User rating prediction for movies

Achal Augustine Manas Pathak

Department of Computer Sciences
University of Texas at Austin
{achal, manas}@cs.utexas.edu

Abstract

In this paper, we investigate the extent to which a movie's average user rating can be predicted after learning the relationship between the rating and a movie's various attributes from a training set. We also present an algorithm for scoring individual members based on the movie rating and also a neural network framework for estimating various parameters for combining the individual scores for predictions. Finally, we present an evaluation of our strategies on a sample dataset and examine the results.

1 Introduction

The Internet Movie Database (IMDb) is a comprehensive online database having information about movies, actors, television shows, production, etc. The site features 963,309 titles and 2,297,335 people and has a web page for each of them. Also, it provides a rating for the movies based on aggregation of the ratings provided by the individual users. Given the large scale of the data and the degree of interactions between the people, IMDb is a fertile source of data mining problems.

In this paper, we explore the inter-relationship between the "quality" of a movie and that of the artists involved with it. We further develop a predictor model for rating movies under production.

First we collect and pre-process the hypertextual data from the IMDb website and extract information from it. Then we use the movies in IMDb which already have been assigned ratings as a training set and learn a prediction model based on it. We use this model to predict the ratings of the new movies.

The outline of the paper is as follows. First we consider some related work done in solving problems with similar issues. In section 3, we formalize the task of evaluations described above. In section 4, we describe in detail the learning framework and how it is used to make the predictions. In section 5, we describe the implementation details of the system for this purpose. Finally, in section 7 we evaluate this approach using our training and test dataset.

2 Related Work

A prediction model based on kernel regression and model trees has been used to solve a similar problem of movie rating predictions [1]. Using kernel regression, a prediction accuracy of 14.11% deviation from the true rating was reported. The model tree exhibited test error of 14.40%, which is slightly worse than the kernel regression.

Also, there is another data mining problem called the Netfilx challenge which is similar to the problem described in this paper. Netflix [4] released a dataset containing 100 million anonymous movie ratings as an open challenge to develop systems to predict the rating for a movie for individual users. Adam Sadovsky et. al at Stanford University have build a model for predicting the Netflix movie rating prediction using a logistic regression framework [2].

3 Problem Formulation

Let the predicted movie rating of the movie i be represented by m_i. This rating can be formulated in terms of the individual scores of the actors, directors etc as:

$$m_i = \theta_1 H_1 + \theta_2 H_2 + \theta_3 H_3 + \theta_4 H_4$$
$$where:$$
$$H_1 = \frac{1}{1+e^{-\sum_{i=1}^{4} w_i a_i}}, H_2 = \frac{1}{1+e^{-\sum_{i=5}^{7} w_i d_i}}, H_3 = \frac{1}{1+e^{-w_8 p}}, H_4 = \frac{1}{1+e^{-\sum_{i=9}^{10} w_i s_i}}$$

and a_i = score of the i^{th} actor

d_i = score of i^{th} director

p = score of the producer

s_i = score of i^{th} screenplay writer

We can see that the rating of a movie is dependent on the scores of its cast. Hence, for estimating it, we need to score all the cast and also learn the weights which are used to combine them, using the training set. Once we do this, we can use the same equation with the estimated parameters to predict the rating of a new movie.

4 Learning framework

The learning framework includes a ranking algorithm which ranks all the cast members, a neural network which estimates the parameters of the model and a missing value replacement algorithm which estimates values of missing data elements.

4.1 Ranking Algorithm

The purpose of the ranking algorithm is to rank all the participants of the movies in the learning set. The steps of the initial ranking algorithm are given below.

```
Initialize ranking weights to say, 10, 1, 0.5, 0.3
Initialize the score of all cast and crew to  1

for each movie
    get IMDb rating
    for each cast member in movie
        increment the score proportional to the rating
```

Algorithm 1

For e.g.	The Matrix	8.6
Actors:	Keanu Reeves	8.6 * 10 = 86
	Laurence Fishburne	8.6 * 1 = 8.6
	Carrie-Anne Moss	8.6 * 0.5 = 4.3
	Hugo Weaving	8.6 * 0.3 = 2.58

To ensure that the lead actors were given higher credit for their contribution, the weights for cast members at different position in the credit list were kept different. The weights chosen were initially guess estimations but later were modified using regression analysis but did not produce any significant change in the ranking or the final prediction results.

