Predicting Movie Incomes Using Search Engine Query Data

Chanseung Lee[1], Mina Jung[2]

[1] South Eugene High School
Eugene, OR USA
chanseunglee0@gmail.com

[2] Electrical Engineering and Computer Science
Syracuse University
Syracuse, NY, USA
mijung@syr.edu

ABSTRACT

Massive data sources resulting from human interactions with the Internet may offer a new perspective on the behavior of market. By analyzing Google query database for search terms related to movies, we present a method of analyzing large numbers of Google search queries to predict revenue of movie incomes. Our results illustrate the potential that combining extensive behavioral data sets offers for a better understanding of collective human behavior.

KEYWORDS

Prediction; Logistic Regression; Data Mining

1 INTRODUCTION

By analyzing the search queries from the surfers, the possible societal applications are extensive. For example, Ginsberg et al. [1] estimated the spread of influenza in the United States based on Google search logs. They processed search queries from Google search logs and generated a comprehensive model for use in influenza surveillance, with regional and state-level estimates of influenza-like illness activity in the United States.

Predicting the financial success of a movie is a challenging open problem. Sharda and Delen [2] have trained a neural network to process pre-release data, such as quality and popularity variables, and classified movies into nine categories according to their anticipated income, from ‘‘flop’’ to ‘‘blockbuster’’. With test samples, it is evident that the neural network classifies only 36.9% of the movies correctly, while 75.2% of the movies are at most one category away from the anticipated category.

Joshi et al. [3] have built a multivariate linear regression model that joined meta-data with text features from pre-release critiques to predict the revenue with a coefficient of determination.

Since predictions based on classic quality factors fail to reach a level of accuracy high enough for practical application, usage of user-generated data to predict the success of a movie becomes a very tempting approach.

Mestyan et al. [4] build a predictive model for the financial success of movies based on collective activity data of online users. They show that the popularity of a movie can be predicted before its release by measuring and analyzing the activity level of editors and viewers of the corresponding entry to the movie in Wikipedia.

In this paper, we try to find out whether a statistical prediction of the box-office success of a movie can be made using worldwide searchers' queries along with other information such as rating, the quantity of theaters the movie was starred in, and the length of the title. Our proposed system builds an automated method of predicting movie incomes from search queries.

In the next section we describe about the data sources used in the prediction, and the characteristics of query data are described in Section 3. In Section 4, we explain the predicting method and experimental results. In Section 5, we conclude this paper and suggest future directions.

2 DATA SOURCES

The main variable we choose is the movie query data which were acquired from Google Trends. There are three additional variables which can possibly improve the performance of the prediction system. We used variables about a movie such as Income(I), Rate(R), and number of theaters(T) from Box Office Mojo site [5]. The data variable we use is described in the following.

1) Movie Query : Google Trends [6] provides weekly search query frequencies of movie titles. Instead of using weekly query data, queries of four weeks are combined into one variable. From query data, we extract four variables (M1, M2, M3, M4) where M1 means the sum of queries during 4 weeks before the release date, and M2 means queries of 5-8 weeks before the release date, etc. Altogether, we are using 16 weeks of query data as input.
A basic and necessary assumption of using search query data for prediction is that people who enter queries of movie title are interested in the movie, and thus likely to see the movie.
Sometimes the words within movie titles are so common (e.g. Highway) that the queries about the title of the movie extensively overlap with unrelated queries. In these cases, some other related queries including actors and directors are used.
We manually typed in movie titles at Google Trends and downloaded query data in csv form. We could collect as many as 138 movies released in January, February, or March of 2014 from Google Trends (4 movies were omitted due to lack of sufficient query data).

2) Rate(R) : We use the rating (G, PG, PG-13, R, NR) of movie.

3) Number of Theaters(T) : The number of theaters the movie made it to.

4) Number of Words in Title(W) : It is sometimes known that movies with simple titles succeed. We want to see the effect of the number of words in movie title.

5) Income of Movie(I) : For the income of the movie, we used the income of the first weekend after the release date. Logarithmic value of Income(I) is predicted in this paper.

3 CHARACTERISTICS OF QUERY DATA

In this section, the characteristics of movie query data are analyzed in terms of frequency trends. In most movies, as expected, their query frequencies increase drastically near the opening date. These movies include *Life of a King, Noah, Pompeii, Ride Along, The LEGO Movie, RoboCop, The Raid 2, Veronica Mars, That Awkward Moment, Gang of Ghosts, Mistaken by Strangers, Muppets Most Wanted, Son of God, That Awkward Moment, The Monument Man, The Nut Job, etc*. As Figure 1 shows, we can see that the queries of movies increase dramatically near the opening date of the movie (the x axis is the number of weeks before the movie was released, and the y axis is the frequency of the search queries of the movie).

