Prediction from Blog Data

Aditya Parameswaran Eldar Sadikov Petros Venetis

1. INTRODUCTION

We have approximately one year's worth of blog posts from [1] with over 12 million web blogs tracked. On average there are 500 thousand blog posts per day. In this project, we are attempting to extract from the blog data the set of features that are predictive of the movie gross sales, critics ratings, and viewers ratings (collected by sites like [3]). Having gained this insight, we are applying machine learning techniques to make predictions on sales and ratings for future movies.

This is useful for example, to see if the 'buzz' surrounding a movie is sufficient to obtain high sales, or whether additional marketing is required. In addition, we could automatically figure out ratings of movies based on their mentions in blogs. This could be useful for movie rating sites like Rotten Tomatoes [3] and IMDB [4], either to verify the correctness of ratings, or to 'seed' ratings for a new movie or a movie that did not exist in their database. Additionally, in some instances information like gross sales may not be publicly available, so we would like to make a prediction instead. At a high level, we would like to see if the "online chatter" is useful for analysis and prediction of the quality and popularity of various items, be it movies, songs, books, restaurants or other commercial products.

In our project, for the top movies of 2008 (≈ 200 of them that have non-negligible blog mentions) and top movies of all time (≈ 250), we try to predict the following: the gross sales, the critics rating and the average viewers rating. We selected a long list of relevant features and populated them after parsing the blog data. After performing training, cross-validation, feature and model selection with a primitive set of models, we have reasonable error rates for most output variables. We find that Naive Bayes and SVM are the best prediction algorithms for our data, and PCA generally works best as a feature selection method. Finally, we have observed some interesting patterns that suggest that quite accurate predictions can take place, given more sophisticated algorithms.

2. DATA AND FEATURES

We have collected titles of the top 300 Box Office movies of 2008 from [2] and the top 400 U.S. Box Office movies of all time from [4]. We have then hand-filtered both lists for the titles that were similar to the common English phrases. This has effectively narrowed down our lists to the 197 top movies of 2008, which we are going to refer to as *new movies*, and the 273 top movies of all time, which we are going to refer to as *old movies*.

Due to the sheer size of the original data set (approximately, 1.5 TB), we had to do parsing in multiple stages. At the first stage, we eliminated posts that were non-English and posts that contained more than 30 links. The former was done because our focus was on the U.S. movies, whereas the latter was done for the purpose of filtering spam. After we performed this step, we constructed a list of regular expressions corresponding to the titles of the new and old movies and ran it against the data set to aggregate the posts that mention movies. This has effectively become our working data set (which was of $\approx$ 15 GB size), from which we have extracted our features.

The features that have been extracted can generally be classified into four categories where each category is described in detail in the following subsections:

- Basic features that quantify movie mentions without regard for the quality, sentiment, or time of the blog posts where they are made
- Features that respect only mentions made within a time window before or after a movie release date
- Features that address the spam issue
- Features that respect only positive sentiment mentions

2.1 Basic Features

We have hypothesized that the importance of a movie mention in a post is proportional to whether it occurs in the title or the text of the post as well as the rank and the indegree of the blog where the mention is made[1]. Accordingly, we have devised our initial 5 features as follows:

1. Number of movie mentions in the title or text of blog posts
2. Number of movie mentions in the title of blog posts
3. Number of movie mentions limited to only top ranked blogs ($\approx$ 4000 blogs ranked)
4. Number of movie mentions weighted by blog rank where weights are equal to $1/\ln(rank)$ or 0 for non-ranked blogs
5. Number of movie mentions weighted by blog in-degree where weights are equal to $\ln(indegree)$ or 0 for $indegree < 3$

Although the weights used for feature 4 and 5 are quite intuitive given our hypotheses, they may not be as effective

[1]Both blog rank (which measures blog popularity) and blog indegree were calculated by http://spinn3r.com/

as the weights determined automatically by a learning algorithm. Hence, in addition to the features above, we have discretized movie mentions by the ranking tier of a blog in which they are made into 20 features as follows:

1. Number of movie mentions in the blogs ranked 1-10
2. Number of movie mentions in the blogs ranked 11-20
...
10. Number of movie mentions in the blogs ranked 91-100
11. Number of movie mentions in the blogs ranked 100-200
...
19. Number of movie mentions in the blogs ranked 901-1000
20. Number of movie mentions in the blogs ranked 1001 and above (but excluding non-ranked blogs)

Finally, in addition to the features that quantify movie mentions, we have included among our features general characteristics of a movie as collected from [2]:

- Genre (discretized to be an integer $\in \{1, \cdots 10\}$)
- Budget
- Distributor (discretized to be an integer $\in \{1, \cdots 10\}$)