We gave the highest rank to cast member who has the highest score. As it can be seen from the above equations this ranking scheme does gives higher ranks to cast members who have acted in large number of movies even if all the movies where not very successful. The rationale behind this algorithm was that successful cast members tend to get more opportunities to participate in more movies and therefore the number of movies of a cast member is indicative of how successful they were in the industry. Although this ranking gave a good estimate of ranking for most of the cast members several popular and highly rated cast members were ranked below others who were less successful but participated in larger number of movies. To mitigate this problem we modified the ranking algorithm so that the final score was normalized by dividing the aggregate score by the number of movies participated by each cast member. Unfortunately, this normalization did not work well either because this gave highest ranks to relatively unknown cast members when one or two movies of their movies got high IMDb rating.

```
Initialize ranking weights to say, 10, 1, 0.5, 0.3
Initialize the score of all cast and crew to  1

for each movie
    get IMDb rating
    for each cast member in movie
        increment the score proportional to the rating – Avg. IMDb movie rating
```

Algorithm 2

We finally used the above algorithm as it avoided the problems of the previous two algorithms by measuring the deviation of a movie from average IMDb rating of the movies in the learned set and then using this deviation as the effective IMDb rating of the movie for calculation of the ranks. This measurement ensured that movies with less than average rating gave a negative score to the participants and movies which performed better than average gave a positive score. As a result, participants with few successful movies were not ranked very high neither were participants who were involved in a large number of mediocre movies. The cast members with top ranks were the ones who produced highest number of movies which were above the average movie rating.

4.2 Prediction Model

The Prediction model uses a neural network [3] with the structure described the figure below.

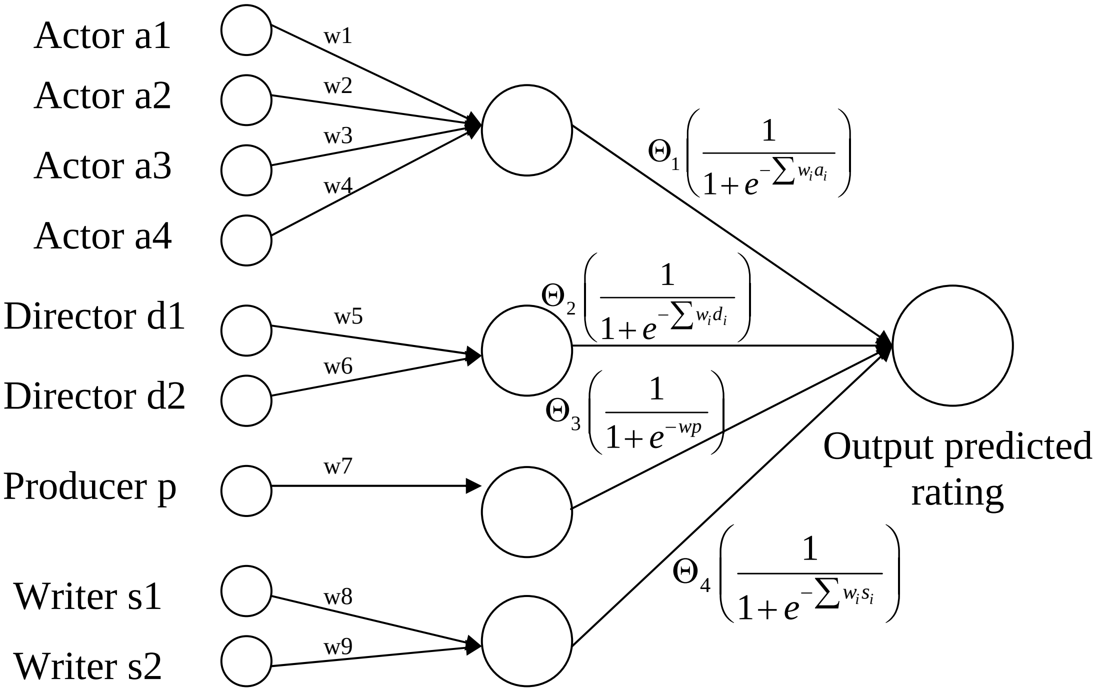
Actor a1
Actor a2
Actor a3
Actor a4
Director d1
Director d2
Producer p
Writer s1
Writer s2
w1
w2
w3
w4
w5
w6
w7
w8
w9
$\Theta_1\left(\frac{1}{1+e^{-\sum w_i a_i}}\right)$
$\Theta_2\left(\frac{1}{1+e^{-\sum w_i d_i}}\right)$
$\Theta_3\left(\frac{1}{1+e^{-wp}}\right)$
$\Theta_4\left(\frac{1}{1+e^{-\sum w_i s_i}}\right)$
Output predicted rating

