A data mining approach to analysis and prediction of movie ratings

M. Saraee, S. White & J. Eccleston
University of Salford, England

Abstract

This paper details our analysis of the Internet Movie Database (IMDb), a free, user-maintained, online resource of production details for over 390,000 movies, television series and video games, which contains information such as title, genre, box-office taking, cast credits and user's ratings.

We gather a series of interesting facts and relationships using a variety of data mining techniques. In particular, we concentrate on attributes relevant to the user ratings of movies, such as discovering if big-budget films are more popular than their low budget counterparts, if any relationship between movies produced during the "golden age" (i.e. Citizen Kane, It's A Wonderful Life, etc.) can be proved, and whether any particular actors or actresses are likely to help a movie to succeed. The paper also reports on the techniques used, giving their implementation and usefulness.

We have found that the IMDb is difficult to perform data mining upon, due to the format of the source data. We also found some interesting facts, such as the budget of a film is no indication of how well-rated it will be, there is a downward trend in the quality of films over time, and the director and actors/actresses involved in a film are the most important factors to its success or lack thereof.

The data used in this paper is not freely distributable, but remains copyright to the Internet Movie Database inc. It is used here within the terms of their copying policy. Further distribution of the source data used in this paper may be prohibited.

Keywords: IMDb, Internet Movie Database, data mining, classification, movies, films.

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)
© 2004 WIT Press, www.witpress.com, ISBN 1-85312-729-9

1 Introduction

The IMDb is an excellent resource to find detailed information about almost any film ever made. It contains a vast amount of data, which undoubtedly contains much valuable information about general trends in films.

Data mining techniques enable us to uncover information which will both confirm or disprove common assumptions about movies, and also allow us to predict the success of a future film given select information about the film before its release. The main difficulty in attempting to use data mining to extract useful information from the IMDb is the format of the source data – it is only available in a number of inconsistently structured text files.

The outcome of this research is therefore twofold; it provides tools/techniques to transform the IMDb data into a format suitable for data mining, and provides a selection of information mined from this refined data, in section 4.2 Experimental results.

The organisation of the paper is as follows: Section 2 provides more details about the problem domain and the particular problems in attempting data mining of the IMDb. Section 3 gives an overview of the techniques we use to perform our analysis. Section 4 describes the actual analysis performed, and then presents the results and a discussion thereof. Section 5 gives the conclusions reached and a note about possible further work.

2 Problem statement

As mentioned in section 1, the main problem encountered when attempting to mine the IMDb data is the source format. The data is provided as forty-nine separate text files. The common factor linking the information in these files is the title of the movie, which is in fact, a title with the production year in brackets appended, to account for multiple different versions, e.g. Godzilla (1954), Godzilla (1998).

The files themselves are in a variety of formats, with no conventions such as Comma Separated Values (CSV) used – the data is laid out to be human readable, not machine-readable. The data is generally consistent, but some errors are present. Much of the data is also free text, such as paragraphs giving film overviews, or lists of quotations. This data is unsuitable for data mining without the additional use of natural language processing techniques for information retrieval/extraction.

These problems mean that a major part of this research will be dedicated to the first part of the Knowledge Discovery in Databases (KDD) process – the Cleaning and Integration of the source data.

Once the data has been suitably cleaned and integrated, it will then need Selection and Transformation, to translate the textual information (where necessary) into numerical information which can be better analysed by data mining processes. This stage will also discard irrelevant data, and may select a subset of the data to be mined, since the original set may still have hundreds of thousands of records.

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)
© 2004 WIT Press, www.witpress.com, ISBN 1-85312-729-9

Finally, data mining will be performed. We intend to perform relevance analysis to see what factors contribute most to a high rated movie, clustering to attempt to detect any relationships between the year a film is produced and its rating, and finally classification to attempt to classify the general rating of upcoming films based upon known information.

3 Methodology

This section describes the general process by which the research is performed.

Not all of the forty-nine source files contain information that would be both feasible to mine and relevant to finding interesting information. Initially we therefore select which of the files to include or exclude.

Next comes the cleaning and integration of the selected files. We develop a Java application to process the files and extract the information. This application transfers the data into a series of tab-separated files. These files are then imported using Microsoft Access to create a relational database, with the movie title being the common attribute to link each table.

