Machine learning on predicting gross box office

Pengda Liu

Dec 2016

1 Introduction

In recent years, the movie market has been growing larger each year.This industry generates approximately billions dollars of revenue annually[1]. The question of what makes a film successful has been asked for over the years, not just by you and me:large companies like Twenty-First Century Fox,Universal Studios award million prizes to those who can improve their recommendation and prediction algorithms. This project attempts to address this question by using machine learning techniques to predict film box office.

2 Related work

There has been few similar attempts in this field. Jason in his project[6], implemented linear regression without a success(only 40% predictions are within 100% error from the real value). The key problem in his project is that there is no feature roughly linear with the gross box office. In our project's linear model, we included the opening weekend box office feature, which we can see from this figure(for each data point the y- axis represents opening weekend and the x-axis represents the gross office), is suitable for running linear regression, when combined with other features. As a result, in my project, the prediction linear regression is much more successful. The genre features for classification is not suitable for direct input into SVM ,in Ericson's project[7], he grouped the films by genres and run SVM within each genre. However, this may be a waste of data, we directly incorporate the genre features in our model using Naive Bayes,which is further explained in the discussion section.

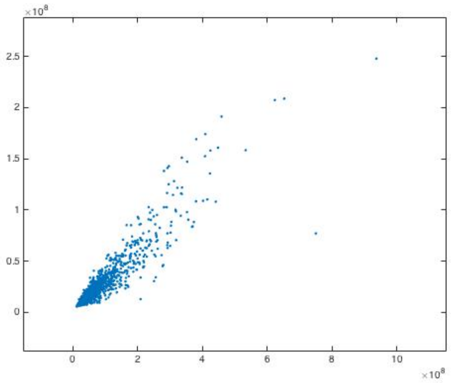

Figure 1: Opening weekend and gross office

3 Data Collection

Because there is no existing data that matches my desire ,the data is from mixed sources: (1)Existing open source data sets from online.(See [1] and [2] for details) (2)Web-crawled(done with a python web-cralwer tool called scrapy) data from Bing search and Youtube. Then I merge and organized the data for different models.Since there is slight difference for the features for each model. For simplification,I let

A={star power, number of opening theatres, Youtube trailer views,IMDB rate, budget}.

Star power is the sum of the likes on the homepage of the fist three actors/actresses.

All movies are from year 2004-2015 because people's taste for movies change and it is probably wise to only include movies from only relatively recent era. One thing to note that since some movies have limited opening policies, I only used the top100 opening weekend box office from each year.

4 General approach description and feature selection

It is divided into two parts, one is using linear models and involving opening weekend box office(it may not serve as a feature, see more detail in linear models section) outputting an exact number. Another is the classification part excluding opening weekend box office and used a modified version of SVM and neural network to output the category indicating rough range of the gross box office. Due to the limited number of data points(the number of films produced from 2004-2015 definitely does not count as "many" in terms of a machine learning project),we used 1200 examples as training size and 400 examples as test size.

5 Linear Models

Implementation: Matlab
Model description
(1)Naive linear model.
Input features:$A\bigcup\{$opening weekend box office$\}$.
Output=gross office.
This method uses opening weekend box office as a direct input, thus the scale of the data varies from below 10(imdb rate) to millions(opening weekend office), we normalize our data using the following formulas:
1. Let $\mu = \frac{1}{m}\sum_{i=1}^{m} x^i$. 2. Replace each x^i with $x^i - \mu$.
3. Let $\sigma_j^2 = \frac{1}{m}\sum_i (x_j^i)^2$. 4. Replace each x_j^i with x_j^i/σ_j
(2)Modified linear regression.
Input features=A.
Output=$\frac{\text{gross office}}{\text{opening weekend office}}$
(3)Locally weighted linear regression.
Input features=A.
Output=$\frac{\text{gross office}}{\text{opening weekend office}}$
For (2) and (3),let y_1 be gross office and y_2 be opening weekend office, let h be the true value of $\frac{y_1}{y_2}$. Suppose that we are have output $h_\theta(x)$, then we simply predict the gross box office to be $h_\theta(x)y_2$. Then we have:

$$\frac{|y_1 - h_\theta(x)y_2|}{y_1} = \frac{|hy_2 - h_\theta(x)y_2|}{hy_2} = \frac{|h - h_\theta x|}{h}$$

