A Movie Rank Prediction

Based on Data Mining Approach

Department of Computer Science

University of Victoria

Huzhen Chen (V00864674)
Yubing Ding (V00801106)
Chengxiang Xiong (V00838781)
Zhenyu Zhang (V00793557)

Abstract

The Internet Movie Database (abbreviated **IMDb**) [1] is an online database of information related to films, television programs and video games, including cast, production crew, fictional characters, biographies, plot summaries, trivia and reviews, operated by IMDb.com, Inc., a subsidiary of Amazon. With massive real-time changing data, IMDb has become one of the most relevant website people would choose for basic information of a specific movie. Our project is aimed to make a prediction of a movie's rank based on its directors, genres and actors. Through deeper analyzation, we present the influence of each attributes and predict the movie rank range (the rank is divided into two parts: 0 to 7 and 7 to 10).

Content

1. Introduction

During the last century from now on, movies have gone from a curiosity to one of the most popular form of entertainment for children and adults alike. There are thousands of movies launched every year. Some of the movies are worth to watching. We wonder if we could know whether the movie worthwhile before we watch it by analyzing the following questions:

1. Who directed this movie?
2. Who acted in the movie?
3. What genres the movie is?

When view the rank of a movie as the standard to see if it is an outstanding movie, we expect to come out a formula which may predict the rank of a movie with its director, actors and genres.

When analyzing the data, we found that there are too many subjective factors may influence the rank of a movie, such as the cast, a famous writer of the original text, pre movie or a range of works and the special effects of the movie. In such reason, we have to filter the data carefully though software and query.

The database of IMDB comes from CSC 370 course resource in SQL form. Director, actor and genre has been selected as exploring elements. PostgreSQL is the software which been used to filter, extract and list relative data. After this process, we transfer the output into a CSV file and import into WEKA.

2. Data Preprocessing

2.1 Database Extracting

We got the raw database from Professor Daniel German. It is a dumped PostgreSQL file. Through the command line below (Figure 1), the database can be extracted to a personal laptop (‘-d’ is followed by the name of database, ‘-h’ represents the hostname, ‘-U’ is the username and ‘-f’ means the address of the Psql file).

```
C:\Program Files\PostgreSQL\9.5\bin>psql -d imdb -h localhost -U postgres -f C:\
Users\Adrianna\Desktop\imdb.sql
SET
SET
SET
SET
SET
SET
SET
```

Figure 1: Command line for extracting the dump IMDb

The whole database has 13 tables (Figure 2), records the products' information and evaluations of films', television programs' and video games' data from 1874 till 2016.

```
imdb=# \d
              List of relations
 Schema |    Name     | Type  |  Owner
--------+-------------+-------+----------
 public | aka         | table | postgres
 public | color       | table | postgres
 public | countries   | table | postgres
 public | directors   | table | postgres
 public | episodes    | table | postgres
 public | genres      | table | postgres
 public | languages   | table | postgres
 public | links       | table | postgres
 public | locations   | table | postgres
 public | persons     | table | postgres
 public | productions | table | postgres
 public | ratings     | table | postgres
 public | roles       | table | postgres
(13 rows)
```

Figure 2: Tables in the IMDb

2.2 Data Filtering

Our theme is to find out the relationship between genre, director, actor and rating. The table we need are Productions, Directors, Genres and ratings, the primary key is id (Figure 3).

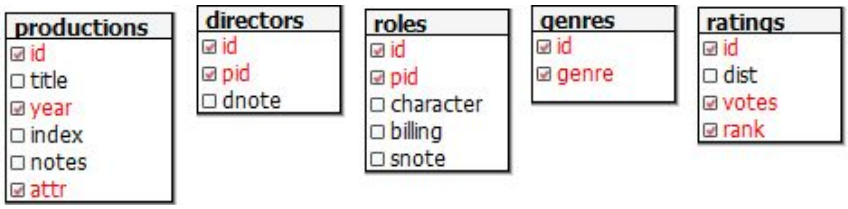

Figure 3: Tables we need

Additional, there are not only movies but also TV drama programs and video games in the IMDB, so we fetch all the movie by select tuples where 'attr is null'. Because of the limitation of Excel, we have to keep our data in a limited size. We made some tests to narrow down the movie data we select.

In order to make the result more suitable for the current situation, the year has been limited into the recent 5 years (2012 up to now) and actor has to act 15 to 24 times. Moreover, if the number of votes is rare, it cannot represent the general opinion, so we set the minimum number of votes to 50000.

The query is as follow:

select id, directors.pid, actor_name, genre, rank from (select * from productions where attr is null and year >= 2012) as a natural join directors natural join (select * from ratings where votes >= 50000) as b natural join (select pid as actor_name, id from roles) as c natural join genres;

For majority of the movie, actors, genres and even directors are not unique, it is impossible to use WEKA directly to compute. We solved this problem by making directors, actors and genres into clusters based on their average rank, which will be introduced in the Data Mining part.