Following the database creation, we move on to selection and transformation of the data. Using the relational database as our source, we transform selected text attributes into a numerical format to facilitate mining, and produce database queries to select the data to be mined. These queries both group the data into views with only the required attributes, and filter the source data to reduce its size for mining in a realistic period of time.

Finally comes the actual data mining. This is performed using Neurosoft Envisioner [2] and, where necessary, Microsoft Excel to produce graphs of the results.

4 Experimental evaluation

4.1 Experimental setup

4.1.1 Step 1: pre-selection

The following data files were discarded as simply being irrelevant to the task, containing no information that would provide interesting mined results:

aka-names, aka-titles, cinematographers, complete-cast, complete-crew, costume-designers, editors, german-aka-titles, iso-aka-titles, italian-aka-titles, keywords, laserdisc, miscellaneous, miscellaneous-companies, movie-links, producers, production-designers, sound-mix, soundtracks, technical

Note: The "keywords" file was originally thought to be relevant, but later discovered to be too inconsistent in its completeness to be of use.

The following data files were discarded for having full text that would require natural language processing techniques to adapt for mining:

alternate-versions, biographies, crazy-credits, goofs, mpaa-ratings-reasons, plot, quotes, taglines, trivia

The following data files were considered possibly relevant, but not enough to warrant the development effort required for their inclusion:

special-effects-companies, writers

With all the previous files eliminated, the following remained:

actors, actresses, business, certificates, color-info, composers, countries, directors, distributors, genres, language, literature, locations, movies, production-companies, ratings, release-dates, running-times

With the exception of "ratings", which had a format structured enough to be imported into Access based upon column-width field delimitation, these files all required Java processing to extract their data.

4.1.2 Step 2: cleaning and integration

In general, each file has some explanation information at the top, and then additional explanation and copyright at the bottom. The data itself is in the middle.

The Java program was developed to accept an input file and a parameter switch (such as "-actors") to indicate the file of file being used. For each file, it then scans the input line by line until some known pattern is found that indicates the actual data starts on the next line. This pattern is different for each file type. It then enters the main processing in a while loop, until a known stop pattern or the end of the file is encountered.

The main processing loop is once again different for each file, and specific to the data in that file. A general problem found in all the input files is inconsistent spacing – one line may have a value separated from the next one by a single space, the next line may use a tab, the next line again may be two or more tabs.

There were also some more specific problems: In the "certificates" file, in some cases, the country was missing but the certificate was "T". Only Spain has this rating, so country was added manually. In other cases, the country was missing but the certificate was "R". Many countries use this rating, so country was manually set to "unknown". In the "color-info" file, if a film is in both Black & White and Colour (e.g. Schindler's List), this is represented by the film being present on two lines of the input file, once with each. To account for this, the code scans ahead to the next line to check if two lines refer to the same film. If they do, the information from both lines is combined in the output. In the "literature" file, we were only interested in knowing which films were adapted from books. Lines in this file are pre-pended with labels (such as NOVL: and MOVI:) which had to be accounted for. In the "release-dates" file, the dates were split up into day, month and year for easier mining, but not all movies had all these fields. In the "business" file, not all values were available for each line – the country for a particular gross figure may be missing for example.

The output from processing the files with the Java program is a series of CSV files. These files were then imported into Access one by one. The files were used to generate tables with the names set as the original filename. The primary key was set to be the "Movie" field wherever this was unique, and all tables were linked on that field.

Finally, the resulting tables were checked for any obvious errors in this step. The only one found was in the "Business" table, where the film "Fear and Loathing in Las Vegas (1998)" was found to have a gross figure of 10^{31} US Dollars. This error was present in the original source file, and the record was deleted.

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)
© 2004 WIT Press, www.witpress.com, ISBN 1-85312-729-9

4.1.3 Step 3: selection and transformation

The main transformation required was to calculate a numerical rating for the directors, actors and actresses. We predicted that these would all play a major part in the success of any movie, and thus represent important information we could not do without. In the default format however, each movie would have multiple records; one for each director, actor and actress combination. This would be useless when attempting to classify the data, since there are far too many discrete values. The actors table, for example, contains over five hundred thousand different actors.

Lacking a source of rating information for directors, actors and actresses, we decided to work back from the data we did have ratings for – the movies themselves. We decided that a suitable measure of the rating of a person would be to take the average user rating of all the movies in which they had appeared or directed.