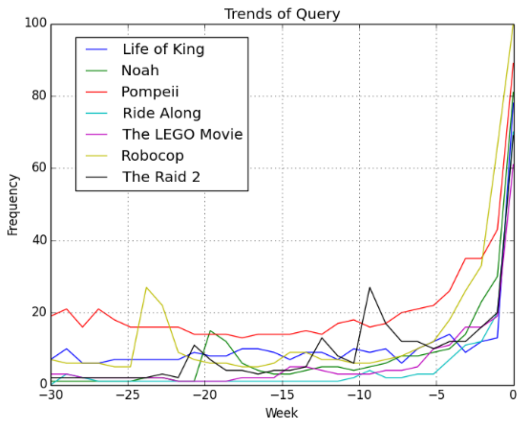

Figure 1 : Characteristics of ordinary movies

In some movies, their titles are so common that their queries do not show significant increase near opening date. These movies include *Gloria*, *Infliction*, *The Rocket*, *Refuge*, *Teenage*, *The French Minister*, *Visitors*, *Highway*, *etc*. When people type in these queries, the queries are not always directed towards the movie. For example, when a Google searcher enters query 'Teenage', he could mean the movie "Teenage" or teenagers itself. For these movies, the queries of movie title is not a good indicator of movie income. Therefore, we used query data for the names of main actor and director of the movie. These queries (title, actor, and director) are combined and their average is used as input. Figure 2 shows the frequency trends of these general term movies.

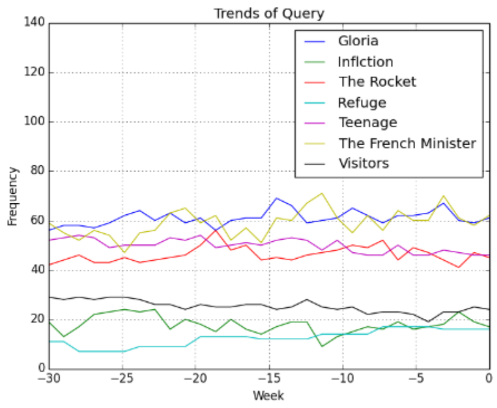

Figure 2 : Characteristics of general term movie

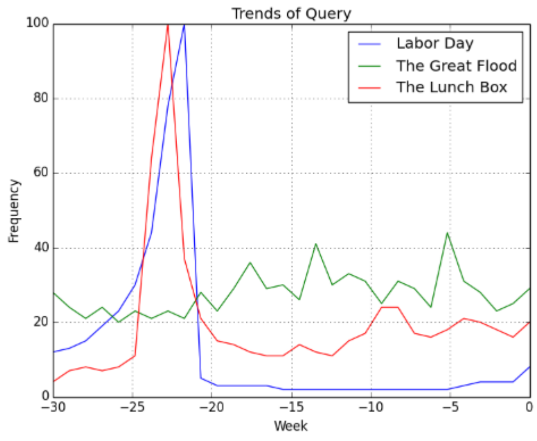

Figure 3 : Characteristics of social factor movie

Queries of some movies are affected by social factors. These movies include *Labor Day* (at peak near Labor Day), *The Great Flood* (at peak when an actual flood strikes), *The Lunchbox* (launched in India first), etc. Figure 3 shows the properties of queries about these movies. In these cases, we used the same approach used with movies with common terms. We used queries about main actor and director of the movie, and their average is used in the experiments.

4 PREDICTION METHODS

Simple Multivariate Linear Regression [7] is the first method we chose to attempt. Linear Regression creates a line among a scatterplot that minimizes residuals (sum of squared errors) given as follows.

$$E = \min \sum_{i} (I_T - I_P)^2$$

Where I_T and I_P mean the true income and predicted income of the movies, respectively. We look for the following line which minimizes the above error E.

$$I_P = w_R \cdot R + w_T \cdot T + w_W \cdot W + w_{M1} \cdot M1 + w_{M2} \cdot M2 + w_{M3} \cdot M3 + w_{M4} \cdot M4 + w_0$$

In this formula, (R, T, W, M1, M2, M3, M4) are the variables we use for linear regression while *w* means the corresponding weight of each variable. The goal of learning is to correctly estimate the weight(*w's*) of each variable in regression line.

We developed a Python program that uses Multivariate Linear Regression from library sklearn [8] of Python. Parameters of Multivariate Linear Regression were set to be the default values. Figure 4 shows the relationship between true Income and predicted Income of the movies which was estimated by the input variables. As we can see in Figure 4, the result was promising. Many prediction points are gathered near/around diagonal line, and its average error is 621.9. This results shows that we are able to predict movie incomes by using query data with a reasonable accuracy.

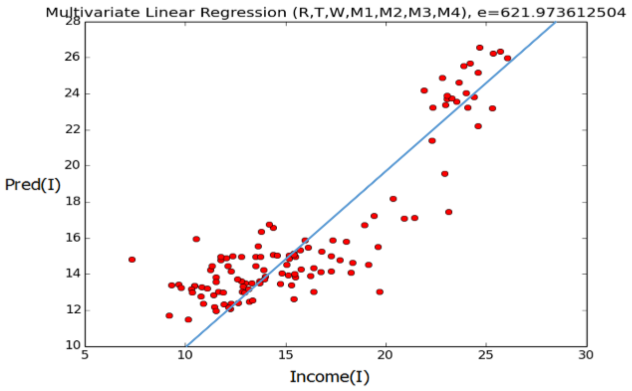

Figure 4 : Results of multivariate linear regression

We were interested in whether we can improve the performance of linear regression system. Therefore, we carried out second experiments using a different linear regression model. The second method we attempted is Regularized Linear Regression (RLR). The reason we tried RLR is that RLR is more robust of noise in data. When noise in data are expected, regularized regression is a powerful tool to disregard noise and solve the problem, since RLR is less considerate to outliers. RLR minimizes both the sum of error and Regularization term; the regularization term is the square of each coefficient w_i.

