Classification of Video Game User Reviews

Derek Huelsman
Stanford CS 229 Project
December 16, 2011

Abstract

Using a set of 800 video game user reviews, I was able to train a supervised learning algorithm to classify user opinions of video games as “worth buying” or “not worth buying” with an accuracy of 92.63%. I used this algorithm to find the user percent approval of eight different games currently in the market as well as the average scores given by those who believed the games was “worth buying” and “not worth buying”. The results differed greatly from the critical consensus, suggesting that user reviews provide a valuable alternate perspective on video game quality.

1. Introduction

The internet has made it possible for millions of users to submit personal reviews for nearly all available consumer products. The impetus for websites to provide this feature is two-fold: it attracts those who wish to have their voices heard and it attracts those who are looking for a reliable source of public opinion. The video game industry in particular is greatly affected by this social internet feature. This is because buying a video game is a non-negligible investment of both time and money; some of the best games of this generation cannot be completed in 100 hours and cost as much as $60 new. Since the video game industry provides few opportunities to get free hands-on experience with a game pre-purchase, website and magazine reviews are perhaps the consumer’s most valuable resource.

Websites like GameRankings.com and Metacritic.com attempt to calculate the average scores of all published professional reviews for each game. However, there is no requirement that different sources rate games on comparable scales. As a result, the average review score from each of GameRankings.com’s 300 review sources ranges from 59.34% to 86.32%, and these are often converted from letter-grading schemes. This suggests the same score often means something completely different coming from two different sources.

The problem is arguably worse at the user level. Gaming websites that allow users to give scores to accompany their reviews rarely provide a qualitative scale to guide users. As a result, each user draws the line between recommendation and condemnation in a different place. Below are some samples of negative and positive reviews from Gamespot.com with provided scores and summaries:

- 8.5 Negative “Quick review of a game that didn’t live up to the hype.”
- 8.0 Negative “Ruined by a rushed release. Bugs, sloppy meshes, oh my!”
- 8.0 Positive “The best of the unpolished games of 2011”
- 7.5 Positive “Excellent sequel”

Another problem is that some users are not afraid to give the maximum or minimum allowable scores for games they do/don’t like just to have the maximum impact on the displayed average user score. As a result, these reviews are effectively weighted more than the reviews from those who choose to use the entire range of scores. It’s not even a covert practice:

- 10 “I don't actually give this game a 10, it’s to level out the ratings of people who put 1.0... I give it an 8 like gamespot”

Occasionally users simply enter a rating they didn’t intend:

- 1.0 “I have never played a game this good, this amazing, its perfection”

This makes it difficult for visitors to these websites looking for game recommendations to get players’ overall impression of a game in a reasonable amount of time.

These problems can be avoided by classifying reviews as positive or negative, or more explicitly, categorizing reviews based on whether the user believes the game is “worth buying” or “not worth buying”. The average of many of these classifications is akin to a probability that your investment will pay off. With this information, a more valuable consensus can be made available to interested consumers. No gaming website that I know of has implemented a system like this, but, using machine learning and text analysis, it is implemented here using existing user reviews.

I will be using user reviews and not professional reviews primarily because there are many more of them, and they currently receive much less focus than their professional counterparts, despite being equally valid and equally valuable opinions of games.

2. Pre-Processing

The training set used consists of 400 positive and 400 negative reviews, approximately ten of each from 40 different top-selling video games between 2008 an 2010. These were scraped from GameSpot.com and GameFAQs.com. Reviews were included from such a large quantity of games in order to avoid bias from genre-specific text. The data stored for each review includes a positivity label of 1 or 0, the user’s rating on a 100-point scale, and the review text itself. The following steps were used to preprocess the data:

- The Stanford Tokenizer provided by the Stanford Natural Language Processing Group (http://nlp.stanford.edu/index.shtml) was used to remove empty lines from the text and put each term or “token” on a separate line.
- All meaningless punctuation was removed. (periods, commas, quotation marks)
- Stop words were removed using a modified list provided by Cornell University (ftp://ftp.cs.cornell.edu/pub/smart/english.stop). A different set of stop words was used depending on if bigrams were included in the term-document matrix, since common words like “not” and “is” become valuable when attached to other words.
- The Porter stemming algorithm was used to combine words with the same stems (http://tartarus.org/martin/PorterStemmer/). For example, “addiction”, “addicting”, and “addictive” would all be reduced to “addict”.
- A vocabulary was created for the entire training set. This included 12,677 unigrams and 40,697 bigrams.

Although it was considered, including trigrams or other n-grams increases computation time dramatically, making it infeasible to include them in this project.

