

Automated Movie Review Ranking

Jordan Timmermann | jtimmer2@uiuc.edu

1 Motivation

Several months ago, I read a response by film critic Jonathan Rosenbaum to the following question, posed by another critic, Josh Timmermann (who also happens to be my brother):

*“In your review of *No Country for Old Men*, you gave the film one star in what was effectively a pan that conceded the film had ‘redeeming facets,’ yet in the ‘Critic’s Choice’ section of the current issue of *Film Comment*, you award the Coens’ film four stars. You were very nearly alone among major American film critics in actually criticizing *No Country*, while now it seems you’ve joined the consensus. What gives?*

*Also, your capsule review of *Redacted* seemed ambivalent but more positive than negative, albeit with significant reservations. In [*Film Comment*], you give that film two stars.*

I’m just curious if you’ve had a change of heart on either or both of these movies, or if there’s some other explanation for this incongruity.”

The response was as follows:

“The simple answer is that I have changes of heart (and mind) all the time...For me, it’s the artificial stars and other rating systems that are screwed up (by virtue of being simpleminded and consumerist), which my shifts in evaluation, both quite conscious, were intended to reflect. The alternative is to stop thinking about films once we affix grades to them – something that I sometimes do but am not especially proud of.

*The fact that *Film Comment* uses five stars while the *Reader* uses only four only adds to the muddle. So if I had to give a star rating now to either *Redacted* or *No Country*, both would come closer to my present position than picking one over the other. I guess this is because for me, movies are like life – i.e., complex.”*

In the initial question, it is observed that a particular film received two conflicting ratings from the same critic (in two separate publications). Rosenbaum notes in his response that he frequently has changes “of heart (and mind),” a situation which I surmise is

very common among those who review films. Unfortunately, the nature of the initial publication makes it difficult to publicly express this change in opinion, so those who read Rosenbaum’s review in *The Chicago Reader* (and did not subsequently read that of *Film Comment*) have no way to know of it. Indeed, neither of the two major web sites that serve to present users with summarizations and ratings of movie reviews (rottentomatoes.com and metacritic.com) indicate his modified opinion of the film.

Furthermore, Rosenbaum points out the fact that one of the publications uses a five-star rating system while the other uses a four-star system. How should a critic, writing for both publications, rate one film on two such scales? If the critic gives the film three of four stars, what is the equivalent on the five-star scale? Considered in this context, the superficiality of these rating systems becomes apparent. This is also clear from the second paragraph of the initial question, where it is observed that the rating given to a film does not seem appropriate, considering the actual review. In this case, the language of the review was not consistent with the affixed rating.

The previously observations served as the motivation for this project: rather than relying on arbitrary and incompatible rating systems, it would be more interesting and useful to have the ratings for movie reviews be based on the actual language used in the reviews. If a system that could generate ratings in such a way existed, it would be an effective solution to all of the issues describe above. First, this system could use one consistent scale to generate the rankings; this would avoid the problems inherent to the various, existing rating systems (four stars, five stars, thumbs-up, 0-10, etc.). Also, since ratings could be generated based solely upon the language of an article, the rating would change appropriately if the review were modified to reflect a change in opinion. When it was noted above that the rating for a film did not seem appropriate for the review itself, it shows that the language of the review was being analyzed, and then a prediction was being made as to what rating the critic had intended; in this particular case, the rating that was predicted did not match that which had been affixed to the review. Predicting the author’s rating, based on an analysis of the language in the review, would essentially be the functionality of

this proposed rating system, therefore preventing the discrepancy that was exhibited in the described case.

2 Functionality

The current system for rating a movie includes affixing some superficial value to a film critic’s review of said movie. There is a myriad of scales used for these ratings, including (but certainly not limited to): out of four stars, out of five stars, thumbs up or thumbs down, 0-10, and 0-100. As explained above, once a review has been labeled with a particular rating, that rating tends to persist despite a potential change of opinion by the author. In addition to this, there is rarely any indication as to how a rating is applied to a review – is the author solely responsible for the rating, or does the publisher have some influence over the the creation (or approval) of the rating?

Some web sites have attempted to combine the ratings by many critics in an attempt to give one unified rating. While this is an interesting idea, there are several issues that make this system less than desirable. First, these sites scale the original ratings from each critic to match their own rating system; if a critic gave a movie three of four stars, for instance, that review will be displayed as a 75/100 on the web site. Although the critic gave the movie a rating that is logically the equivalent of a 75/100, the rating might have, in reality, been a way of expressing a positive opinion for the movie, while refraining from saying that it was exceptional. That is, a 75/100 is much more specific, and cannot be interpreted in the same manner. Furthermore, these sites include the opinions of a large number of critics (determined by the site), and users see a summary over all of these critics, with no option to see that of some subset of critics. Some critics (and publications) are arguably less reliable than others, and the user might not want the ratings for those reviews contributing to the total score that they see. Finally, these sites have no way of incorporating unrated, non-commercial reviews (e.g. blogs). This latter point is not necessarily a fault with the existing systems, but is still a functionality that would be quite useful.

