April.21, 2017

CIS 700 - Social Media Mining

Predict Movie Box Revenue

Project #2

Instructor- Dr. Reza Zafarani

Jiachen Ma

SUID:325271944

CONTENTS

Predict movie box revenue by analyzing social media content

1. Introduction

The growth in social media networks has significantly affected the way people interact with others nowadays. Indeed, interacting through online social networking has become ubiquitous and a major factor of making decision in daily life.

Since the content of social media networks has such a great impact on people's life, the interest of this project is to investigate its power at predicting real-world outcomes. Previous research had been done focusing on predicting crime, president election and sentiments through social media information [1],[2]. More specifically, in this project, we considered this task as predicting movie revenues using the information such like cast of players, critic and audience reviews, news and interviews, from rotten tomatoes, which consists of billions of users positively join in creating interesting reviews. Surprisingly, rotten tomatoes.com provides an access to its API, where we can collect critic and audience scores, critic reviews and movie metadata. Later we found that these comments and scores from rotten tomatoes could indeed be used to make quantitative forecasts of man-made markets. This information, such as rules is large enough, is reasonably designed, and is normally more accurate than traditional methods.

Our goal of this project is to build a effective model for predicting film box revenue by analyzing the reviews and scores from rotten tomatoes. The main contribution of this project are as follows:

1. Based on the idea discussed in paper[3], we improve a effective model for predicting film box revenue by analyzing the scores and cast players from rotten

tomatoes.com. And final results shows predictions from our model are better than those produced by normal market website.

2. Our analysis of emotion mining from audience reviews shows a great improvement in prediction model.

And the paper is organized as follows. Section 2 will introduce the background and some related works which has been studied in this area. Section 3 will illustrate the prediction problem and proposes an optimal predictive model. And in this section, we will also examine different features can be measured on rotten tomatoes, and evaluate their practicality for predicting revenue future trends. More specifically, we will construct an autoregression model by evaluating the movie metadata information and audience scores. Besides, we will study how movie reviews will effect on movie revenue, how the opinions with strong emotion spread and how they affect people. Also in the part, by using Random Forest Classifier we will conduct the sentiment analysis and optimize the whole autoregression prediction model based on Sentiment Probabilistic Latent Semantic Analysis. Finally, this paper will test the whole model by using the dataset crawled from rottentomatoes, comparing the final results and real-world movie outcomes, figuring out the accuracy of matching rate between prediction and real-world revenue.

2. Related Work

There has been some prior research on analyzing the correlation between social media content and its performance in daily life. Yao and others[4] illustrated how to generate automated queries to mine blogs for predicting the peak of movie and book sales, while they only used the metadata information. Hazem Hajj and others[5] presented a framework for emotion mining from text in online social networks, however, they did not provide any application for their framework. Although Sitaram Asur and Bernardo A. Huberman have used the chatter from Twitter.com to forecast box-office revenues for movies[3], their prototype is a little simple. And since Twitter is not a social media network about movie, the prediction based on twitter is not so accurate. So on the basis of those outstanding research the goal of this project is proposing a detailed prediction model based on the movie fans review and critic scores from rotten tomatoes.com. Rotten Tomato is an American Film and Television Comprehensive Assessment website. According to the audience survey service by Quant cast , the monthly visit to rottentomatoes.com domain name is 26M Global (14.4M US). And through Fandango API and Tomatometer, we can access movie, theater and ticketing information, we can also collect critic reviews and ratings through the Rotten Tomatoes APIs.

What's more, after deeply digging the Sentiment Analysis on Movie Reviews competition, the project also shows the sentiments extracted from reviews can be used to improve the revenue predicting model.

3. Model

The key issue we plan to solve is trying to accurately predict real-world outcomes by analyzing social media content, specifically, in this project, from the critic and audience reviews on rotten tomatoes.com, and the movie box revenue from Box Office Mojo (http://boxofficemojo.com).

The whole prediction model is consists of two parts. In the first part, we discovered that those movie metadata such like critic and audience scores, casts and file cost can be used to build a effective autoregressive model for predicting film box revenue. Meanwhile, in the second part, we analyzed a sentiment content in the critics and users reviews to study the correlation between the audience review of the film and the performance of the movie box revenue.

3.1 Autoregressive model

We start our research with a case of autoregressive models that only had a few days before revenue data and examine how those factors may influence the model, since the autoregressive model is widely used as a baseline in prediction problem.

The autoregressive model can be used to predict a current statistic based on previous information, which means the value of this function depends on the value of its own previous value. The notation indicates an autoregressive model of order p. The model is defined as:

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$$

where $\varphi_1, \ldots, \varphi_p$ are the parameters of the model, c is a constant, and ε_t is white noise. In this case, suppose Xt indicates the film revenue on one day t. Our goal is to build an autoregressive model for the time period. Here, we analyze the model by defining various features as parameters. There are mainly four types of features: rating features representing rate information about a film, where we use audience scores, cast features representing all the actress and actors shown in the film, staff features representing crew members behind scene such like director and producer. The detail features are illustrated as follows.