Figure 1: Neural Network

The inputs a_i, d_i, p, s_i to the network are the scores of the cast members of the movie. The hidden layer of the network consists of four elements representing the four different types of cast members namely actors, directors, producers and screen play writers. The individual scores for each type of cast members are normalized using the sigmoid function and are combined together with parameters Θ_i. We trained the neural network on the input data to get the set of parameters of the network using standard back propagation to give minimal error in the predictions.

4.3 Missing Values

The predictor model takes input the ranks of the cast members of the movies under consideration. The problem of missing values arises when no information is available for any of the cast members. It is possible to deal with missing values either by ignoring those values or by using certain heuristics to substitute them. Whenever the score of a cast member is unavailable we use the scores of the other relevant cast members in the movie as an indicator of their rank and then calculated the approximate value of the cast members missing value. The algorithm for finding approximate value for a missing value of director rank is given below.

```
directorRange = maximum score of any director – minimum score of any director
“castMember”Percentile = “castMember”Rank /”castMember”Range
if directorRank  is Missing :
        if producerRank is not Missing:
            directorRank = producerPercentile * directorRange
        else if actor1Rank is not Missing:
            directorRank = actor1Percentile * directorRange
        else if actor2Rank is not Missing:
            directorRank = actor2Percentile * directorRange
        else if sPlayWriterRank is not Missing:
            directorRank = sPlayWriterPercentile * directorRange
        else:
            directorRank = directorRange /2
        directorPercentile = directorRank /directorRange
```

Algorithm 3

Similar algorithm was used for replacing missing ranks of other cast members

4.4 Regression Analysis

Once the scores of the cast members are calculated, the prediction parameters can be estimated using the neural networks. We used a regression analysis of the weights for estimating the ranks and parameters. We changed the weights of ranking algorithm in small steps and re-estimated the prediction parameters for the prediction model each time by retraining the neural network with the new set of ranks. We repeated this process until the weights of ranking and the parameters finally converged.

