Assignment 3: Data Analysis

Team: Dafi & Xiaoyang

Overview of the project

The idea of this project is to explore that existing consensus on social media has great effects on people's attitudes towards the same topic and there also exists anchoring bias when people making decisions. We chose the latest showing movies, collected IMDb rating data and the tweets related to these movies and tried to figure out the correlations between the sentiment of the Tweets and the tendency of the rating. The CM theory we covered in this project is elaborated as followed:

1. **Anchoring**

   Biases usually creep into decision-making processes. Anchoring means decisions are unduly influenced by initial information that shapes our view of subsequent information. In our situation, we assumed that the existing rating of the movies on IMDb affected people's later evaluation and people inclined to rate similar even the same rating with former people.

2. **Bandwagon effect**

   The bandwagon effect is a phenomenon whereby the rate of uptake of beliefs, ideas, fads and trends increases the more that they have already been adopted by others. The tendency of following the actions or beliefs of others can occur because individuals directly prefer to conform, or because individuals derive information from others. In our project, we want to verify that people tend to follow the consensus of the comments of the movies on Twitter and there's correlation between the tendency of the sentiment ratio and the rating of the movies.

Data collection

We focused our scope of movies to the movies that are opening the weekend of 16-20th of April. The reasoning for this is that we assume that newly opened movies will generate the most traffic on Twitter, and also will have the most changes on the average votes in its IMDb page. The movies we selected are: Paul Blart: Mall Cop 2, Unfriended, Child 44, Monkey Kingdom, True Story, Alex of Venice, Beyond the Reach, Felix and Meira, Monsters: Dark Continent and The Dark Lands. We checked these movies against their IMDb rating and the mentions of the movies on Twitter, both as hashtags and normal mentions. The following describes the method we used for each data collection.

- **IMDb data:**

We created a simple Rails app, hosted on Heroku scraping the IMDb rating of the 10 movies. Since there is no API available for IMDb data, we scraped the whole page, then grabbed the value at the exact location the ratings will be. We used Heroku's scheduler add-on to run rake task of scraping the rating and storing into the database the rating and the time the rating was taken. The collected data then returned as a JSON file for further processing. The JSON file can be seen at (cm-imdb-twitter.herokuapp.com/ratings)

- **Twitter data:**

  We used Twitter Streaming APIs to collected data from April 16th to April 20th. The Twitter Streaming API work as an agreement with Twitter to send us Tweets with the keyword that we specify. The expected Tweets receives is around 40% of the actual amount of tweets. We think that this is an adequate representation of the Twitter community to measure their sentiment. We collected tweets using the movies' name as keywords for searching. At the end of the data collection, we have collected 526491 tweets.

Data Analysis

- **Twitter data:**

1) Parse Twitter data:

   The only relevant data we need out of the Twitter stream is the text of the tweet and the time it was created. We use regular expression to match the tweets with each movie and split them into their respective movies. In order to wipe off the interferential data, we have ignored the tweets containing two or more movies in it. Once the parsing is done, we are left with the following number of tweets for each of the movies:

Paul Blart: Mall Cop 2	54501	Unfriended	345526
Child 44	8673	Monkey Kingdom	6231
True Story	55962	Alex of Venice	109
Beyond the Reach	1671	Felix and Meira	334
Monsters: Dark Continent	11724	The Dead Lands	1260

2) Language detection:

   As the sentiment analysis only works in English, we used Python's 'langdetect' library to do language detection and ignored the tweets written in other languages besides English. The function will return 'en' when the detected language is in English.

3) Sentiment Analysis:

   To detect the sentiment of the tweet, we used the free sentiment analysis API available at http://sentiment.vivekn.com/docs/api/ . The API uses probability analysis bases on his own Naive Bayes classification of past IMDb reviews. A more advanced sentiment analysis API requires a paid service, and we feel that this API works well enough to work on our tweets. The API also provided batch analysis, which is really helpful as we're dealing with a lot of tweets.

