Machine learning on predicting gross box office

{Pengda Liu} pengda@stanford.edu

Problem

In this project I explored how several film parameters would help predict the gross box office. It is divided into two sections, 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.

Data and features

Data is from mixed sources: (1)existing open source data sets from online.(See [1] and [2] for details) (2)web-crawled 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, 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.

SVM with Naive Bayes

For both SVM and neural network, we categorize our data as follows. Input features:$A \cup \{\text{genre}\}$.

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

There are 22 genres on IMDB website, as a result, 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.
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.$P(j)$ is just $\frac{\text{number of movies in category j}}{\text{total number of movies}}$.
Train error: 29.43%. Test error:30.12.%
Compared to the grouping method in [7],we directly incorporate the genre features in our model using Naive Bayes,which is further explained in the discussion section.

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] http://cs229.stanford.edu/proj2013/EricsonGrodman-APredictorForMovieSuccess.pdf

Linear Models

(1)Naive linear model.Input features:$A \bigcup$ {opening weekend box office}. Output=gross office.
(2)Modified:Input features=A.Output=$\frac{\text{gross office}}{\text{opening weekend office}}$
(3)Locally weighted linear regression. Input features=A. Output=$\frac{\text{gross office}}{\text{opening weekend office}}$
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: $\theta = (X^T W X)^{-1} X^T W y$.
Error:

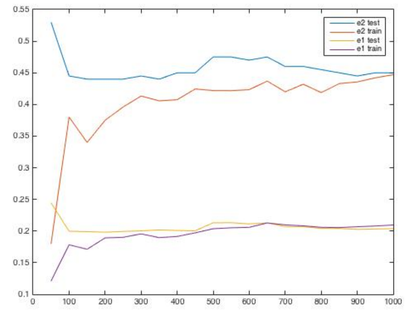
	e_1 train	e_1 test	e_2 train	e_2 test
(1)	23.30%	27.07%	45.77%	47.50%
(2)	20.1%	20.3%	43.22%	44.1%
(3)	19.38%	20.03%	24.3%	25.7%

We define:

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

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

as the two criteria for error, where x_i stands for the ith data point.



Learning curve for (2)

Neural network

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%.

A Future Direction

Feature improvement: I would include features related to 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.
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.)If I had more time, I would try to learn some deep learning models whose knowledge I do not possess currently.

Discussion and analysis

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.

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.Then:

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

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

We then score each one using $\log p(y|x) + \log p(y|v) - \log p(y)$ to make a prediction.