5 Implementation

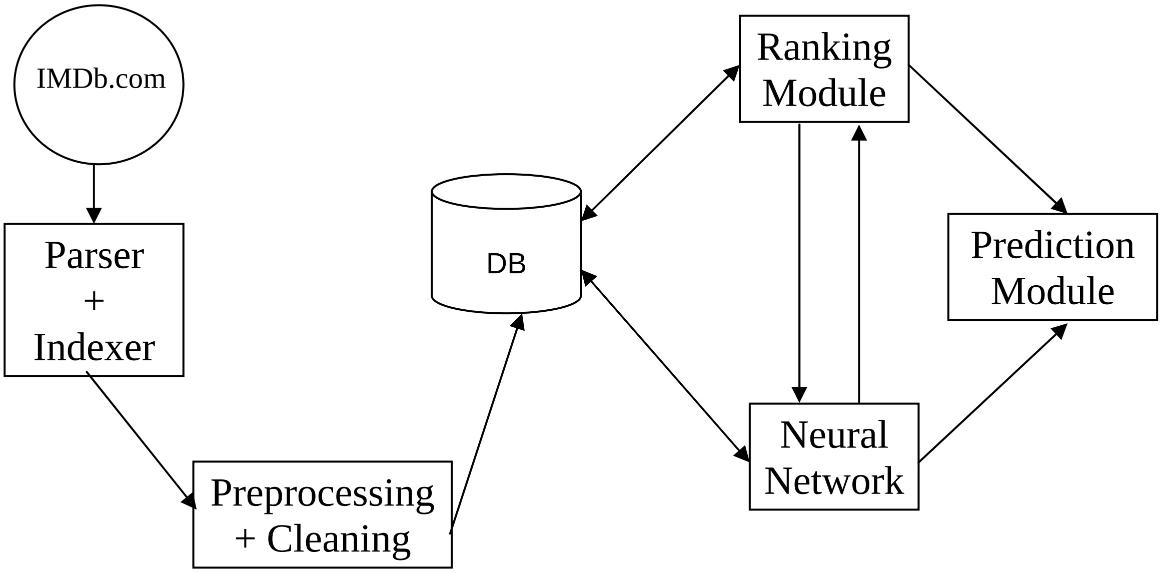

Figure 2: System implementation

As can be seen from the figure 2, the system consists of three major modules: Parsing & Preprocessing module, Database, Prediction module.

a. Parsing & Preprocessing Module

The purpose of this module is to get the raw data about the movies into the database. For doing this, we wrote scripts for crawling and parsing in the Python programming language. For getting the information about the movies, we crawled the imdb.com website (see next section). There was also a module for basic preprocessing tasks like correcting missing data, removing corrupt data etc.

b. Database

For storing the results of parsing and pre-processing, a MySQL database was created. All the movies were indexed by their cast and crew. The prediction modules further used this data for scoring and parameter estimation.

c. Prediction Module

This module consisted of a scoring module and a neural network for weight estimation. Both of these were fully “hand-coded”, i.e. did not use any standard packages, in order for us to have finer control over the step size, number of iterations and precision of the neural network.

6 Experimental Results

To evaluate the prediction framework described above, we ran the following experiment on a sample dataset.

6.1 Data Collection & Preprocessing

For running the prediction experiments, we extracted the data about 1100 movie titles from imdb.com. These titles consisted of top 100 movies of each year from 1995 to 2006. We used the movies from 1995 to 2005 to train our learning system and those from 2006 for prediction results. The movies from 2006 were used to test our predicted results.

Also, we crawled all 8000 movies of 2007 and proceeded to rank them based on the predicted rating to get the top 20 unreleased movies of the year. (see Appendix B for list of top 30 movies of 2007)

6.2 Movie Predictions

We ran the model on learning data and predicted the movie ratings for movies the top 100 movies of 2006. We used the following equation for estimating the error in our prediction.

$$Average\ \ test\ \ error = \frac{1}{\left\|\{test\}\right\|} \sum_{x \in \{test\}} \left| predicted\ \ rating(x) - actual\ \ rating(x) \right|$$

The average error for 100 movies was found to be 0.983, i.e. the error in prediction was 9.83%. Also, 35 of 100 movies were predicted with an error less than 0.5.

The larger errors in prediction were reported in cases where there were missing data about cast members and also in predicting movies which had exceptionally high rating or very low rating. The results showed a great level of reliability given the fact that movies in general show high variance in their quality and a large number of factors influence the success of a movie. (see Appendix A for tabulated results of the experiment).

7 Conclusions & Future Work

We have shown that it is possible to predict the ratings of a movie based on the scores of its crew members. Using the prediction model described above, we were able to do so with the accuracy of 9.83%.

In our model, we realized that a major source of error while making predictions is the fact that for some movies the crew members are not there in the training set. Although it is difficult to predict the ratings for the movies with a totally unknown cast, we have employed certain heuristics to minimize the error. Also, there can possibly be other factors like release date, location, plot, etc. which influence the success of a movie. However, we have not considered them at all in our model due to their non-quantifiable nature. Also, as the model is "time independent", the scores of artists with long tenures tend to get averaged out and may not be very accurate in predicting their successes in the current movies.

However, more sophisticated techniques can be used to further improve the accuracy of our model which we can incorporate in the future. One aspect is to make the algorithm more intelligent to avoid local minima in parameter estimation. Also, to further improve the accuracy, we plan to include better encoding the movie attributes, perhaps even parsing semantic textual data such as plot description.

