "Apache Struts," [Online]. Available: https://struts.apache.org/.

[5] "Hibernate," [Online]. Available: http://hibernate.org/.

[6] "Apache Tomcat," [Online]. Available: http://tomcat.apache.org/.

[7] "Django," [Online]. Available: https://www.djangoproject.com/.

[8] "MySQL," [Online]. Available: https://www.mysql.com/.

[9] "MQTT," [Online]. Available: http://mqtt.org/.

[10] "OMDb API," [Online]. Available: http://www.omdbapi.com/.

[11] "MovieLens," [Online]. Available: https://movielens.org/.

[12] "Masonry," [Online]. Available: https://masonry.desandro.com/.

[13] Balabanović, M. and Shoham, Y., 1997. Fab: content-based, collaborative recommendation. *Communications of the ACM*, *40*(3), pp.66-72.

[14] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," Adv. in Artif. Intell., vol. 2009, pp. 2-2, 200

[15] J. Linwood and D. Minter, Beginning Hibernate, Apress, 2010, p. 118.

[16] L. Seinturier and R. Pawlak, Foundations of AOP for J2EE Development, Apress, 2006, p. 100.

[17] D. Rubio, J. Long, G. Mak and M. Deinum, Spring Recipes: A Problem-Solution Approach, Apress, 2014, p. 48.

[18] "Apache," [Online]. Available: https://httpd.apache.org/.

[19] "Nginx," [Online]. Available: https://nginx.org/en/.

[20] "NumPy," [Online]. Available: http://www.numpy.org/.

[21] "Git --fast-version-control," [Online]. Available: https://git-scm.com/.

[22] "PuTTY," [Online]. Available: http://www.putty.org/.

[23] "ubuntu," [Online]. Available: https://www.ubuntu.com/.

[24] "DigitalOcean," [Online]. Available: https://www.digitalocean.com.

[25] "Alibaba Cloud," [Online]. Available: https://intl.aliyun.com/.