   The analysis returns the sentiment (Positive, Negative and Neutral) with it's probability in percentage form. We did an inverse probability in negative sentiment so that the sentiment is represented as a range from 0-100, where high percentage of Negative sentiment is represented as 0-45 in the sentiment range, high percentage of Positive is represented as 55-100 and Neutral is represented as 50.

4) Matching interval

   Since that the IMDb rating was collected with the interval of 10 minutes, and the tweet sentiments are collected whenever tweets arrived, we needed to match the interval. We averaged the tweet sentiments in 10 minutes interval, starting at the moment we started collecting the IMDb data and finishing when we finished collecting the IMDb data.

5) Zero fitting:

   We try ignore the 0 sentiment caused on no tweets available at that given interval of 10 minutes, as that does not mean that the movie has low sentiment rating. We fit the 0s to the line of available sentiment on either side.

6) Averaging Interval

   We figured that 10 minutes interval of tweet sentiment would create a very noisy data. Therefore, we decided to average the interval into a longer period of time. After much trial and error, we decided on increasing the interval into a 60 minutes interval. We then simply took an average of 6 inputs from the previous 10 minutes interval input.

7) Linear regression

Lastly, to detect the trend of the tweet sentiment data, we did a simple linear regression to find a linear function of time and sentiment that represents the trend doing the sentiments.

- **IMDb data:**

  For the IMDb data, we simply created a list of ratings parsed from the JSON file. The resulting data is separated into different text files for different movies. We also did average the interval from 10 minutes to 60 minutes using simple averaging as with the sentiment data. We did not fit the IMDb data, because as we can see on the graphs, there are less variance on this set of data.

- **Compare and plot:**

  We created 2 types of plots, the first one is a combined plot of all movies' IMDb rating. This represents the trend of the ratings of all movies, and allows us to compare and contrast the IMDb rating trend of each movie.

  For the second set of plot, we plotted each movie's IMDb rating with their twitter sentiment data and the fitted sentiment data to show the sentiment trend. From this plot, we can compare the relationship between the movie's rating and sentiment trend, as well as analyzing timely breakdown of each of the data.

Results

Figure 1 depicts the rating tendency of each movie in IMDb during the experiment period. The data is from our IMDb dataset described above.The figure compares two categories, time (on the X-axis); the rating of each movie is represented on the Y-axis. The time interval of X-axis is 10 minutes. The points on each broken line represent the IMDb rating of that movie at that specific time. The figure suggests that the rating of each movie doesn't change much compared to the former rating, which verifies the anchoring bias hypothesis that people tend to give similar or even same rating score based on former evaluation.