3. Term-Document Matrix Creation

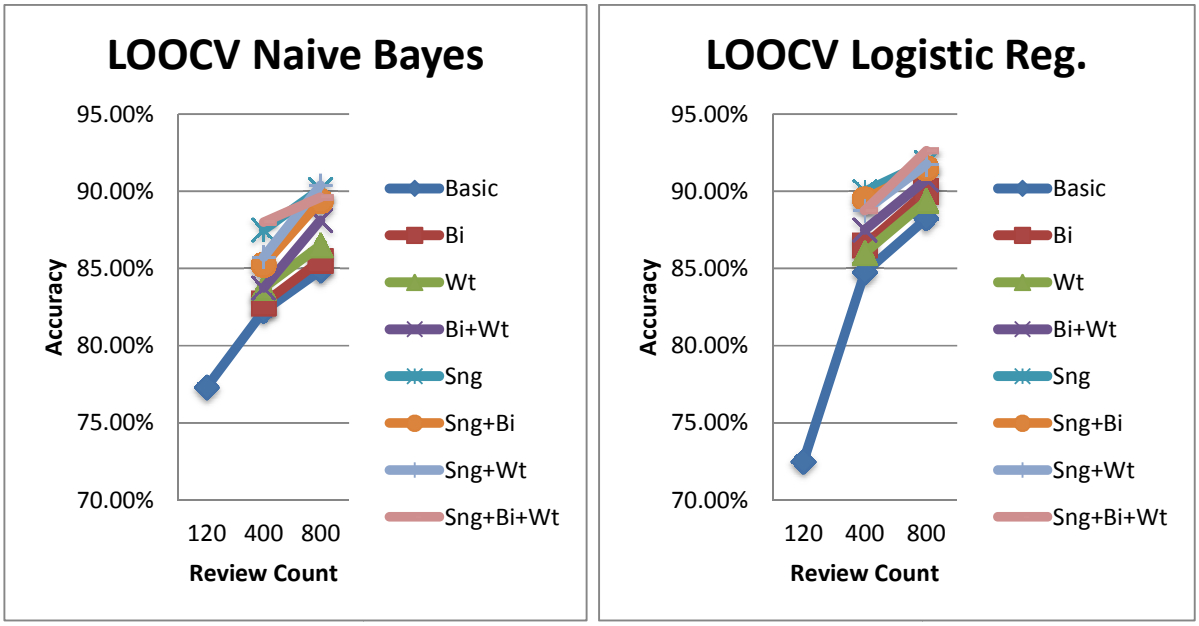
A term-document matrix is made up of the number of occurrences of term j in document i, or in this case, review i. From this, several learning algorithms can be applied under the assumption that the probability of a term occurring in a review is independent of any other term occurring in the review. Eight (2^3) distinct term-document matrices were created using the training data in an effort to test how much different combinations of three matrix preparation methods would affect review classification accuracy:

- Including bigrams in addition to unigrams (increasing the number of columns from 12,677 to 53,374).
- Doubling the term weight of the first 5% and last 10% of each review. These are approximately the locations where reviewers were most likely to giving their overall opinion of a game. Several different weight distributions were tested.
- Counting only if a term occurs in a document or not, rather than the number of occurrences.

I also tried several implementations of tf-idf (term frequency-inverse document frequency), but none of them improved accuracy.

4. Training Results

I used Naïve Bayes, Logistic Regression, and Support Vector Machines along with Leave-one-out Cross Validation (LOOCV) to find which algorithm provided the highest classification accuracy for the training data. L^2-regularized Logistic Regression performed on average about 0.25% better than the best SVM algorithm, which is nearly negligible, so I will provide the training results for Naïve Bayes and Logistic Regression with the assumption that the best Logistic Regression and SVM results are identical.



Bi: Bigrams used Wt: Weights used Sng: Single counts used

With the set of 800 reviews, using bigrams, weights, and single counts with L^2-regularized logistic regression achieved the highest accuracy, 92.63%. It is interesting to note that using only unigrams and single counts achieved the second best accuracy of 91.87% (and the best accuracy using only 400

reviews). Thus, this would be a reasonable alternative if the increased computation time of including bigrams became a problem. Logistic Regression benefitted the most from increased review count, and it is likely another increase in review count would increase accuracy a little further.

Among the most common tokens indicative of a positive review were “must-have”, “awsome [sic]” “finest”, “underrated”, “addicting”, “is+awesome”, “personal+favorite”, “must+buy”, “highly+recommended”, and “great+addition”. Among the most common tokens indicative of a negative review were “disgrace”, “horrible”, “unacceptable”, “overrated”, “lame”, “not+buy”, “is+boring”, “not+worth”, “bargain+bin”, and “huge+disappointment”.

