IMPROVED RANKING OF RATED CONTENT THROUGH LINEAR PREDICTION

Thesis Supervisor:

Dan E. Tamir, Ph.D.
Department of Computer Science

Approved:

Heather C. Galloway, Ph.D.
Dean, Honors College

IMPROVED RANKING OF RATED CONTENT THROUGH LINEAR PREDICTION

Honors Thesis

Presented to the Honors Committee of
Texas State University-San Marcos
in Partial Fulfillment
of the Requirements

for Graduation in the Honors College

by
Robert Russell Kuykendall

San Marcos, Texas
December 2012

Improved Ranking of Rated Content Through Linear Prediction

Robert Russell Kuykendall

December 14, 2012

Abstract

The most common methods of ranking rated content do not account for the trend of ratings. However, the perception of quality for rated content may shift over time. We explore incorporating linear prediction of ratings into the calculation of ranking, using linear predictive coefficients. The difference between the mean of predictions and future values was compared to difference between the mean of current and future values. The results show promise and provide motivation for future research.

1 Foreword

My interest in rated content began in 2007 with a personal web project involving user created game modifications. After adding the ability for users to submit ratings, I wanted to display a list of everything on the site ranked by user reviews, so that new users could easily find the best content that we had to offer. To do this, I ranked every item by the mean of it's user reviews. Unfortunately, the best content on the site had a mean rating of a very high four, and as soon as a new item was submitted and received a 5 star rating, it jumped to the top of the list. The item would soon settle lower as new ratings were submitted, but I saw this as a bug. Clearly, a new item with a single five star rating was not the best content available, or even as likely to be good as an item with a large number of four star reviews. To remedy this, I developed a very lightweight formula to increase or reduce the ranking of items based not only the mean of their ratings, but the number of ratings. I would not recommend it for any sizable audience, but it worked perfectly on a small project. Since then, I became fascinated with the problems and solutions of ranking rated content around the web.

$$W = M + (M - 2.5) * (N/10.0) \tag{1}$$

For curious minds, I have included the formula above. W represents the new ranking number, M represents the mean of the item ratings, and N represents the total number of ratings for that item. I used 2.5 as the divider between positive and negative reception. Any item with a mean above 2.5 was ranked higher with more reviews, and any item with a mean below 2.5 was ranked lower with more reviews.

2 Introduction

"When we post and then tag pictures, we are teaching the Machine. Each time we forge a link, we teach it an idea. Think of the 100 Billion times *per day* humans click on a Web page, teaching the Machine. The machine is us."

–Michael Wesch, Associate Professor of Cultural Anthropology at Kansas State University [1]

Ratings and rankings are ubiquitous online. Almost every Facebook post and YouTube video users see has been ranked as important enough to bring to their attention. Every comment written or video played acts as a rating which spreads that content to other users. As the amount of content generated daily continues to grow, so does the importance and necessity of user ratings and robust ranking.

2.1 Background

Ratings, such as "4/5 Stars" are often used to quantitatively represent user perception of quality. Ratings indicate "how much" a user likes something, in a way that can be quickly and easily compared to the experience of other users. Content is ranked against other content, on lists such as "Best Books on World War II," by reducing vast numbers of ratings to a single number. For example, a list of books might be ranked by the mean of hundreds of ratings for each book.

Research into improving the ranking of rated content has dealt with several issues in a number of ways. One problem is that new content has fewer ratings than established content, which can present a bias towards established content if ranking considers the total number of reviews. Researchers have worked to predict the future ratings of new content by comparing it established content [2], or by comparing the way content has received early ratings to the early ratings of established content [3]. The reputation of users has also been used to improve rankings, based on attributes such as activity level [4], or their ratings of other content [3].

2.2 The Problem

In situations where the trend of product ratings indicates a change in user perceptions of quality, the ranking of products should reflect this more clearly than the mean of all ratings can accomplish. For example, if a book is extremely well received at first, but its popularity fades very quickly, then all the positive ratings it has received earlier may be misrepresenting its current popularity.

By reflecting the moving trends of product ratings more accurately, the ranking of products can more accurately represent the current feelings of users. Financial opportunities and content quality suffer when the trends of products or content are not taken into account in the ranking or presentation.

2.3 Possible Solution

Linear predictive coding can be used to predict future ratings based on existing ratings. Predicted ratings can then be incorporated into content ranking to bring the results closer to future rankings.

3 Background

This paper concentrates on the ratings, and ranking, of online content. Additionally, it leverages the use of the linear predictive coding.