Our first attempt at constructing an average rating was the following SQL query:

```
SELECT Actors.ActorName, Avg(Ratings.Rating) AS AvgOfRating
FROM Ratings INNER JOIN Actors ON Ratings.Movie = Actors.Movie
GROUP BY Actors.ActorName;
```

This query had two problems – firstly, it took far too long to execute. A test taking only the actors whose surnames began with the letter "A" took over 10 minutes to execute. More importantly, it gave unfair bias towards actors who had appeared in a single movie with only a few votes but a high rating. For example, "Garland L. Yee" had only appeared in "Bingo (2003)", which was given a perfect 10 by the nine people who had reviewed it. He thus earned a perfect 10 rating, whilst the famous actor Sean Connery got a rating of 6.6. Clearly, this was not an accurate picture of the real world, since people might go to see a movie solely on the basis of a popular actor, such as Sean Connery, playing the lead role. The same cannot be said of Garland L. Yee.

To remedy both of these problems, an extra condition was added to the query which builds the average actor ratings – now, it would only calculate the average of the films which had at least one thousand votes. This both eliminates the bias of an unknown movie with a few high votes, and reduces the amount of data to be processed. The new query was thus:

```
SELECT Actors.ActorName, Avg(Ratings.Rating) AS Rating
FROM Ratings INNER JOIN Actors ON Ratings.Movie = Actors.Movie
WHERE (((Ratings.NumberOfVotes)>999))
GROUP BY Actors.ActorName;
```

This still took a significant amount of time to execute however, and so the query was modified to build a new table, ActorRatings, containing the results. This table has 66,075 records, and holds the ratings for all actors who have appeared in at least one movie with one thousand or more votes. The process was repeated with modified parameters to build ActressRatings and DirectorRatings tables.

With numerical ratings for the directors, actors and actresses, we were ready to build a query to total up how these contributed to a film. Initially, we tried a simple sum of all the ratings for each director involved in a particular film, to get a "Director Rating" for that film. This gave an unfair advantage to some films

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)
© 2004 WIT Press, www.witpress.com, ISBN 1-85312-729-9

with numerous directors however, and again did not reflect reality – having five high-rated directors working on a film will not make it better than one or two directors will. We decided to take the average rating of the directors involved instead, for a more realistic value.

For the contribution of actors and actresses to a film, we decided to keep the idea of the sum function. Whilst this would mean that films with more actors and actresses got a higher actor or actress rating figure, it is the case in the real world that having six or seven famous actors in a film will make it more likely to succeed than only one or two.

Duplicates are a major problem when attempting to perform a Sum or Average on a query that joins several tables, and the figures we were getting were incorrect. We eventually remedied this problem, using some complex queries that reference other queries as their data sources. The exact queries used are not listed due to their length. As these queries took an exceptionally long time to execute (more than forty-five minutes), we cached the results in a new table, MovieDirectorActorActressRatings.

Several other queries were produced in this step in preparation for the data mining process. As with step 2 in the knowledge discovery process, the new tables were then checked for errors, and again one was found – "Around the World in Eighty Days (1956)" had an actor rating of 5988 and actress rating of 2758. These figures were over three times the value of next highest ones in the database, and did not seem to tally with the numbers and ratings of the actors/actresses involved. The record was deleted.

4.1.4 Step 4: data mining

With the data finally cleaned, integrated, selected and transformed, the actual data mining could begin.

Many of the mining operations we would be performing would relate to the user rating of a film, which ranges from 1-10. To aid classification and other analysis of this continuous numeric value, we generalised the rating into four categories:

7.5-10	Excellent
5-7.4	Average
2.5-4.9	Poor
1-2.4	Terrible

Our first mining was to analyse the first theory put forward in our abstract: are big-budget films are more popular than their low budget counterparts?

We decided to define "popular" by the rating of a film, rather than its box office takings, since the budget and gross figures vary wildly over time; from a budget of just 180 British pounds for "Hamlet (1910)", to a gross of over 1.2 billion US dollars for "Titanic (1997)"

To reduce the complexity of dealing with the full source data, which contains budget information in many currencies, we restricted the query to movies whose budget was in US dollars. We then produced a simple analysis – a classifier for rating, based solely upon budget. The result of this is given in the next section.

