Using Twitter Data to Predict Box Office Revenues

P. Thomas Barthelemy (bartho@stanford.edu)
Department of Computer Science, 353 Serra Mall
Stanford, CA 94305

Devin Guillory (deving09@stanford.edu)
Department of Computer Science, 353 Serra Mall
Stanford, CA 94305

Chip Mandal (cmandal@stanford.edu)
Department of Computer Science, 353 Serra Mall
Stanford, CA 94305

Abstract

We summarize an effort to predict box office revenues using Twitter data. It is hypothesized that an increased number of tweets about a movie before its release will result in increased box office revenue. Our task can be decomposed into two subtasks: the estimation of the frequency of tweets about particular movies and revenue prediction given this frequency.

Keywords: Tweet classification, Naïve Bayes.

Overview and Motivation

The strategy was to first identify the number of tweets about the movie prior to movie opening, and then to use regression to create a model for predicting box office revenue. The former was the more challenging task, and we approached it in three different ways. The most basic way was to count the occurrence of the title in tweets, although there are clear cases in which this is not expected to perform well. Next, we attempted a variant of Naïve Bayes. Finally, we utilized a bag of words model to estimate the frequency of tweets which are about the movie.

We did not use hashtags during classification, as our Twitter dataset is from 2009, before hashtags were commonly used.

Data and Processing

We used two separate data sources: Twitter data and movie reviews from IMDB.

Twitter Data

The Twitter data included a sampling of approximately half a billion tweets over the last 6 months of 2009. Because we wanted to predict revenue for 80 movies and tweets about movies occurred at a rate of 1/100 at best, and usually far less than that, it was not feasible to label tweets for each movie.

We manually examined 10k tweets to label those that were about movies in general—that is, about any movie. This was used to identify the prior probability of a tweet being about any movie, a value used in the Naïve Bayes analysis.

On occasion, we used a *search-labeled* set of tweets by searching for movies for which it was unlikely to mistake the title or keyword for a non-movie reference, and we assumed that this correctly labeled the tweets. For instance, we used "transformers" as an indicator for *Transformers: Revenge of the Fallen* and "inglourious" for *Inglorious Basterds*. Such a classification method was expected to bias our probabilities of movie-specific words—that is, we would expect an overestimated probability of the movie title—and thus was not used for such purpose. Rather, the approximately labeled tweets were used for identifying movie general words (e.g. "movie", "watch", "tonight") or for validating classification.

Initial word counts were performed using grep. The performance of grep was slow, especially since some of our algorithms required searching multiple keywords in a given file. For better performance, we indexed the tweets using Apache Lucene. Direct frequency calculation was performed using this index. Inference was implemented using tweet-by-tweet classification.

IMDB Review Data

We used IMDB for two goals: to identify general attributes for each movie (e.g. opening day, box office revenue) and to observe the probability of generating a particular word in reference to a movie. Concerning this latter point, we assumed that the probability of generating a word in an IMDB review about a given movie was the same as the probability of generating the same word in a tweet about the same movie. 30 reviews per movie were taken from IMDB. Each set including about 10,000 words in total.

Models

We used multiple methods to estimate the frequency of tweets as input to our regression model. The two initial strategies attempt to classify these tweets individually, and the remaining strategies consider instead a particular day as simply a mix of a non-movie-specific bag of words and a movie-specific tweets bag of words.

Title Search

To provide baseline performance, we fit linear regression model using a keyword search, as shown in Figure 1. That is, we simply searched the occurrence of the title (case insensitive) in all of the tweets in the week before their respective opening days.

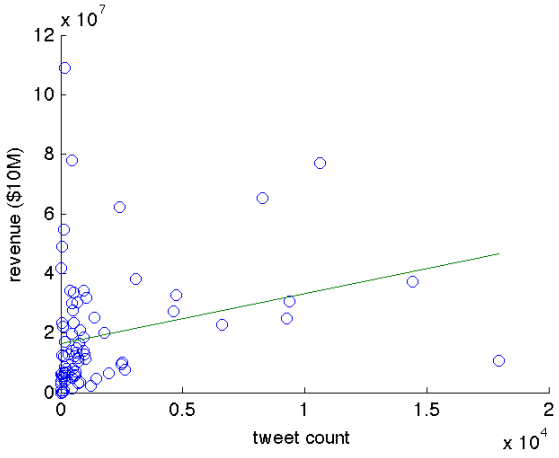

Figure 1: Count of tweets having title words versus movie revenue.