5. Testing Results

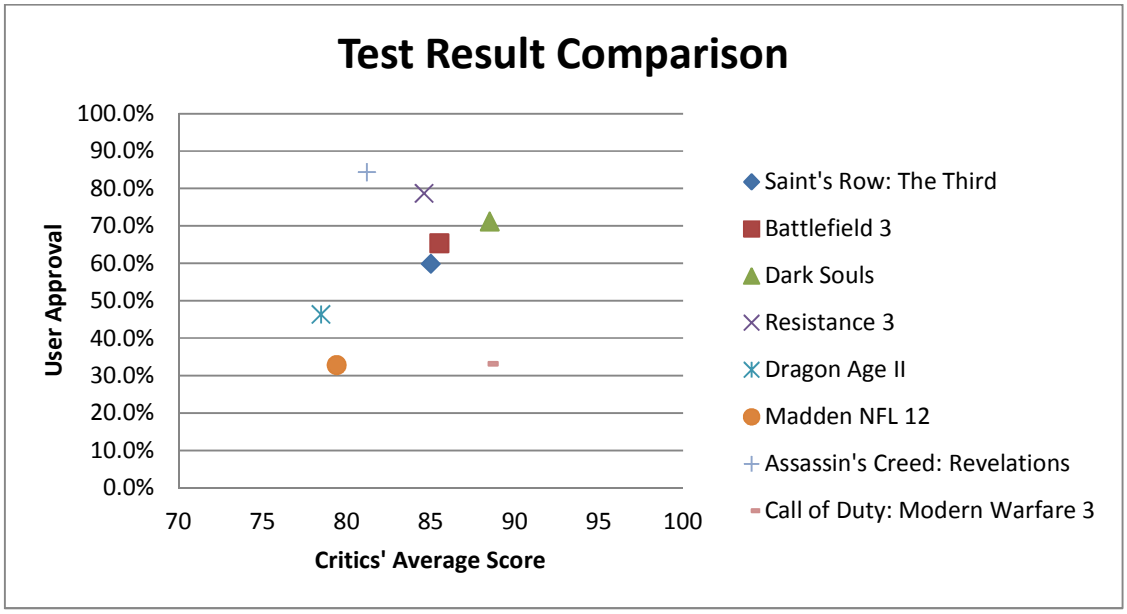
In order to apply the best-performing algorithm to a real-world situation, I acquired the 80 most recent reviews (or the total number of reviews available) for some of the most highly anticipated games of 2011. Games like Batman: Arkham Asylum, The Elder Scrolls V: Skyrim, and The Legend of Zelda: Skyward Sword received nearly unanimous praise from critics and users alike, so they were excluded here. I was interested in games that received critics’ scores in the high 70s and 80s, where it may be unclear for consumers what the score actually means. Adjusted % Approval was calculated from the Test % Approval by taking into consideration the classification error from the training results: 6.6% of bad games were classified as good while 8.1% of good games were classified as bad. Using these values, we can find the expected value of the actual percent approval by solving the equation,

$$Adj. + 0.066\,(100 - Adj.) - 0.081\,Adj. = Test$$

for $Adj.$ As the accuracy of an algorithm increases, the Adjusted % Approval approaches the Test % Approval. Critics’ scores were aggregated by GameRankings.com.

Game Title	Critics’ Average	User Review Count	Test % Approval	Adjusted % Approval	Average Approval Score	Average Disapproval Score
Saint's Row: The Third	85	64	57.8%	59.9%	90.4	69.8
Battlefield 3	85.5	80	62.5%	65.4%	90.8	72.8
Dark Souls	88.5	80	67.5%	71.2%	94.2	75
Resistance 3	84.6	46	73.9%	78.7%	90.3	75
Dragon Age II	78.47	80	46.3%	46.4%	82.57	57.44
Madden NFL 12	79.4	49	34.7%	32.8%	77.7	59.1
Assassin's Creed: Revelations	81.2	80	78.8%	84.4%	90.2	75.6
Call of Duty: Modern Warfare 3	88.43	80	35.0%	33.2%	79.82	50.29

The Average Approval Score is the average rating provided by those who believe the game is worth buying. The Average Disapproval Score is similar. This provides some additional useful information about the game. For example, even though Dark Souls received only the third highest Adjusted Approval, it received the highest Average Approval Score, indicative of the fact that it tends to be a love it or hate it game.



Perhaps surprisingly, there is no clear correlation here between the critics' average score and the user approval percentage. In fact, Assassin's Creed received by far the highest approval, despite having the third lowest Critics' Average. It would become more obvious that a correlation exists if we also included unanimously praised and panned games, but the fact that there are significant outliers in this score region suggests that critics' scores may not be enough to get an accurate read on the quality of a game.

6. Discussion

The information acquired in the testing results is particularly valuable because it took critics' scores spread over a range of less than 10 on a scale of 100, and spread them to a range between 32% and 84%. It is now much easier to distinguish between games that one may otherwise have believed were equally valued. While professional reviews remain the most valuable resource for consumers interested in the technical aspects of games, it makes more sense to get information on how worthwhile a game is from as many different sources as possible. In this case, the vast number of user reviews freely available on the internet should suffice.

There remain some unresolved issues that lower classification accuracy:

- Some reviews simply offer no personal opinion on the game. How can these be ignored? (And should they be ignored?)
- Since the video game industry attracts a younger crowd, rampant spelling mistakes can leave little for the classifier to work with. Would a SpellCheck implementation be worthwhile?
- There's a problem with users repeatedly referring to other games with the opposite reaction ("That was the greatest game of all time but this one is not"), which can throw off the classifier. I'm not sure of an easy way to work around this problem.

With more time, I would have liked to try using a parser to extract adjectives from sets of reviews for games in order to obtain consensus game descriptors.

All websites referenced are given at the relevant location in the text. The CS299 lecture notes were used to understand and implement all learning algorithms. MATLAB was the primary computing environment.