Our next mining attempt was to try to identify any "golden age" of films, with a group of high rated old classics. After some unsuccessful attempts at clustering using Envisioner, we switched to using Microsoft Excel. When performing this analysis, we found it necessary to consider all movies with ten or more votes, rather than the more usual one thousand or more, since most of the very old films do not have many votes. We first attempted a simple plot of all movies on a scatter graph with year on the horizontal axis, and rating on the vertical. Whilst this did show a trend of there being few low-rated old movies, it was not meaningful and easily understandable information – the right side of the graph was a solid mass of points. Our first refinement attempt was to group all the movies with the same year and rating, and plot a bubble chart with the size of the bubble representing the number of films at that point. This gave better results, but was still rather difficult to interpret, with many overlapping values. Having noticed the benefit from the grouping in the previous attempt, we then produced new queries to round ratings of movies to the nearest whole number, and group them by decade. We then plotted these results on a bubble chart. This gave easily readable results, which are given in the next section.

Our next mining attempt was to see what factors are most relevant to the rating of a movie. We used a new UniversalClassifier query, which included all the attributes we thought may be relevant. We then performed relevance analysis on this query. The results are presented in the next section.

Finally, we had hoped to use this relevance analysis to build a classifier to predict the rating category of movies currently in production. However, this proved to be impossible, since certain factors such as the certification and budget were not available to us for movies not yet completed. We proceeded to build a classifier using only the information about the most important factors – the actors, actresses and directors involved. Using a query that joined ratings with the MovieDirectorActorActressRatings table, we produced a classifier for rating. We then used another query to select all the movies due for release later this year or in 2005-2006, and build their director, actor and actress ratings. Finally, we applied the classifier produced earlier to these movies to attempt to predict their rating. The results are in the next section.

4.2 Experimental results

For the first classifier, the rating based solely upon the budget, the results were as we expected; budget alone is no means to tell if a film will be good or not. The classifier generated a very large number of leaf nodes, being unable to classify all the records with 100% certainty even after allowing ten levels in the decision tree.

For the bubble chart of the number of movies per rating per decade, the chart clearly shows an increase in the relative number of low-rated movies in recent years, even accounting for the overall increase in movie production. Going back to 1960 and earlier, the relative number of movies with a rating of 4/10 or below is much smaller. The bubble chart is presented below, in figure 1.

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)
© 2004 WIT Press, www.witpress.com, ISBN 1-85312-729-9

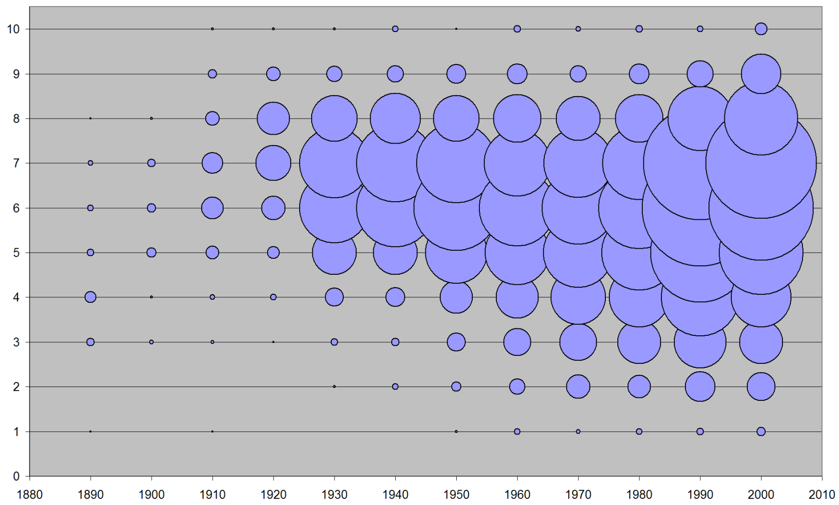

Figure 1: Bubble chart of movies per rating per decade.

Table 1: Predicted movie ratings.