The proposed system, however, is designed to use one consistent rating system; for this project, a 0-100 system was used. Rather than scaling the various rating systems to match this scale, the proposed system will attempt to generate altogether new ratings based on the language of reviews; this method should bypass the issues described above regarding the various rating scales. Also, since an initial rating is unnecessary in order to generate a rating, unrated reviews (e.g. blogs) can have ratings generated just as easily as any

other review. This system would be especially useful as the back-end to a web site interface, where users could choose which critics or publications they would like to use in determining the ratings for particular films. In analyzing the language of reviews, there are several other possible functionalities for such an interface. First, portions of the reviews could be deemed as important and highlighted to give the user an indication of how the review was scored, as well as making it easier to more quickly read a lengthy review. Second, we could potentially create summarizations consisting of the most important few sentences of a review, which would also allow for the user to more quickly digest the text of reviews.

3 Implementation

Of the many domains to which sentiment classification may be applied, movie reviews are among the most difficult; the high level of language used by film critics makes it harder to extract the opinion-expressing words from the reviews; unlike product reviews (which are often written by consumers) movie reviews are less likely to contain the words most commonly associated with expressing sentiment (“good,” “bad,” “great,” etc.). Nonetheless, the methods used to classify sentiment in these other domains can be applied to that of movie reviews.

Lee, Pang, and Vaithyanathan (2002) found that several machine learning techniques could be used to effectively classify movie reviews as positive, negative, or neutral. This research treated sentiment classification as a case of text categorization, where the two classes used to categorize reviews were positive sentiment and negative sentiment.

Unlike the latter work, the goal of this project was to generate a specific rating—from 0 to 100—based on the language used in each movie review (rather than to just identify the general sentiment). (With this in mind, a positive review would receive some rating between 50 and 100, while a negative review would receive a rating between 0 and 50.) Therefore, more than two classes had to be used to classify reviews. Specifically, five classes were used, corresponding to reviews with ratings of 0, 25, 50, 75, and 100. While researching for this project, it was noticed that reviews with perfect ratings contain significantly different language than those with (the equivalent of) a 75 or 80 rating, so an additional class was used for reviews with perfect ratings.

The categorization algorithm used was the Naive Bayes classifier algorithm, which assigns a document to the class in which it most probably belongs. That is, for a given document d , the document is assigned

1	bad	6	waste	11	lame	16	memorable	21	script
2	worst	7	awful	12	perfect	17	dull	22	plot
3	stupid	8	wasted	13	life	18	poorly	23	subtle
4	boring	9	outstanding	14	supposed	19	excellent	24	performances
5	ridiculous	10	mess	15	wonderfully	20	perfectly	25	terrible

Table 1: Words with the highest mutual information (positive/negative data set).

1	power	6	hottie	11	filmmaking	16	knocked	21	life
2	country	7	london	12	worst	17	nottie	22	friend
3	story	8	film	13	performance	18	vulgar	23	scene
4	national	9	great	14	paris	19	hilton	24	storytelling
5	johnson	10	french	15	marvelous	20	family	25	shot

Table 2: Words with highest mutual information (rated data set).

to the class $c^* = \arg \max_c P(c|d)$. In order to create the classes needed to categorize each document (movie review), training data had to first be obtained; since most reviews by major critics are initially given some rating (as described above), those reviews and their corresponding ratings were used as training data for the system (more information later on what training data was used, and how it was obtained). Once sufficient training data had been obtained, a unigram language model was used to represent each class. Since the majority of the words that appear in movie reviews (like most other documents) do nothing to indicate the sentiment of the review, an initial tactic was to use the training data to find a set of feature words $\{f_1, \dots, f_m\}$; these words should be those that are the most discriminating for the documents in our training set. The words that contained the highest mutual information throughout the training data were used for this set. Once the set of feature words had been obtained, each document d was represented as a vector $(n_1(d), \dots, n_m(d))$, where the value $n_i(d)$ is the number of times the feature word f_i appeared in the document. In order to obtain the Naive Bayes classifier, one must first observe Bayes' rule:

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

Assuming that the f_i values are conditionally independent, an estimate for $P(d|c)$ can be made:

$$P_{NB}(c|d) := \frac{P(c) \prod_{i=1}^m P(f_i|c)^{n_i(d)}}{P(d)}$$

In this case, the prior probabilities for each class c_i were estimated as $P(c_i) = \frac{N_i}{N}$ where N_i is the number of documents in class c_i and N is the total number of documents in the training data. Once the probabilities of each class c_i were obtained for the document, there were a variety of methods that could have been used to generate the document's rating. A fairly basic method would be to take the sum of the products

of the class probabilities and their numerical values:

$$\sum_{c_i \in C} P_{NB}(c_i|d) * V_i$$

where V_i is the corresponding values for each class (0, 25, 50, 75, and 100).