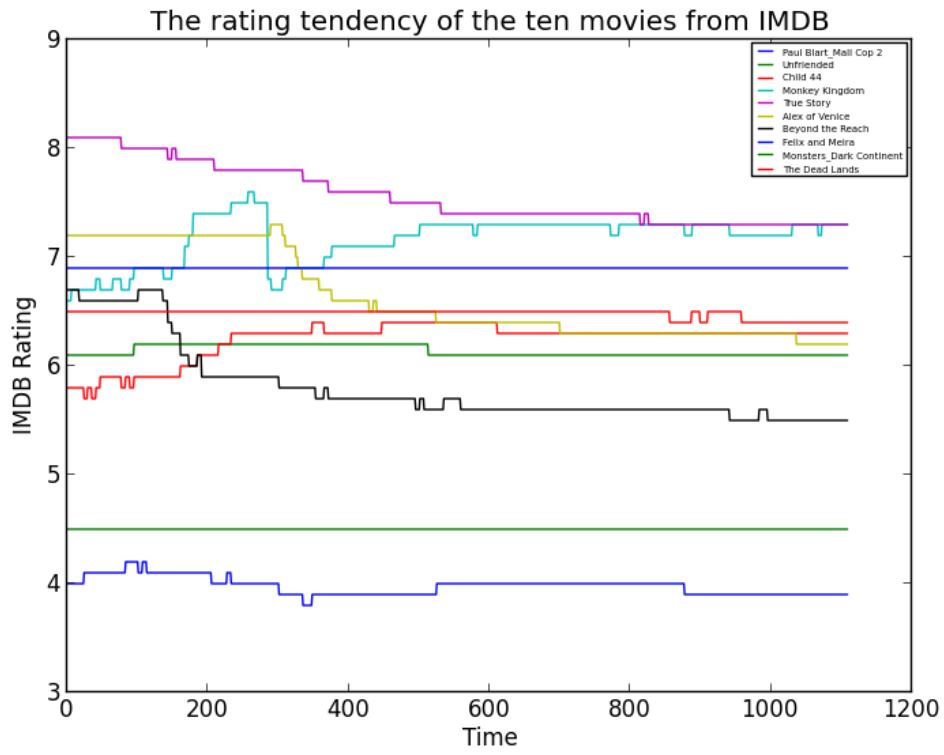
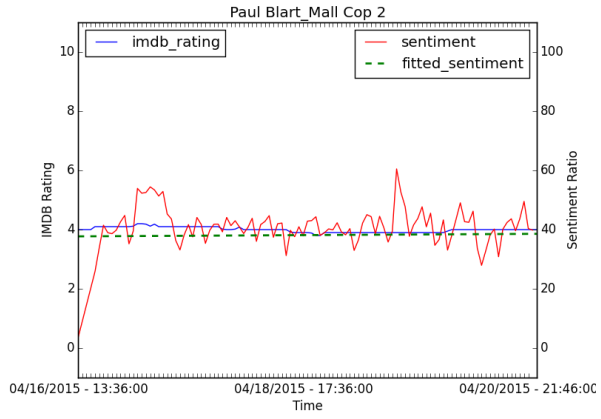
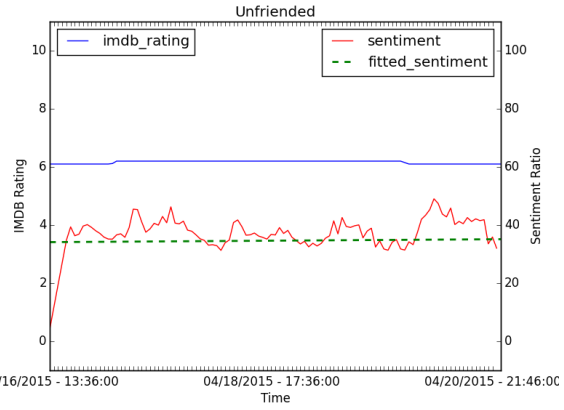


Figure 1

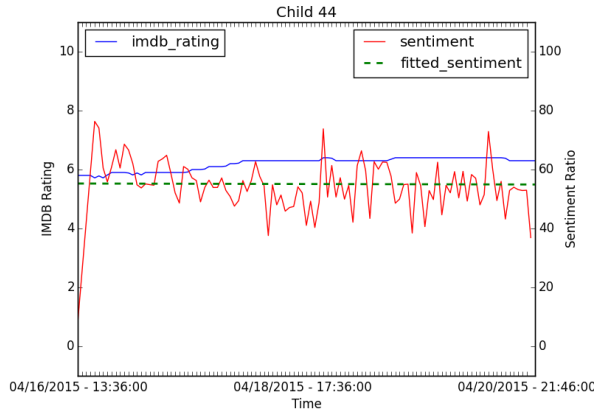
The following ten figures depicts the rating tendency and the sentiment ratio tendency of each movie. The data is from our Twitter dataset and IMDb dataset described above. The blue lines represent the IMDb rating with the range of 0-10, the red lines represent the twitter sentiment data with the range of 0-100, and the broken green line represents the fitted sentiment data based on the linear regression line of the twitter sentiments. The X-axis represents time, which is from April 16th to April 20th, and the Y-axis represents the range of the each measurements of rating and sentiment.