Searching for the movie title is not always a good indicator that the tweet is about a movie. There are two common cases in which this posed a problem. First is the case in which the movie title is long and infrequently mentioned in its entirety. For instance, *The Lord of the Rings: The Fellowship of the Ring* is often referred to as "Lord of the Rings" or even "LOTR". Second is the case in which the title is very short and likely to be contained in tweets that do not refer to the movie. One example of this is *Shorts*, a movie released in August 2009.

Naïve Bayes

We could not use conventional Naïve Bayes for tweet classification because we did not know the prior probability of a tweet being about a specific movie. (If we did, this part of our project would be trivial!) We considered circumventing this problem by decomposing the causal model into one for which the probabilities could be estimated.

Let m_a be the variable representing the tweet being about any movie (M_a) or not about any movie ($\neg M_a$), let m_s be the variable representing a tweet being about a movie (M_s) or not about a movie ($\neg M_s$), and let W represent the generation of a particular word. Given the graphical model in Figure 2, and using the simplifying assumption that $p(W|M_a,M_s) = p(W|M_a)p(W|M_s)$ and the fact that $p(M_s|\neg M_a) = 0$, we can represent the probability of a tweet being about a specific movie given a word.

$$\begin{aligned} p(M_s|W) &= \frac{p(M_s,W)}{p(W)} \\ &= \frac{\sum_{m_a} p(W|M_s,m_a)p(M_s|m_a)p(m_a)}{\sum_{m_a}\sum_{m_s} p(W|m_s,m_a)p(m_s|m_a)p(m_a)} \\ &= \frac{p(W|M_s)p(W|M_a)p(M_s|M_a)p(M_a)}{\sum_{m_a}\sum_{m_s} p(W|m_s)p(W|m_a)p(m_s|m_a)p(m_a)} \end{aligned}$$

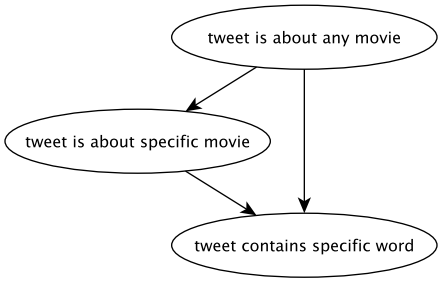

Figure 2: The graphical model representing the probability of a tweet containing a particular word conditional upon it being about any movie and conditional upon it being about a particular movie.

There is a similar derivation for $p(\neg M_s|W)$, though the summand in the numerator remains. This gives us many more probabilities that we have to estimate. However, there are ways to approximate them:

- $p(m_a)$ was calculated using the 10k hand labeled tweets. It was observed that the prior probability of the tweet being about a movie is roughly 1/200.
- $p(W|m_a)$ was calculated using search-labeled data. Because the search-labeled data was selected on the basis of the movie title, the sampling method was not expected to adversely effect the probabilities of movie-general words. Words with high $p(W|M_a)$ included "movie", "watch", and "tonight".
- $p(W|m_s)$ was approximated using IMDB data. That is, we assumed that the distribution of words in IMDB movie reviews matched the distribution of words in tweets about the same movie. For the movie *Transformers: Revenge of the Fallen*, words with high $p(W|M_s)$ included "transformers", "bumblebee", and "optimus".
- $p(m_s|m_a)$ could not be measured. Our strategy was to assume that it would be fixed and calculate it by optimizing the F1 score using a few movies.

Frequent Itemset Analysis As an optimization, we limited the set of movie-general words using frequent itemset analysis. That is, we identified a set of movie-general words having the highest interest before calculating the probability $p(W|M_a)$ for each. "Interest" is defined as:

$$Interest(W) = p(M_s,W) - p(W)$$

Here, we estimate $p(M_s,W)$ by identifying the frequency of word W as it appears in the result of an exact title search. The other probability $p(W)$ is simply the frequency of word W in all tweets.

Letting S_n be the n highest interest words for a particular movie, we calculate how common the word W is using:

$$C_W = \frac{1}{N}\sum_{n=0}^{N} 1\{W \in S_n\}$$

For words common to many movie-specific tweets (e.g. "movie"), this value is larger than 1/2. For words specific to only one movie, it is close to $1/N$.