Thus the error in (2) and (3)(the ratio error) also stands for the error of the actual error in predicting gross office,thus we are judging the performance of (1) (2) (3) in the same way.
Math formula
Since our training set is not huge, we simply use normal equations to fit our θ:
Linear regression:

$$\theta = (X^T X)^{-1} X^T y$$

Locally weighted linear regression:

$$\theta = (X^T W X)^{-1} X^T W y$$

Result

	e_1 train	e_1 test	e_2 train	e_2 test
(1)	23.30%	27.07%	45.77%	47.50%
(2)	20.1%	20.3%	43.22%	44.1%
(3)	19.38%	20.03%	24.3%	25.7%

We define:

$$e_1 = \frac{1}{m}\sum_{i=1}^{m} \frac{|h_\theta(x^i) - y^i)|}{y^i}$$

$$e_2 = \frac{1}{m}\sum_{i=1}^{m} 1\{\frac{|(h_\theta(x^i) - y^i)|}{y^i} > 0.3\}$$

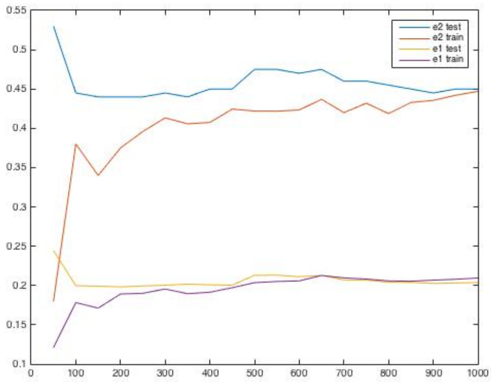

Figure 2: Learning curve for (2)

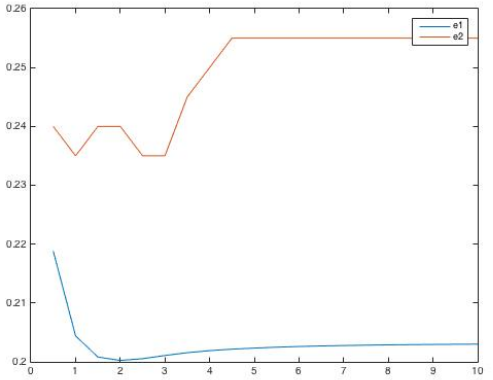

Figure 3: Curve of error rate and learning rate τ

as the two criteria for error, where x_i stands for the ith data point. e_1 is just the error rate in usual sense, e_2 gives us a sense of how much percent of examples do we predict with more than 30% error.(We here set 30% as a threshold and other threshold is also acceptable.)
For locally weighted linear regression, we use hold out cross validation to experiment with the learning rate τ and get the following curve of performance. We can see that
arg $\min_\tau e_1 = 2$, arg $\min_\tau e_2 = 3$.

6 Classification

For better performance and ease of implementation, we separate film box office into 10 categories as follows:

Table 1: Categorization

Million	<20	20-40	40-60	60-80	80-100	100-120	120-140	140-160	160-180	>180
Category	1	2	3	4	5	6	7	8	9	10

6.1 SVM with Naive Byes

Implementation: Using scikit-learn in python and Matlab.
There are 22 genres on IMDB website, as a result, the genre feature for film i is the 22 dimensional binary valued vectors $v_i = (v_{i1}, v_{i2}, ..., v_{i22})$ such that:
$v_{ij} = 0$,if film i does not have genre j.
$v_{ij} = 1$,if film i has genre j.