2.2 Times Series Features

Most of the box office sales are made within the first few weeks after the movie release date and the success of the movie is often highly dependent on the promotional campaign prior to the release date. Hence, we have hypothesized that the movie "hype" in online chatter just before and right after the release date should in general be a good indicator of the box office sales and public ratings. Accordingly, we have discretized movie mentions in a time series features as follows:

1. Number of movie mentions 5th week (35-28 days) before the release date
2. Number of movie mentions 4th week (28-21 days) before the release date
...
6. Number of movie mentions 1st week (0-7 days) after the release date
...
10. Number of movie mentions 5th week (28-35 days) after the release date

2.3 Features that address spam issue

Although we filtered a large number of spam posts during parsing by discarding those that contained more than 30 links, we have still ended up with many spam posts in our working data set. We observed that the majority of spam posts were either very short in length (usually a sentence or two with a link) or quite lengthy with many HTML tags and images embedded in their content. Hence, we decided to employ the following heuristic for filtering these spam posts: if a post is either less than 200 characters long or contains on average less than 20 characters of text in between the HTML tags, then it is a spam post. To avoid the risk of false positives (non-spam posts classified as spam), instead of employing this heuristic across all features, we have decided instead to add a set of additional features that relied on this technique:

1. Number of mentions in the title or description of non-spam posts
2. Number of mentions in the title of non-spam posts

3-13. Times series features that respect only non-spam posts

Since ranked blogs are unlikely to contain spam posts, we have not included features that respected blog ranking among the new features.

2.4 Sentiment Features

A sheer number of movie mentions in blog posts may not always be indicative of high viewers' ratings or high box office sales and it seems intuitive to consider the sentiment of the posts when trying to predict ratings or sales. Since a movie mentioned in a post may not always be a central theme of the post, to measure the polarity of a movie mention, we focus only on the 5 sentences surrounding the movie title in the text of a blog post. To determine a movie mention's sentiment, we employ the hierarchical classification approach described in [6] using LingPipe classifiers [5]. Specifically, we use a subjectivity classifier to select from the 5 sentences only those ones that meet a subjectivity threshold[2] and then classify those for sentiment using a polarity classifier. Both LingPipe subjectivity classifier and polarity classifier are 8-gram language model classifiers with the former trained on the IMDB [4] plot summaries and Rotten Tomatoes customer reviews [3] and the latter trained on the full text IMDB movie reviews (described in more detain in [5]).

We have found that LingPipe polarity classifier is very conservative when determining the sentiment and tends to assign negative classification more frequently than the positive one. Thus, we have decided to add two sets of features: one associated with a conservative assessment of sentiment, where we give credit to only those posts that LingPipe classifies as positive and, one associated with a more aggressive assessment of sentiment, where we give credit not only to the posts classified as positive but also to the posts classified as negative with a very low confidence[3]. For each of these sets we have added the time series features (e.g., number of movie mentions with positive sentiment in the first week after the release date) and the rank tiers features (e.g., number of movie mentions with positive sentiment in blogs ranked 20-30).

2.5 Output Variables

For both new and old movies, we have collected the following output variables from [3] and [2]:

- Average rating of critics
- Average rating by users (viewers)
- Gross box office sales (2008 only for new movies and all time for old movies)

3. MODEL

As described in the previous section, we have a set of input features (a total of 118) and a set of output variables (3) that we would like to predict. As a first step, we discretized the output variables to 10 buckets, or deciles. We wish to use classification to predict which decile the movie lies in

[2]The probability of a sentence being subjective estimated by the classifier needs to be at least 0.7.

[3]We set the threshold level for aggressive sentiment to be the 0.05 cross-entropy between positive and negative sentiment probabilities

for the given output variable, given the values of the input features. We consider a classification successful if we are able to predict the right region in which the movie lies (i.e. we say that the prediction is 'correct' if we predict the correct decile by +/-2, e.g., if the actual value is 4, then a correct classification would be any of the deciles {2,3,4,5,6}). The algorithms we used for prediction were the following:

1. Naive Bayes: The input features are discretized into 10 buckets, and the probability of the output variable being in any given decile given the values of the input variables are calculated. The 'best' decile is picked.
2. Linear Regression: The input features are fed into the linear regression framework, and the value of the output variable is returned. We say that the prediction is correct if the value of the output variable lies in +/- 2 deciles of the actual decile.
3. Locally Weighted Linear Regression: Similar to Linear Regression, except that we chose an appropriate weighting function[4].
4. Multiclass SVM: We trained our data to create hyperplane classification boundaries for each 'class' - i.e. decile. This allowed us to say if a movie is in a particular decile or not. If a movie was classified into many deciles, we picked the decile hyperplane such that the distance to the hyperplane is the maximum among all deciles for which the movie was classified into. If a movie is not classified into any 'bucket', then we arbitrarily output the value '5', i.e. that the movie is average w.r.t. the output variable.