4 Results

Two different sets of data were used while testing the implementation of this project. The first—from Lee, Pang, and Vaithyanathan (2002)—contained two sets of documents, one consisting of positive reviews and the other of negative reviews. While this set is not directly applicable to this particular project (since it only contains the two categories), it was an effective way to test the feature set creation method using highest mutual information, as the set contained 2000 documents total (1000 for each category). The results for the top twenty-five feature words are shown in Table 1.

Similarly, a set of data was assembled that could be used within the context of this particular project. The best way to do this was to utilize web sites like rottentomatoes.com and metacritic.com, which contain the scaled versions of the original review ratings, along with links to the respective reviews. Unfortunately, neither of these sites have an API for obtaining said data, so the only automated option was to crawl the site, recording the rating for each review, then following the links to those reviews and parsing the page source. I chose metacritic.com, as it provides a cleaner interface and only has reviews from the more respectable publications (which are assumedly of a higher caliber). However, several problems existed in using the method described above for obtaining the text of reviews. First, there is no guarantee that the links to the reviews are active, and in-

deed many of the reviews older than several months had either been moved or no longer existed. Second, some of the reviews existed on sites that require subscriptions in order to access content. The final, and most common, issue was that the links produced pages containing multiple reviews (in a "This Week in Movies" format). A crawler was initially created but, for these reasons, it was found that manually obtaining the reviews seemed to (unfortunately) be more effective. In total, approximately seventy-five reviews were obtained for each class; while the most discriminating words for these data (Table 2) do contain several words that one might expect to find, the majority of the words seem to exist because the volume of training data was simply not large enough. In fact, some of the words (e.g. "country" and "nottie") exist because they are parts of the titles of movies that were rated the best and worst of this year (due to the issue with non-recent reviews being hard to obtain, most of the reviews that were obtained tended to be recent). Using this training data, however, reasonable ratings were generated for various reviews. For instance, given the (unrated) review below, the algorithm generated a rating of 75.6.

"Ellen Page is spot-on in what feels like the softer flip-side to her no less striking Hard Candy turn. Michael Cera is Michael Cera (or the on-screen persona he's carefully cultivated since at least Arrested Development), which is just fine. Jason Bateman, J.K. Simmons, and Allison Janney are first-rate, too. But this movie's most significant revelation was (for me, anyway) its most significant star. I think I only watched maybe two or three episodes of Alias so maybe I missed something, but Jennifer Garner demonstrates real range in a role that provides her with only marginally more to work with than she had in, say, 13 Going on 30. That scene in the mall where she leans down to try and hear the baby kick inside Juno's belly should be her Oscar clip-if there's any justice (and if there's any Oscars)."

Once more training data was added to the system, this review increased to a 76.2; this seems reasonable, as the critic appears to have a rather high opinion of everything discussed.

5 Discussion

The primary issue with this project seems to have been a shortage of training data. While truly sufficient training data surely exists, there does not seem to be any clear method for obtaining it automatically. Using the manual method—copying and pasting the text of reviews into files according to their rating—

seems like it might be a task suited for a system like the Amazon Mechanical Turk.

There are also some methods that could be employed which could possibly make the algorithm more effective. The first issue again involves the training data: if reviews could be randomly selected – and then if only reviews from a fixed time frame were selected in order to improve the prior probabilities of the classes – then we would have fewer words like movie titles and actor names appearing in the set of feature words. Similarly, there could be ways of filtering out only the words that would be likely to determine opinion, before the set of feature words is computed. For instance, part-of-speech tagging could be performed on the reviews, and then only the words that are adjectives and adverbs could be retained. Similarly, a record of people and titles associated with movies could be kept and updated, such that all of those words would be removed from each review. We could also modify the method of scoring reviews from using the class probabilities and numerical class values to using a series of binary comparisons (positive or negative, then 0-50, 0-25, etc.), which would perhaps allow for better classification of the document. Another area for improvement might be to use only the reviews for one particular critic, which could allow for the use of less training data (since the language is likely to be more consistent).

Furthermore, there are several interesting applications (some of which were mentioned earlier) that could result from this system. The most important parts of reviews could be highlighted (or indicated to a user in some other way), and summaries of entire reviews could be generated, in order to allow for quicker processing of reviews by users. We might also be able to find cases where reviews are given misleadingly high or low ratings (as compared to the language used by the critic), and perhaps find patterns of this for certain publications.

6 References

- Bo Pang, Lillian Lee and Shivakumar Vaithyanathan. *Thumbs up? Sentimental classification using machine learning techniques*. In Proceedings of EMNLP 2002, pp.79-86.
- Bo Pang and Lillian Lee. *Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales*. In Proceedings of ACL 2005, pp.115-124.
- Li Zhuang, Feng Jing and Xiao-Yan Zhu. *Movie Review Mining and Summarization*. In Proceedings of CIKM 2006, pp.43-50.