Estimating $p(M_s|M_a)$ We selected a $p(M_s|M_a)$ to optimizing the F1 score for particular test movies. It was expected that optimizing over various movies would allow us to select an average value which we could use for Naïve Bayes.

We computed the F1 score by measuring both precision and recall. We used our search-labeled dataset to measure both values, though this allows one to obtain a recall value greater than one. For instance, when comparing classification of tweets about *Transformers: Revenge of the Fallen* for which the label is defined based on the presence of the word "transformers", our classification algorithm identified some tweets that were about the movie but did not contain the word "transformers". For instance, the algorithm identified tweets referring to the character Bumblebee. Nevertheless, the goal of this exercise was to *optimize* the F1 score, which could be performed regardless of the denominator used in recall.

The ultimate concern with the Naïve Bayes model is apparent in Table 1, which shows the optimal conditional probability for a set of movies. Note that the optimal value of $p(M_s|M_a)$ varies greatly from movie to movie; it varies from 1/500 to 9/10. Correspondingly, the precision and recall was greatly reduced when using $p(M_s|M_a)$ far away from the optimal value for that movie. Thus, the probability of a tweet being about a specific movie given that that tweet is about any movie cannot be well approximated by a fixed value. Thus, our Naïve Bayes model would not be useful by itself to predict the number of tweets about a movie. It is for this reason that we did not use it to predict movie revenue.

Table 1: Optimal $p(M_s|M_a)$ values.

Movie Title	$p(M_s\|M_a)$
Law Abiding Citizen	0.002
District 9	0.01
Transformers ...	0.9

Performance In Table 2, we compare performance of Naïve Bayes to the direct title search. To do so, we searched over a subset of tweets on opening day of the particular movie. Determining precision is straight-forward: we can simply hand classify the positively-labeled tweets. Recall is more challenging: because it is not feasible to label enough tweets for a usable sample, we must presume a number of *true positives*. However, in our case, we provide the F1 values primarily to compare two classification strategies, and thus the number of true positives is arbitrary as long as we keep it consistent. For each movie, we use as the number of true positives the maximum value of correctly identified tweets over the two classification techniques.

Table 2: Comparison of Title Search and Naïve Bayes performance using F1 score.

Movie Title	Title Search	Naïve Bayes
Law Abiding Citizen	0.64	0.73
Fame	0.29	0.28
Zombieland	1.00	0.35
Transformers ...	0.04	0.84

Notably, there is a large performance improvement when searching for movies with long names like *Transformers: Revenge of the Fallen*, which was expected. However, there is not a significant gain in the identification of tweets relavent to a movie titled with a short, commonly used word like *Fame*. Further, the title search performs much better when looking for movies titled with short, uncommonly used words like *Zombieland*. In general, the Naïve Bayes model provides more consistent F1 score, which suggests that it would result in better revenue prediction were it not for the fact that we cannot hold fixed $p(M_s|M_a)$.

There were observable differences in the word frequencies of IMDB data and Twitter data. The primary difference is that reviews about a particular movie infrequently reference the movie title, as this context is understood by the audience. Conversely, such a context is not understood in the Twittersphere; the audience would not know that a tweet is about a movie unless it contained a movie title or an obvious reference. Thus, we augmented our IMDB data by adding the title to the IMDB reviews at a frequency of one out of every 20 words, which corresponds to the assumption that each tweet about a movie contains roughly one mention of the title. This ensured that the most indicative word for the movie was generally a word in the movie title itself. However, there were cases in which other words were still more indicative, as in "mj" (for Michael Jackson) for the movie *This is It*.

Additionally, there was noticeable difference between IMDB vernacular and Twitter vernacular. For instance, it was observed that the IMDB reviews about *This is It* used the word "mj" less frequently than tweets about the same movie. This keyword appeared with a frequency of approximately one mention every thousand words in IMDB reviews. However, when simply searching for tweets with the phrase "this is it", the incidence of "mj" was approximately one order of magnitude higher ($\approx 1/100$). We consider this to be a conservative estimate because this set of tweets included some tweets that were not about the movie (that is, searching for the movie title was not completely precise). Thus, the incidence of "mj" in tweets about *This is It* is certainly over 1/100.