Movie	Rating
5-25-77 (2005)	Average
Ask the Dust (2005)	Average
Batman Begins (2005)	Excellent
Because of Winn-Dixie (2005)	Poor
Bewitched (2005)	Average
Bridget Jones: The Edge of Reason (2004)	Average
Cars (2005)	Excellent
Cinderella Man, The (2005)	Average
Go Go Tales (2005)	Average
Great New Wonderful, The (2005)	Poor
Harry Potter and the Goblet of Fire (2005)	Average
In Her Shoes (2005)	Average
King Kong (2005)	Average
Kingdom of Heaven (2005)	Average
Last First Kiss (2005)	Average
Lord of War (2005)	Average
Madagascar (2005)	Average
Pink Panther, The (2005)	Average
Pirates of the Caribbean 2 (2006)	Excellent
Robots (2005)	Average
Sin City (2005)	Average
Solo (2005)	Average
Son of the Mask (2005)	Average
Southland Tales (2005)	Excellent
Star Wars: Episode III (2005)	Average
Stealth (2005)	Average
Toyer (2004)	Average
Walk On (2005)	Average
Wallace & Gromit Movie: Curse of the Wererabbit, The (2005)	Excellent
X-Men 3 (2006)	Excellent
Zorro 2 (2005)	Average

For the relevance analysis of the rating of a movie, the results show that by far the most important factors are the actors and actresses appearing in the movie, being over 90% relevant each. The director also plays a large part, at around 55%, and the budget is quite significant at around 28%. The certification the movie receives and its genre play only a small part, less than 5% each.

Finally, we come to the classifier to predict the rating of a movie. The table shows that most of the upcoming movies are predicted to be average, with 6% poor, and around 19% excellent. These figures seem consistent with the observed movies over the past few years. Only time will tell if the actual predictions were accurate. The predictions are shown below in table 1.

4.3 Discussion

The results found by our analysis generally seem to confirm some popular assumptions about movies. Firstly, that big budget by no means ensures a high quality film. When attempting to classify on budget alone, the classifier formed very small ranges to fit individual movies or clusters of movies, and generalised very poorly.

There is another popular assumption about movies, which is that there was, at some time in the past, a “golden age” of films, when truly great movies were made, and without any modern special effects or big budgets. When viewing our bubble chart of number of movies per rating per decade, it is important to look not at the absolute size of each bubble, but rather at the relative sizes of each rating bubble within a single decade. This chart seems to show that there were not a large proportion of very highly rated movies in decades past, but there were a significantly lower percentage of low rated movies.

Our relevance analysis of the factors contributing to the rating of a movie yielded nothing particularly unexpected, though one might have assumed the certification would play a larger part, with more movies intended for adults receiving higher ratings.

Finally, our classifier to predict the rating of a movie may be somewhat simplistic – we were not able to include many factors, since much of the information was not available for movies that are unreleased. Although the predictions look believable, it would be surprising if the accuracy of this classification were high given the limited factors evaluation. Time will tell.

5 Conclusions

Overall, we have found that it is difficult to apply data mining techniques to the data in the IMDb. The data needs extensive cleaning and integration, and this consumed a large proportion of the time available for this analysis. In addition, much of the data is in textual rather than numerical format, making mining more difficult. Much of the source data could not be integrated at all, without using natural language processing techniques. Despite these problems, we performed some useful data mining on the IMDb data, and uncovered information that cannot be seen by browsing the regular web front-end to the database.

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)
© 2004 WIT Press, www.witpress.com, ISBN 1-85312-729-9

More importantly, we believe that our research shows promise for further development in this area. Given additional time to incorporate more of the source data available, and some use of natural language processing techniques, other interesting patterns in the data may become apparent. A more accurate classifier is also well within the realm of possibility, and could even lead to an intelligent system capable of making suggestions for a movie in pre-production, such as a change to a particular director or actor, which would be likely to increase the rating of the resulting film.

References

[1] Han, J., Kamber, M., Data Mining Concepts and Techniques, Morgan Kaufmann Publishers: San Francisco, 2001.

[2] Neurosoft S.A., Neurosoft Envisioner, www.neurosoft.gr/products/envi.asp, 1999.

[3] Thearling, K., Data Mining and Analytic Technologies, www.thearling.com, 2004.

[4] Hamilton et al., Knowledge Discovery in Databases, www2.cs.uregina.ca/~hamilton/courses/831/, 2002.

[5] Attar Software Ltd, Active Data Mining Solutions, www.attar.com/tutor/deploy.htm, 2004.

[6] Labovitz, M. L., What Is Data Mining and What Are Its Uses?, www.darwinmag.com/read/100103/mining.html, 2003.

[7] Nautilus Systems Inc, The Data Mining Process, www.nautilus-systems.com/process.html, 1996.

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)
© 2004 WIT Press, www.witpress.com, ISBN 1-85312-729-9