Query for genres' average:

with T1 as (select id, dir_name, pid, rank, genre from (select * from productions where attr is null and year >= 2012) as a natural join(select id, pid as dir_name from directors) as d natural join (select * from ratings where votes >= 50000) as b natural join (select pid, id from roles) as c natural join genres), T2 as (select genre, avg(rank) as avg from genres natural join T1 group by genre) select id, dir_name, pid, genre, rank, avg from T1 natural join T2;

Query for directors' average:

with T1 as (select id, pid , actor_name, rank, genre from (select * from productions where attr is null and year >= 2012) as a natural join directors natural join (select * from ratings where votes >= 50000) as b natural join (select pid as actor_name, id from roles) as c natural join genres), T2 as (select pid, avg(rank) as avg from directors natural join T1 group by pid) select id, pid, actor_name, genre, rank, avg from T1 natural join T2;

Query for actors' average:

with T1 as (select id, dir_name, pid, rank, genre from (select * from productions where attr is null and year >= 2012) as a natural join(select id, pid as dir_name from directors) as d natural join (select * from ratings where votes >= 50000) as b natural join (select pid, id from roles) as c natural join genres), T2 as (select pid, avg(rank) as avg from roles natural join T1 group by

pid) select id, dir_name, pid, genre, rank, avg from T1 natural join T2;

After filtering the raw database, we got 144640 entries of movies. Each tuple contains only one id, genre, actor, director and rank, followed by this id's rank, genre's average rank, actor's average rank and director's average rank.

2.3 Data Exporting

For WEKA can accept files in CSV form, the filtered data is exported in csv format. There's the query we used:

Copy 'Query that Need to be Exported' to 'Destination + FileName.csv' with csv header;

For some unknown reason that WEKA cannot read quotation marks, so we replace all of them by space.

3. Data Mining

WEKA cannot generate random data, we have to make data into clusters. We each genres' average, directors' average and actors' average by using query below:

With T as (select genre, avg(rank) as avg from genres natural join ratings natural join productions where attr is null and votes >= 50000 group by genre) select genres, avg from T1;

With T1 as(select pid, avg(rank) as avg from directions natural join ratings natural join productions where attr is null and votes >= 50000 group by pid) select pid, avg from T1;

With T1 as(select pid, avg(rank) as avg, count(pid) as count from roles natural join ratings natural join productions where attr is null and votes >= 50000 group by pid) select pid, avg from T1 where count >= 17 and count <= 24;

Through WEKA, we separate directors into eight clusters, actors into six clusters and genres into five clusters. IMDb uses Bayesian posterior mean to calculate the rank of top 250 films (Figure 4). We choose 7.0 to separate the rank into two parts, rank<7.0 considered as bad movies, rank>=7.0 as good movies.

$$W = \frac{Rv + Cm}{v + m}$$

where:

W = weighted rating

R = average for the movie as a number from 1 to 10 (mean) = (Rating)

v = number of votes for the movie = (votes)

m = minimum votes required to be listed in the Top 250 (currently 25,000)

C = the mean vote across the whole report (currently 7.0)

The W in this formula is equivalent to a Bayesian posterior mean (See Bayesian statistics).

Figure 4: IMDb rating [1]

After we input the clustered data into data, we get following result.

Algorithm	**Accuracy**
IBk	71.1934 %
Classification Via Regression	70.3704 %
Logistic	69.5473 %
J48	69.1358 %