A. Rating Feature

This paper use audience scores from rotten tomatoes.com as a rating feature by comparing the rating scores and corresponding movie box revenue on the Box Office Mojo. First we collected the Top 100 2016 DOMESTIC GROSSES movies from Rogue One: A Star Wars Story to Kevin Hart: What Now? and their rating scores. We found 89% of these movies has a very high rating scores above 85% tomato meter. We also collected 50 movies whose tomato meter are less than 60%(rating number more than 1000) and obviously their box office were bleak. 96% of them have a revenue under 1 million. So we give this feature value four regions: those tomato meter above 85% has value 1, range from 75% to 85% has value 0.75, range from 60% to 75% has value 0.5, while value 0.25 is for those rating below 60%.

B. Cast Features

We use 1-K vector to represent an actor, where actors have their own dimension and others are null. The sum of the actor vectors shown in the film calculate as an cast feature vector of this movie. Normally if we considering each actor as a word, this is a

bag-of-words (BoW) model of the cast. In previous research[6], there are 14,767 unique actors in our data collection, including 668 different movies. According to our previous experiments, in this paper we won't use pairs of actors (co-occurrence) as a feature. Considering producer and film maker will try their best to avoid the film repeat, there would rarely have the situation that same actors appear in different movies, and double-related information learning is not ideal in this project. What's more, we also propose a classify system for actors: for those leading stars like Chris Evans, Robert Downey, and Jennifer Lawrence, they would have a high vector value in cast feature.

C. Staff Features

Considering all other features involved in the film, this paper also came up with a idea by using five detailed factors: (i) director, who decides the actors listing and controls the whole film schedule, (ii) scriptwriter, who writes the movie script, (iii) film company, who invest the film, (iv) author, book writer on which the film is based, and (v) singer, who is responsible for singing the movie theme song.

3.2 Sentiment Analysis

A. Optimized model

Many previous research have been studied in sentiment analysis. So in this part, we choose to improve Sentiment Probabilistic Latent Semantic Analysis (S-PLSA), proposed by X.Yu [7]. In S-PLSA model, the dataset of a specific film has two sets data, reviews R={r1,r2,…rm} and key words W={w1,w2,…wn}. Therefore its mathematical formula can be composed as a M×N matrix, where element (i,j) is the number of times of word i in a online review j.

Considering S is the set of all sentiment factors. Follow the S-PLSA model, the probability for word-pair (r, w) can be formulated as:

$$pr(r,w) = \sum_{s \in S} pr(s)pr(w \mid s)pr(r \mid s)$$

As shown in the model, the probabilities is firstly initialized and then we use Expectation-Maximization (EM) algorithm as a estimated method to compute a maximum value of the log likelihood function.

Combined with the autoregressive model described in the previous section, in this part, the whole model optimized by adding sentiment factor. We use n_t as the number of audience reviews for a movie on a specific day t. And $\varphi_{i,j,k}$ denotes the probability of the k_{th} factor on the j_{th} review on a specific day t. Then we have $\varphi_{i,j,k} = Pr(z = k \mid t, j)$, hence for a sentiment factor k the average probability at that day t is given as:

$$\omega_{t,k} = \frac{1}{n_t} \sum_{j=1}^{n_t} pr(z = k \mid t, j)$$

In this project k is chosen to be 5, the same as a kaggle Sentiment Analysis competition, where 0 is negative,1 is somewhat negative, 2 is neutral,3 is somewhat positive, and 4 is positive. So the total autoregressive model can be optimized as:

$$\sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \sum_{k=1}^{K} \rho_{i,k} \omega_{t-i,k} + \varepsilon_t$$

B. Random Forest Classifier

In the later research, this paper uses a random forest classifier, which is an integrated learning method for classification, regression, and other tasks is performed by constructing multiple decision trees at training time and outputting patterns as

categories (categories) or improving the predicted classes of the trees. In this project we use random forest classifier with a concatenation of 3 kinds of features:

- The decision functions of set of vanilla SGDClassifiers trained in a one-versus-others scheme using bag-of-words as features.
- The decision functions of set of vanilla SGDClassifiers trained in a one-versus-others scheme using bag-of-words on the wordnet synsets of the words in a phrase.
- The amount of "positive" and "negative" words in a phrase as dictated by the Harvard Inquirer sentiment lexicon

During prediction, we also check for duplicates between the training set and the train set, which are crawled from rotten tomatoes.com. After collecting the sentiment value from the dataset, we add this feature as an additional variable to the autoregression model on the box revenue, which is described in the previous part.

4. Results

4.1 Experiment for Autoregressive Model

In the first part, this paper used linear support vector regression (SVR) for prediction model since almost all the value of these features is either 0 or 1, and it's obvious that the accuracy of this model is better than using a radial basis function kernel [6]. And all the necessary movie metadata are crawled through rotten tomatoes API. Also we applied Liblinear as the toolkit for regression and the prediction accuracy is calculated by using the cross-validation method. The process is as follows.

1) Select Top-grossing 650 movies from Box Office Mojo and sort in time series order.