In addition to the forementioned machine learning techniques, we have also experimented with Softmax regression. However, the observed performance was quite poor and decided not to use it among our algorithms.

Due to the small size of the training data ($\approx$ 200 new movies and $\approx$ 300 old movies), we couldn't use the original set of features we had extracted (118 total), which prompted us to look for a feature selection technique. On the initial set of 118 features, we measured the training and test error and found the test error to be much greater than the training error, which prompted us to reduce the dimensionality of the data, given that we cannot generate more training examples - our list of movies is the largest available list on the web [2]. In addition some of our features could be noisy (e.g. sentiment analysis), which makes feature selection even more crucial. We wish to only keep the most important features for training, and remove the ones that are seemingly 'random'. Hence, we had three different ways to select features for each output variable:

1. KL-Divergence: We picked 15 features that the output variable has the least KL Divergence with. This indicates that the output variable has low entropy given the input feature.
2. Correlation: We picked 15 features that are most correlated to the output variable.
3. PCA: We performed Principal Component Analysis on the data and picked the top 15 eigenvectors. We then projected all the columns onto those eigenvectors, thus reducing the dimensionality of the matrix.

[4]We experimented with various functions, and chose $w(x,y) = 1 - \frac{norm(x,y)}{max_z norm(x,z)}$ for both the old and the new movie set

Given the three feature selection methods and 4 prediction techniques, we ended up with $4 \times 3 = 12$ models. After separating the training from the test data (85 % for training), we proceeded to perform 10-fold cross-validation with the training data. For each output variable, we evaluated the average error for each of the 12 models over the cross-validation sets and then for each output variable picked the model that performs best 'on average' (i.e. we picked the least $\Sigma\widehat{\epsilon}_{S_j}(h_{ij})$, i.e. the estimated generalization error).

4. RESULTS

The estimated generalization errors for each of the 12 models is given in Table 1 for the new movies and Table 2 for the old movies. Table 3 shows the best model selected for each of the output variables, where Training Error column shows the estimated generalization error obtained with the best model. For each output variable, we then train a selected model (which includes both the algorithm and the feature selection method) on the entire training set, and test it on the test set. The test error obtained as a result for each output variable is shown in the last column of Table 3.

Naive Bayes and SVM have turned out to be the more effective algorithms for most of the output variables and PCA has been consistently effective as a feature selection method. On the other hand, Linear Regression has performed poorly for most of the variables, which may indicate that the output variables as functions of features are not linear. Nonetheless, no selected model has performed extremely well on a test set for any of the output variables. This can be explained by any of the following: low number of training examples, inherently noisy data set, poor sentiment analysis, inter-dependence between movies released in the same time period.

As seen from Table 3, for some output variables (e.g., Critics Rating for the new and old movies and Gross for the old movies), the training error is very close to the test error. However, for some of the other variables (e.g. 2008 Gross and User rating for the new movies) the difference between the training error and the test error is still high. Thus, we hypothesize that the results could be improved by selecting a better model, or by including more features (for example, by employing a better sentiment analysis approach). In addition, note that the value of the test error for the last two rows of Table 3 is smaller than the training error. This might be due to the fact that we have a limited number of examples, and the initial split of data into training and test examples may have given rise to a large number of the "highly predictable" movies falling in the test set.

Our work demonstrates that blogs could be a very good source of information for analysis and prediction of movie quality and popularity. In spite of the relatively poor performance of the selected models on the test set, we have indeed observed interesting patterns that suggest predictive power of blogs. Specifically, Graph 1 shows that the number of movie mentions is directly related to the movie gross. Graph 2 shows correlation of 2008 Gross to the times series features, i.e. mentions in the 5th week before the release date, 4th week before the release date, ..., 5th week after the release date, and demonstrates that the closer we are to the release date, the more correlated number of mentions are to the actual gross (the same applies to the other output variables). Finally, Graph 3 shows correlation of 2008 Gross to the rank tier features, i.e. mentions in blogs ranked

Table 1: Estimated generalization error for each of the 12 models and 3 output variables on the *new* movies (1st row: gross sales, 2nd row: critics rating, 3rd row: user ratings)

LR-KL	LR-CL	LR-PC	WR-KL	WR-CL	WR-PC	NB-KL	NB-CL	NB-PC	SV-KL	SV-CL	SV-PC
0.6866	0.7137	0.7151	0.6804	0.7137	0.7269	0.3174	0.3113	0.2626	0.4774	0.4522	0.4618
0.5290	0.5051	0.5058	0.5628	0.5178	0.5369	0.4916	0.4751	0.3612	0.3984	0.4337	0.3893
0.3768	0.4492	0.4388	0.2961	0.2987	0.4376	0.4379	0.4012	0.3669	0.3302	0.3226	0.3528