Table 1: Results Through WEKA

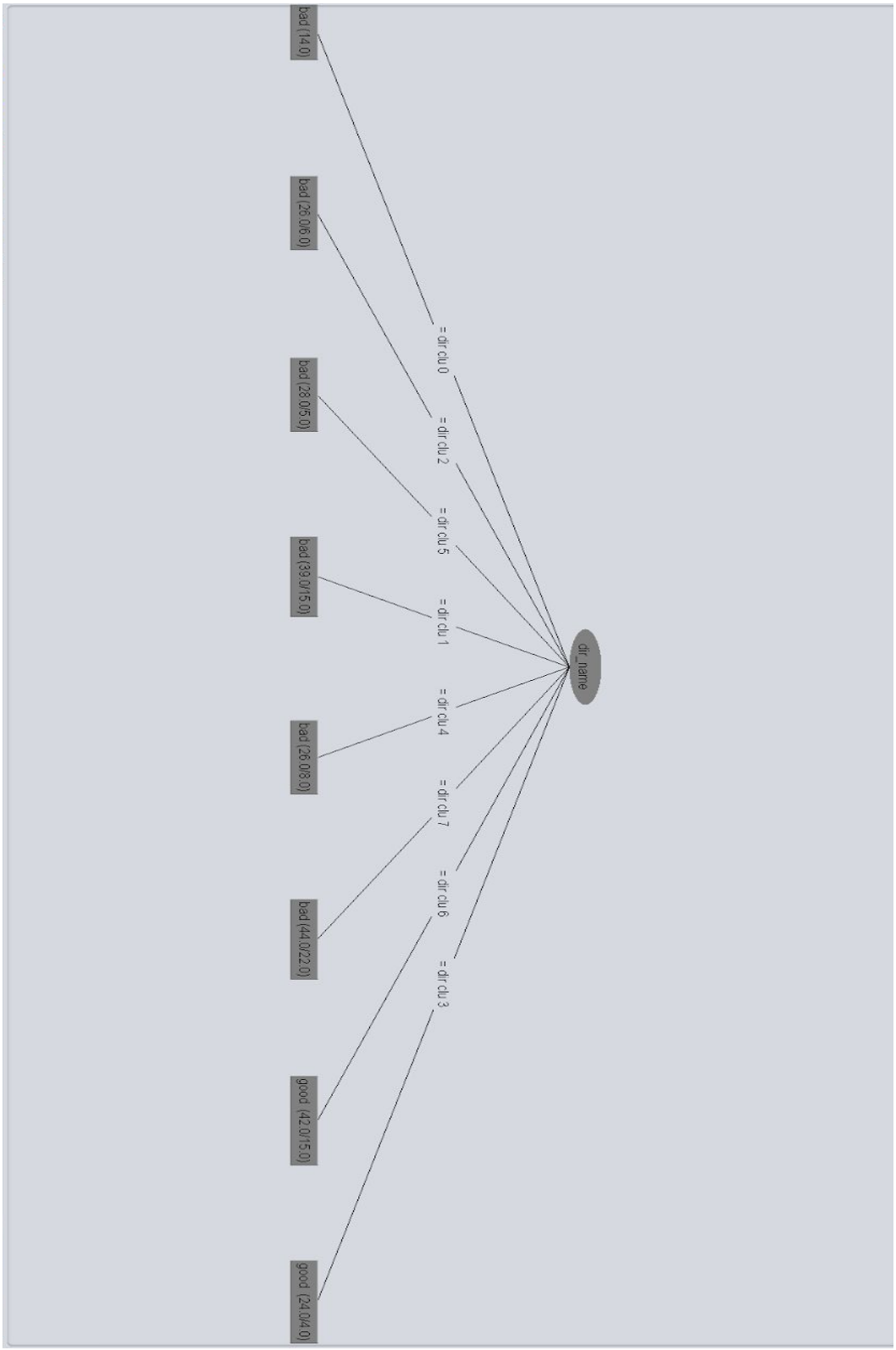

Figure 5: J48 Tree

4. Discussion and Future Work

By comparing the data in Table 1, IBK gives the most accurate result. In the Figure 5, it is clear that director can be determined as a decisive factor.

As mentioned in the previous part, the rank of a movie could be predicted by its director; however, another important element in a movies, actors, seems cannot make this contribution. There are two main reasons which lead to this problem. Firstly, for the majority movies, IMDB just lists all of the movies' actors(staff) including actors and staff without classifying. Therefore, some of those actors might be carefree casts, who make small impact on the rank. Secondly, in order to avoid fortuitous, actors should be filtered, for example, they have to involve at least 5 movies. However, under the filtered sample, the number of satisfied actors is very limit and not accurate if there are someone just be the guest performer of many movies. Thus we still need to work on the algorithm on filtering the main casts and staff who can strongly influence the rank of a movie.

5. Conclusion

Generally, in order to predict a movie is good or bad. We selected 4 attributes which are director, actor, genre and rank to make the prediction. The original data was too huge to use the Weka to do some prediction, therefore, we choose the recent 5 years movies, select the actors who plays 15 to 24 times from 2012 and limit the the numbers of votes to bigger than 50000. We use the Sql shell to output the data as CSV file to imported in the data mining tool WEKA.

Furthermore, because there are countless actor's name and director's name, we separate the actors and director to several clusters according to their average score by the WEKA. Later, we implemented the different classifiers over the data and finally we chose IBk, Classification via Regression, Logistic, J48 which given the acceptable accuracy arround 70%. According to the result, we found some interesting conclusion shows that directors is the most important element to predict the success of a movie, even though we suppose actors and genre play important roles of the success of movie as well.

6. Reference

[1] Wikipedia contributors. "IMDb." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 2 Apr. 2017. Web. 3 Apr. 2017. <https://en.wikipedia.org/wiki/IMDb>