Paul Blart: Mall Cop 2	Unfriended

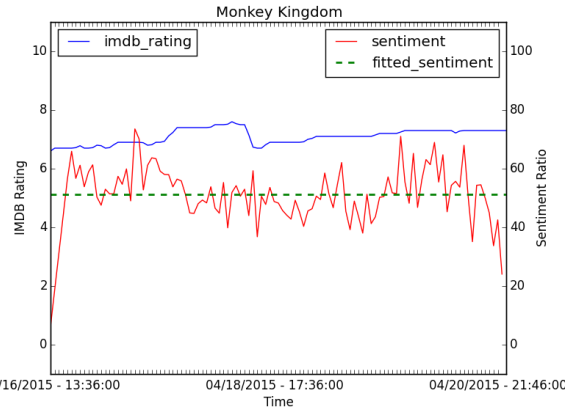




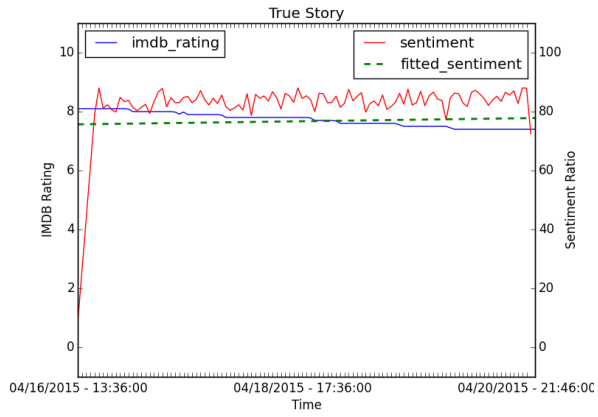
Child 44



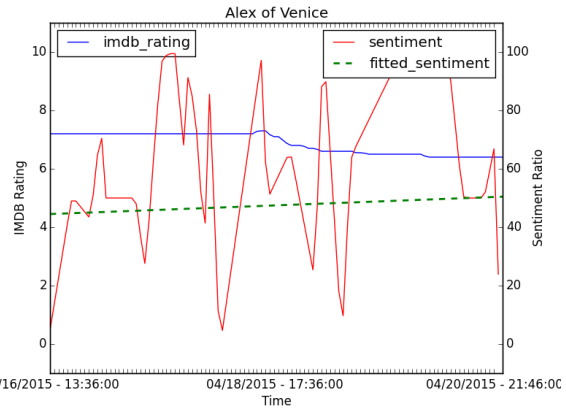
Monkey Kingdom



True Story

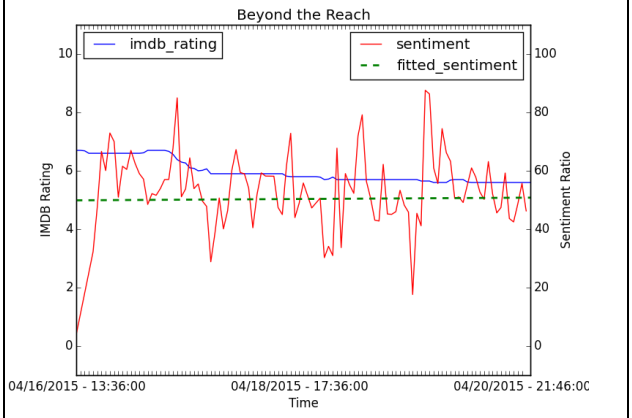


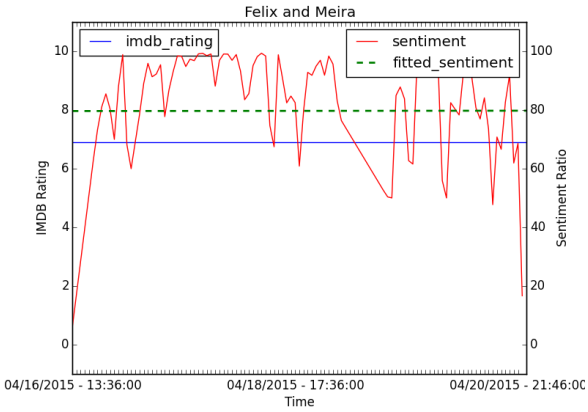
Alex of Venice



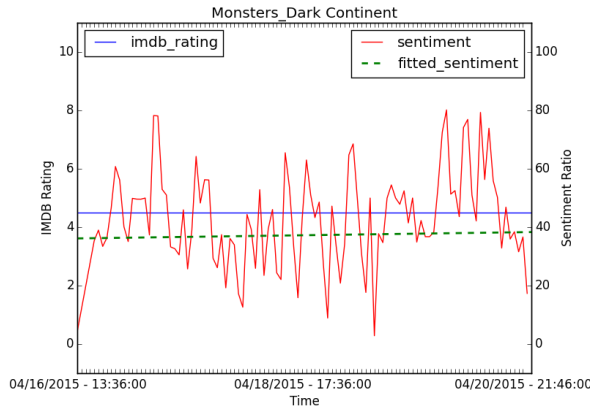
Beyond the Reach

Felix and Meira

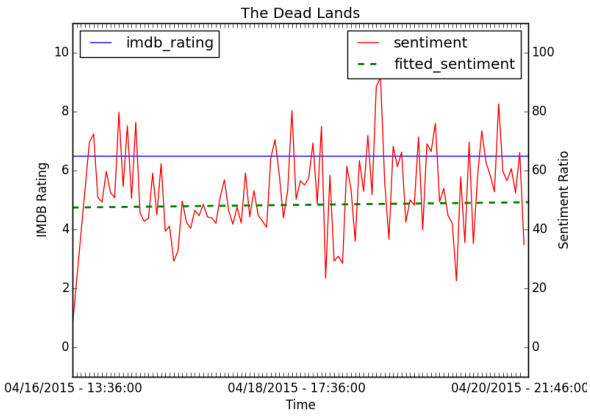




Monsters: Dark Continent



The Dead Lands



Analyzing these figures, we can detect several trends, such as:

- Other than 'Unfriended', the fitted sentiment line and the rating line are within 1 point of one another. Within the factor of time, both line also seems to be converging. Ignoring the actual quality of the movie, it appears that the movie rating is following the Twitter sentiment until the point that it converges. This might be an indication of IMDb raters are following the trend of the public, instead of objectively judging the movie
- The Rotten Tomatoes rating of the movies are as follows; Unfriended (60%), Child 44 (24%), Beyond the Reach (34%), Alex in Venice (77%), Felix and Meira (76%), True Story (47%), Paul Blart: Mall Cop 2 (4%), Monsters: Dark Continent (23%), The Dead Lands (69%), Monkey Kingdom (93%)
- If we consider Rotten Tomatoes rating as indication of the movie actual quality (reasonable assumption, as they are an aggregation of critics rating, which should be less biased), the sentiments are closer to the IMDb rating

compared to Rotten Tomatoes', which lends additional support to that IMDb rating is following sentiments instead of quality.

- 'Unfriended' might be an outlier of the set, because the word 'unfriended' is a common word associated with a negative behaviour (i.e. 'I got unfriended by my best friend, life sucks), might drops the overall sentiment.
- Another trend we can see is that the movie's initial rating before the movie is released may be an anchor to start the movie sentiment. We can see from the graph, the first jump from 0, is always very close to the early rating of the movie (with the exception of 'Unfriended' and 'Alex in Venice'). Again, there is rarely the indication of the actual movie rating, only of how the movie is first perceived on the opening moment.
- Most of the fitted sentiment line shows a generally straight line, with low slope. This also can be a case of bandwagon effect following the early responses
- The biggest sentiment slope is on the movie 'Alex of Venice', which increases over time. Incidentally, the movie also has the biggest negative slope for the IMDb rating. This may seem counter-intuitive, however, if you look at the graph, their initial difference from sentiment-ratings is so high that the progress only made it closer together, which support the theory that IMDb ratings is controlled by public sentiment