2) Create a basic regression model of the film $d1$, ..., dt and predict the box revenue for dt+1, where t indicates different time.

3) Repeat Step 2 for t = p...,n−1, which means here we should have at least p number of movies to build a model and n is the total number of films we have.

4) Caculate the Mean Squared Error (MSE) between the predicted revenue by our model and outcome in the real world from dp+1,…,dn.

Here, we define p as 60. For the sake of finding an optimal cost parameter for SVR, we firstly looker for the range from 2^{-5} to 2^{11} with a step size of 2^2, and later the range of 2^{-4} to 2^3 with a step size of $2^{0.2}$. Finally we would choose the cost value with a highest performance.

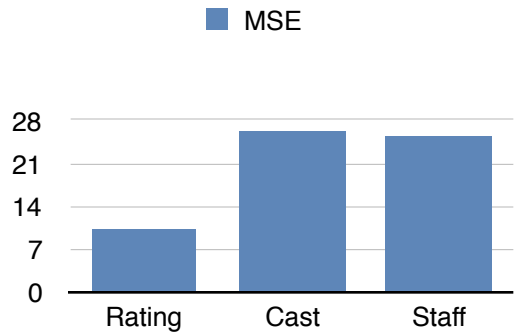

Fig. 1. Mean Squared Error (MSE)

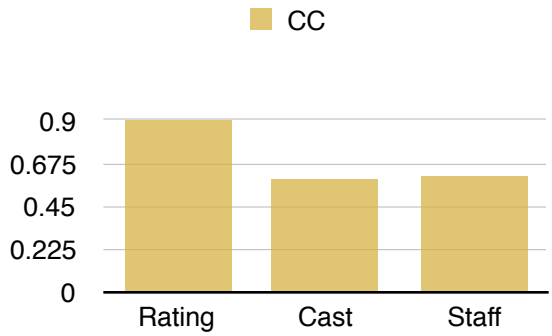

Fig. 2. Pearson correlation coefficient (PCC)

Results are shown in Figures 1 and 2, where Mean Squared Error and the Pearson correlation coefficients between the real-world outcome and predicted revenue by using different features. Note that these parameters associated with each feature were optimized to minimize MSE. It's clear that the rating feature was the best among these features. Therefore we may combine three features by adding other two features to the rating feature.

4.2 Experiment for Sentiment Analysis

In the study, a Random Forest classification model is applied with a concatenation of 3 kinds of features, and dataset is a set of audience review from rotten tomatoes.com. Although we crawled the movie metadata from the rottentomatoes, all the dataset is not labeled. So we decided to use the review dataset downloaded from a Kaggle competition as a training dataset, which is a data corpus of movie reviews for sentiment analysis. In their work on sentiment treebanks, Socher et al. [8] used Amazon's Mechanical Turk to create fine-grained labels for all parsed phrases in the corpus. Then we use random forest classifier training the dataset. After getting the result, we add sentiment feature into previous autoregression model, and figure out its coefficient variable.

Finally we compute Mean Absolute Percentage Error(MAPE) to evaluate the accuracy of this model comparing with the true value, where n is the total number of prediction revenue values resulted by the model:

$$MAPE = \frac{1}{n}\sum_{1}^{n}\frac{|pred_i - real_i|}{real_i}$$

Sentiments analysis result shows the accurate rate of prediction is about 0.65844 by using the random forest classifier. Unfortunately for the lack of time, although this paper propose a MAPE method to measure the whole project, we didn't implement this part of experiment.

5. Reference

[1] Al Boni, Mohammad, and Matthew S. Gerber. "Predicting crime with routine activity patterns inferred from social media." Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. 2016.

[2] Wicaksono, Andy Januar. "A proposed method for predicting US presidential election by analyzing sentiment in social media." Science in Information Technology (ICSITech), 2016 2nd International Conference on. IEEE, 2016.

[3] Asur, S. and Huberman, B.A., 2010, August. Predicting the future with social media. In Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on (Vol. 1, pp. 492-499). IEEE.

[4] Yao, Rui, and Jianhua Chen. "Predicting movie sales revenue using online reviews." Granular Computing (GrC), 2013 IEEE International Conference on. IEEE, 2013.

[5] Yassine, Mohamed, and Hazem Hajj. "A framework for emotion mining from text in online social networks." Data Mining Workshops (ICDMW), 2010 IEEE International Conference on. IEEE, 2010.

[6] Fukushima, Yusuke, Toshihiko Yamasaki, and Kiyoharu Aizawa. "Audience Ratings Prediction of TV Dramas Based on the Cast and Their Popularity." Multimedia Big Data (BigMM), 2016 IEEE Second International Conference on. IEEE, 2016.

[7] X. Yu, Y, Liu, J.X Huang and A. An., “Mining Online Reviews for Predicting Sales Performance: A Case Study in the Movie Domain” IEEE Transaction on knowledge and data engineering, Vol. 24, No. 4, April 2012, pp.720-734.

[8] Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Chris Manning, Andrew

Ng and Chris Potts. Conference on Empirical Methods in Natural Language Processing (EMNLP 2013).