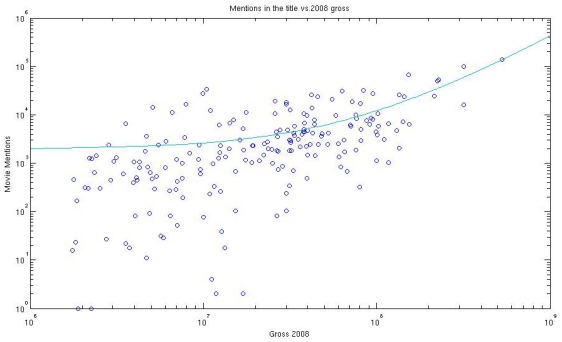
Table 2: Estimated generalization error for each of the 12 models and 3 output variables on the *old* movies (1st row: gross sales, 2nd row: critics rating, 3rd row: user ratings)

LR-KL	LR-CL	LR-PC	WR-KL	WR-CL	WR-PC	NB-KL	NB-CL	NB-PC	SV-KL	SV-CL	SV-PC
0.6968	0.6293	0.6995	0.6843	0.6173	0.6953	0.4638	0.5015	0.4982	0.5253	0.5476	0.5228
0.7101	0.6666	0.7094	0.6927	0.6118	0.7137	0.4742	0.4362	0.4412	0.4473	0.4340	0.4525
0.7490	0.8011	0.7399	0.7537	0.7936	0.7355	0.4754	0.4320	0.4344	0.4356	0.4596	0.4313

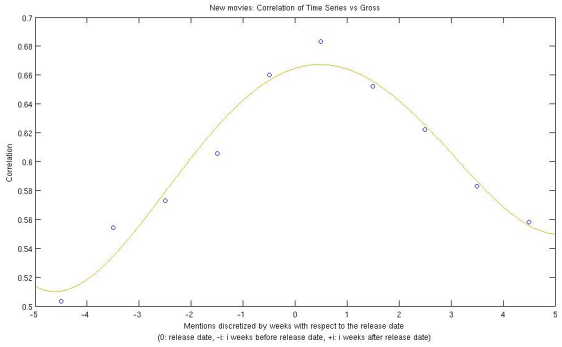
Table 3: Estimated generalization error and test error for each variable's selected model

	Output Variable	**Best Model**	**Training Error**	**Test Error**
New Movies	2008 Gross	NBayes-PCA	26.26%	36.66%
	Critics Rating	NBayes-PCA	36.13%	40.74%
	User Rating	WLR-KL	29.61%	38.46%
Old Movies	Gross	NBayes-KL	46.38%	47.50%
	Critics Rating	SVM-Corr	43.40%	37.50%
	User Rating	SVM-PCA	43.43%	31.58%

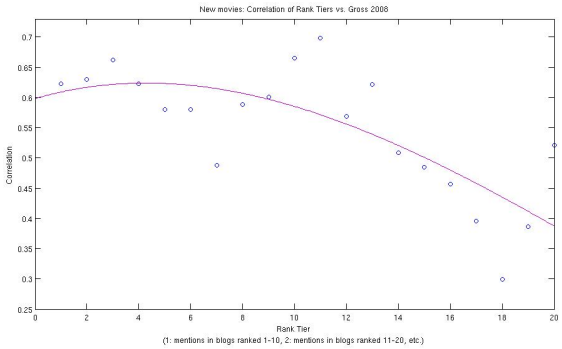
Graph 1: Number of movie mentions in the title vs. 2008 Gross



Graph 2: Time Series Feature Correlation to 2008 Gross



Graph 3: Rank Tier Feature Correlation to 2008 Gross



1-10, blogs ranked 11-20, ..., blogs ranked 1000+. From the graph it can be seen that the higher the ranking of the blogs where movies are mentioned is, the more correlated the number of mentions are to the Gross (same applies to other output features).

Given the relationship of times series features to the gross demonstrated in Graph 2, an interesting problem that could be examined is predicting sales of the ith week after the release date given blog posts until the $i-1$th week. We have not tackled this problem in our project but it would be worthwhile to address in the future work.

Acknowledgements

We would like to thank Jure Leskovec and Spinn3r for providing us the data.

5. REFERENCES

[1] http://spinn3r.com/
[2] http://www.the-numbers.com/
[3] http://www.rottentomatoes.com/
[4] http://www.imdb.com/
[5] http://alias-i.com/lingpipe/

[6] Bo Pang and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. ACL Proceedings.