Frequency Estimation

If we divide the tweets on opening day into two groups, tweets about the specific movie and all remaining tweets, we can consider both separately as bags of words. Next, if we assume that on some arbitrary day far from the movie opening, the bag is entirely not about the movie, then we can estimate the "mix" of the bag on opening day. We estimate the following values:

- $p(W|\neg M_s)$ can be approximated as the frequency of word W on an arbitrary day far away from opening day.
- $p(W|M_s)$ can be approximated as the frequency of word W in IMDB reviews in the same manner as defined in the Naïve Bayes analysis.
- $p(W)$ is essentially a mixed bag observed near or on opening day. So, this probability is equal to the proportionate contribution of each bag.

$$p(W) = p(W|M_s)p(M_s) + p(W|\neg M_s)(1 - p(M_s))$$

Ultimately, we can solve for the prior probability of the movie, which provides the equation below. Assuming that the length of each tweet is constant, then $p(M_s)$ gives us the fraction of tweets about our movie.

$$p(M_s) = \frac{p(W) - p(W|\neg M_s)}{p(W|M_s) - p(W|\neg M_s)}$$

Theoretically, this equation should hold for any word. However, this equation is sensitive to errors when the denominator is close to zero. Our strategy for avoiding this circumstance is to use the word with the highest ratio of $p(W|M_s)$ to $p(W|\neg M_s)$, which tended to prefer words in the movie title like "pelham" from *The Taking of Pelham 1 2 3*.

Performance Results of this process are shown in Figure 3. Some outliers are noticeable: it is impossible for a frequency to be negative, and it is unlikely that nearly all tweets on one day were about one movie. The data point having frequency 0.93 represents *This is It*, for which the frequency is grossly overestimated because we underestimated the frequency of "mj" in tweets about the movie. Ignoring movies with predicted frequencies less than zero and greater than 0.1, we have Figure 4.

Table 3 shows a few comparisons between observed and predicted frequencies, where the observed tweets were identified using a keyword search. Assuming that a majority of the tweets about the test movies contain the keyword specified in the table, then the observed frequency should be within a factor of two of the actual frequency. Notice, however, that the predicted frequency is one to two orders of magnitude too large.

This is could be caused by an underestimation of $p(W|M_s)$. The scope of discussion on IMDB movie reviews is much more rich than what is generally included in movie tweets and is likely to contain a much more varied set of movie-specific

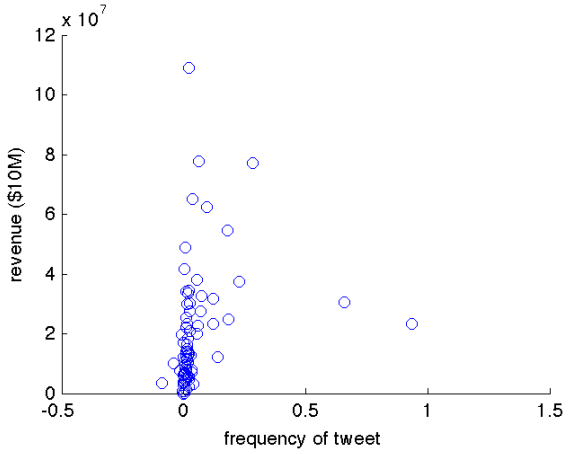

Figure 3: Estimated frequency of movie-specific tweet versus movie revenue, uncorrected.

Table 3: Observed and predicted frequencies of tweets about a movie.

Movie Title	Keyword	Obs.	Pred.
Ice Age 2	"ice age"	0.0004	0.003
Sherlock Holmes	"sherlock"	0.0005	0.01
Transformers ...	"transformers"	0.007	0.02

words. That is, since a tweet about a movie must make obvious to its audience that it is about a particular movie within only a few dozen words, it is unlikely that the tweet will mention an obscure (or, improbable) word related to the movie. IMDB reviews, on the other hand, are free to discuss more nuanced topics at length and in detail. In short, the distribution of words for tweets is skewed towards a smaller set of words. Thus, when selecting the word with the maximum frequency, we underestimate $p(W|M_s)$, which would in turn make our predicted $p(M_s)$ too high.

Nevertheless, the source of this error affects the data in a sufficiently regular fashion that the model offers slightly improved prediction over the title search model, which is discussed in the *Model Comparison* section.