References

[1] N Armstrong, K Yoon. Movie Rating Prediction. Technical Report, Carnegie Mellon University

[2] A Sadovsky, X Chen. Evaluating the effectiveness of regularized logistic regression for the Netflix movie rating prediction task. Stanford University

[3] R Hecht-Nielsen. Theory of the backpropagation neural network. IJCNN 1989, San Diego, CA

[4] J Bennett, S Lanning. The Netflix Prize. KDD Cup & Workshop 2007, San Jose, CA

Acknowledgement

We thank Prof. Joydeep Ghosh for his valuable guidance during the course of this project and paper.

Appendix A

Rank	**Actor**	**Score**
1	Tom Hanks	115.798
2	Johnny Depp	104.754
3	Edward Norton	68.542
4	Ewan McGregor	68.244
5	Russell Crowe	64.742
6	Kevin Spacey	60.612
7	Sean Penn	59.636
8	Al Pacino	56.506
9	Tom Cruise	50.402
10	Tobey Maguire	48.556
11	Billy Bob Thornton	45.732
12	John Cusack	44.79
13	Jake Gyllenhaal	42.85
14	Sean Astin	41.546
15	Audrey Tautou	40.87

Table 1: Top 15 actors in the learning set

Rank	Director	Score
1	Steven Spielberg	96.428
2	Peter Jackson	95.688
3	Joel Coen	67.073
4	Quentin Tarantino	66.792
5	Richard Linklater	57.74
6	David Fincher	56.792
7	Ang Lee	54.688
8	Hayao Miyazaki	53.844
9	Alejandro Amenabar	51.792
10	Pedro Almodovar	47.792
11	Christopher Nolan	46.844
12	Nick Park	46.066
13	Frank Darabont	45.844
14	Martin Scorsese	44.688
15	Tim Burton	44.636

Table 2: Top 15 directors in the learning set

Rank	Company	Score
1	DreamWorks SKG	92.44
2	20th Century Fox Television	89.792
3	Channel Four Films	87.428
4	Universal Pictures	65.596
5	Miramax Films	64.012
6	Aardman Animations	62.792
7	Canal+	51.428
8	Fox Searchlight Pictures	50.74
9	Good Machine	39.74
10	Pixar Animation Studios	37.844
11	Canal+ Espana	35.844
12	American Empirical Pictures	34.844
13	DENTSU Music And Entertainment Inc.	33.896
14	Bandai Visual Co.	33.844
15	A Band Apart	32.792

Table 3: Top 15 companies in the learning set

Rank	Writer	Score
1	Quentin Tarantino	72.74
2	J.R.R. Tolkien	67.844
3	Jane Austen	56.792
4	Alejandro Amenabar	54.74
5	Andy Wachowski	52.636
6	Richard Linklater	49.792
7	Pedro Almodovar	47.792
8	Paul Thomas Anderson	43.792
9	Woody Allen	41.532
10	Kar Wai Wong	40.844
11	Kevin Smith	40.688
12	William Shakespeare	39.74
13	Andrew Niccol	39.74
14	John Lasseter	38.844
15	Krzysztof Kieslowski	38.844

Table 4: Top 15 writers in the learning set

Appendix B

Predicted Rank	Movie	Predicted Rating
1	The Lovely Bones	6.865670718
2	Route 66	6.845175763
3	Babylon Fields	6.828446707
4	Back to You	6.828446707
5	Burn Notice	6.828446707
6	Company Man	6.828446707
7	Crowned	6.828446707
8	Fugly	6.828446707
9	Journeyman	6.828446707
10	K-Ville	6.828446707
11	The Minister of Divine	6.828446707
12	Miss/Guided	6.828446707
13	Nice Girls Don t Get the Corner Office	6.828446707
14	Supreme Courtships	6.828446707
15	Two Dreadful Children	6.828446707
16	Women s Murder Club	6.828446707
17	The Lord of the Rings Online: Shadows of Angmar	6.689158445
18	Can You Hack It?	6.67245495
19	Accepted	6.659543616
20	Northanger Abbey	6.618652548
21	Persuasion	6.618652548
22	Se jie	6.602253088
23	I Think I Love My Wife	6.585830959
24	Ceux qui restent	6.584359905
25	Kid Svensk	6.584359905
26	Utomlyonnye solntsem 2	6.584359905
27	The Savages	6.579958314
28	Scrolls	6.579958314
29	Sweeney Todd	6.544343893
30	Shine a Light	6.541211279

Table 5: Predicted top 30 movies of 2007