Note that the genre features are binary-valued,different from the continuos/integer valued features like star power,youtube trailer views, etc. So we first run the training set using Naive Bayes with only genre feature and then we run the SVC algorithm on scikit-learn[4] with feature set A.
For a film i, output

$$\arg\max_j(\log p_j + \log q_j - \log P(j)).$$

Where $\log p_j$ is the log probability of film i being in category j from Naive Bayes model, and $\log q_j$ is that from the "predict_log_proba" function from scikit-learn.

$$P(j) = \frac{\text{number of movies in category j}}{\text{total number of movies}}$$

.
Train error: 29.43%. Test error:30.12.%

6.2 Neural network

Implementation: Using scikit-learn in python.
We used the MLPClassifier from scikit-learn ,which implements a multi-layer perceptron (MLP) algorithm that trains using Backpropagation.We use (5,3) hidden layers with sigmoid neuron and lbfgs solver.
Train error:23.41%. Test error:25.03%.

7 Discussion and analysis

Linear models

We see that in the linear models section,(2) and (3) does a much better job than (1), this is due to, in part, that by outputting $\frac{\text{gross office}(y_1)}{\text{opening weekend office}(y_2)}$, we help to negate the the influence of inflation,population:suppose that the influence factor is λ ,then $\frac{\lambda y_1}{\lambda y_2} = \frac{y_1}{y_2}$.

Moreover, when finally outputting gross office, we let $y_1 = h_\theta(x)y_2$, which is different from simply including y_2 then run linear regression. This gives y_2 much larger "weight" in our prediction.

Another thing to note is that, for usual linear regression,when looking at the output error table for the test examples,I note that the data fluctuates in a very big range. Sometimes the error rate is amazingly low(below 3%), sometimes, it's very high(above 80%) ,with a standard deviation of 0.249.
However, locally linear regression produces an error with only 0.146 standard deviation.
This agrees with the fact that the e_2 for locally weighted linear regression is much lower(by almost twice) than the linear regression because the locally weighted linear regression is more stable and the output is more smooth. Thus the majority of the predictions are within a reasonable error range. But one big flaw is that locally weighted linear regression is more computationally expensive.

Classification

For our modified version of SVM, we suppose that y stands for a category, x stands for the random variables A and v stands for the genre feature.We assume that genre feature and set A are independent .Then:

$$p(y|x,v) = \frac{p(x,v|y)p(y)}{p(x,v)} = \frac{p(x|y)p(v|y)p(y)}{p(x)p(v)}$$
$$= \frac{p(x|y)p(y)p(v|y)p(y)}{p(x)p(v)p(y)} = \frac{p(y|x)p(y|v)}{p(y)}$$

Which gives: $\log p(y|x,v) = \log p(y|x) + \log p(y|v) - \log p(y)$.
We then score each one using $\log p(y|x) + \log p(y|v) - \log p(y)$ to make a prediction.It is interesting to note that linear kernel is the only choice that converges well and produces reasonably good result.

8 A future direction

Feature improvement

I would include features directed related to the plot:the dialogue,story key words,etc. Also, background music, promotion budget would also be good additions, but unfortunately ,I was unable to

obtain these data.If anyone happened to have any of these resources(e.g. know someone working in the film industry), I would be happy if you contact me.
Model improvement:
Maybe we can combine SVM and linear regression together in some way.(For example, first let SVM predict a category then we use that category information as an input into linear regression.)Because scikit-learn could not produce the probability score for MLP classifier, I was unable to combine Naive Bayes with neural network.If I had more time, I would try to learn some deep learning models whose knowledge I do not possess currently.

9 Conclusion

Through this project, we can see that predicting gross box office is indeed a challenging topic. I have to admit that currently, my model and result are still not good enough to be put into actual commercial use. But I think that with more advanced knowledge of machine learning and better data set, we can reduce the error significantly in the future.

References

[1] http://www.boxofficemojo.com/yearly/

[2] https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset

[3] A Comparison of Methods for Multi-class Support Vector Machines,$Chih - WeiHsuandChih - JenLin$

[4] http://scikit-learn.org/stable/

[5] Neural network and deep learning http://neuralnetworksanddeeplearning.com/chap1.html

[6] Predicting movie box office gross http://cs229.stanford.edu/proj2013/vanderMerweEimon-MaximizingMovieProfit.pdf

[7] A Predictor for Movie Success http://cs229.stanford.edu/proj2013/EricsonGrodman-APredictorForMovieSuccess.pdf