Frequency Estimate Variant As a slight tweak, if we take the words with the highest $p(W|M_s)/p(W|\neg M_s)$ and scale the numerator of the $p(W)$ equation above, we are left with an equation that biases movies for which the indicative words have large changes from the control day and for which the word strongly indicates the movie. Roughly, our score is as follows:

$$\text{keyword score} = (p(W) - p(W|\neg M_s))\frac{p(W|M_s)}{p(W|\neg M_s)}$$

This score further improves our ability to predict box office movie revenue.

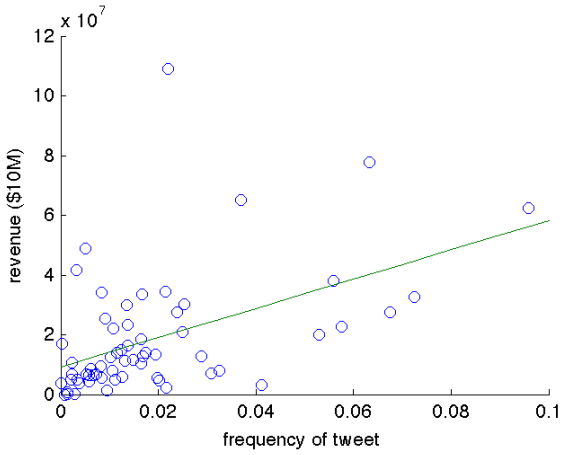

Figure 4: Estimated frequency of movie-specific tweet versus movie revenue, with outliers removed.

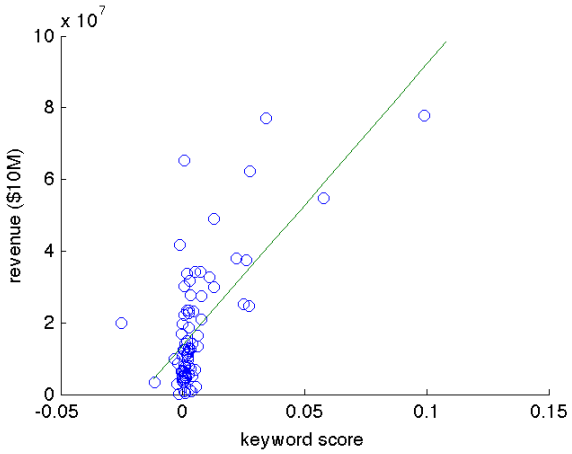

Figure 5: Keyword score and movie revenue.

Model Comparison

For each model, we predict movie revenue using linear regression. Next, we use leave one out cross validation (LOOCV) to compare the performance of our models.

As shown in Figure 6, the average RMS error is improved from $14M to $10M, which one could compare to the average revenue of movies in our set, $14M. Further, we tried fitting to higher order polynomials, but we observed marginal improvement at second order and overfitting at higher order polynomials. In short, the accuracy of our models' predictions leaves room for future work.

Conclusion and Future Work

We demonstrated various techniques for estimating tweet frequencies and attempted to use this to predict movie revenue. Prediction accuracy was improved over a simple title search,

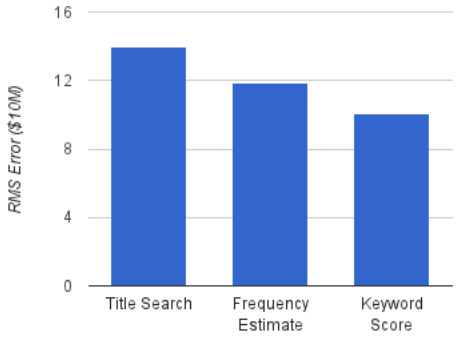

Figure 6: Performance of the various models. Average RMS error from LOOCV decreases with each subsequent model.

though there remains room for improvement.

We demonstrated the potential to use labeled data from an alternative source when labeled data from the target source is absent. However, the result is strictly an approximation and ignores the differing contexts and colloquialisms idiosyncratic to a particular medium.

The revenue of a movie may also be determined by other factors, like revenue of the lead actor, budget of the movie, etc. Adding additional features like this to suppliment tweet frequency could provide a better model for revenue prediction.

Future work could also combine the frequency estimation with classification. That is, if one can estimate the prior probability of tweets about a movie, then one could apply our variant of Naïve Bayes. This could allow one to apply more sophisticated text analysis, like sentiment analysis.