$$E = \min \sum (I_T - I_P)^2 + \alpha \cdot w_i^2$$

where w_i means the coefficient of input variables (R, T, W, M1-M4, etc) and α means regularization constant. When we run RLR function in sklearn with default parameter values. Figure 5 shows the relationship between true Income and predicted Income using RLR. Many predictions are gathered near the diagonal line, but more spread. The result was worse than Simple Linear Regression, and shows greater error (error=949.1). This result tells us that, in the proposed movie prediction system, most input data are not outliers nor noise.

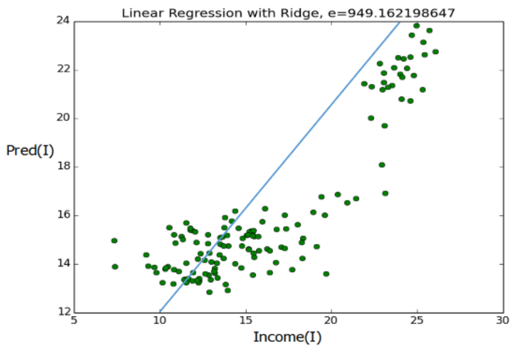

Figure 5 : Results of Regularized multivariate linear regression

In order to see the effect of search query data in movie income prediction, we repeat the same experiments using (R, T, W) data only. Figure 6 shows the result of prediction, and the error has increased to 770.7. This result shows that the search query data could improve the performance of the movie prediction system. Figure 7 shows the result of prediction using query data (M1-M4) only. The error for this case is 1699.2.

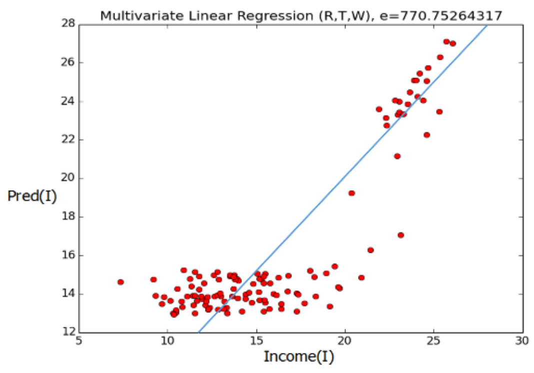

Figure 6 : Results of multivariate linear regression using R, T, W only

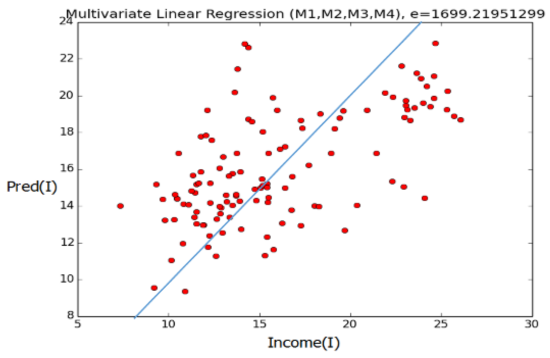

Figure 7 : Results of multivariate linear regression using M1-M4 only

5 CONCLUSIONS AND FUTURE WORK

We developed a program that can predict the future income of movies by using search query data (from Google Trends) and some additional information about movies. We find that a movie's success (total income) is largely based on searchers' queries and is slightly affected by its length of title, rating, and quantity of theaters it reached. Without query data, the prediction drops significantly.

Regardless of its robustness of noise, we found that Regularized Linear Regression wasn't enough to show better performance in predicting movie incomes.

As future work, we will include some other important variables about query and/or movie such as trend of query, regional data, etc. We will also use other learning algorithms to better predict the success of movies.

REFERENCES

[1] Jeremy Ginsberg et al. Detecting influenza epidemics using search engine query data. Nature, 45(9), 2009.

[2] Ramesh Sharda and Dursun Delen. Predicting box-offce success of motion pictures with neural networks. Expert Systems with Applications 30(2), 2006

[3] Mahesh Joshi, Dipanjan Das, Kevin Gimpel and Noah A. Smith.. Movie reviews and revenues: An experiment in text regression. In In Proceedings of NAACL-HLT 2010, 2010.

[4] Marton Mestyan, Janos Yasseri, and Janos Kertesz. Early prediction of movie box office success based on wikipedia activity big data. PLoS ONE, 8(8), 2013.

[5] Box office Mojo. http://www.boxofficemojo.com.

[6] Google trends. http://www.google.com/trends.

[7] Wikipedia. http://en.wikipedia.org/wiki/Linear_regression

[8] Scikit-learn: Machine learning in python. http://scikit-learn.org.