3.1 Ratings and Rankings

In the simplest terms, a **rating** signifies "how much" people like something, and **ranking** proposes an ordering of opinions that can be used to evaluate which of two things is better.

A **rating** is a classification which represents the comparative assessment of one item against many others. Film critics often include a rating, such as "Two Thumbs Up" or "4/5 Stars," with their film reviews. A rating of "Two Thumbs Up" is just one outcome among "One Thumb Up," and "Two Thumbs Down." This can be represented mathematically as the range $0..2$. Alternatively, "4/5 Stars" is not as good as "5/5 Stars," but not as bad as "1/5 Stars," on the range $1..5$. Ratings can also be much simpler, such as the Facebook "Like" button which can only be true or false, $0..1$.

The most basic way for using collected ratings to inform users is to simply display each individual rating. When users browse rated content, such as products online, a selection of ratings can give them a general idea of the public reception for that product. Additionally, if the user knows other members of a site, ratings from those members can be brought to their attention. However, if two products are being directly compared, it is useful to develop a method of quantifying all of the ratings, in such a way that they can be quickly and easily compared to all the ratings of other products. This quantification governs the way that the products are ranked based on their ratings.

The **ranking** of a set is an ordering such that the first item in the set is the most preferred of all the items, and every item is preferred less or as much as the one before. The most common ranking of a set of rated content is by the arithmetic mean of all the ratings for a product. For example, if a product has ratings $2, 2, 3$, the mean would be 2.33, and that product would be ranked before another product with ratings $2, 2, 2, 3$ and a mean of 2.25.

3.2 Linear Prediction

Linear prediction is a way of predicting the future values of a signal. Generally, linear prediction is applied to *discrete time signals*, which do not have continuous values, but are instead a series of samples at a set interval. For example, time is continuous. When a singer sings, every instant, no matter how small, is different from the instant before it and the instant after it. However, when that song is recorded at 44,000 Hz, or one sample every 1/44,000th of a second, the resulting recording is a discrete time signal. A simpler discrete time signal would be temperature readings, sampled once every hour from the continuous values of a mercury thermometer.

3.2.1 Linear Predictive Coding (LPC)

One popular method of prediction is linear predictive coding, or LPC. LPC works by computing a set of n numbers to be coefficients, each of which are multiplied by the n previous values of a signal to predict the next value. This calculation can be represented by the following formula:

$$\widehat{x}(n) = \sum_{i=0}^{p} a_i x(n-i) \tag{2}$$

where $\widehat{x}(n)$ represents the predicted value, a_i represents the ith coefficient in the set, and $x(n-i)$ represents the ith previous value. The following the same equation in pseudo code:

$$Value[current+1] = \sum_{i=0}^{x} Coefficient[i] * Value[current-i] \tag{3}$$

For example, the coefficients $0.4, 0.6$ could be used to predict the next value in the signal $0, 2, 0, 2, 0, 2, 0, 2$ by multiplying the last value by 0.4, and the second-to-last value by 0.6 ($0.4 * 2 + 0.6 * 0$) to get a predicted value of 0.8. However, It is clear that the next value of that signal will almost definitely be 0, not 0.8. Better coefficients for this signal would be $0, 1$, which would completely disregard the last value and rely on only the second-to-last value, predicting correctly that the next value in the signal would be a 0.

The coefficients used are the key to the accuracy of this prediction. One method for fitting the model is to find the coefficients which can predict past values with the least sum of squared errors on a set of training data. This is expected to reduce the squared errors of the predictions on future data.

4 Related Work

There has been significant work on leveraging rated content online. Amazon product ratings, of one to five stars, are only one part of a user's review of a product. The full reviews can themselves be rated by other users as “Helpful” or “Unhelpful.” This allows Amazon to weed out unhelpful product reviews and highlight helpful product reviews. However, new reviews must be rated by users before Amazon knows if they are helpful or not. “Helpful or Unhelpful: a Linear Approach for Ranking Product Reviews” presents a model to predict the helpfulness of product reviews using the content of reviews which already have many ratings [2].

Similarly, “Ranking Comments on the Social Web” presents a machine learning-based approach for ranking comments based on community preferences [4]. Community preference is determined by a combination of user reputation, activity level, and metrics from the comment body text such as complexity.

“Building Reputation Systems for Better Ranking” presents another method of improving the usefulness of user ratings by calculating user reputation [5]. Through iterative refinement, object quality is determined by reputable users, and reputable users are determined by their ratings of object quality. In this way, the ratings of malicious users are marginalized while users with quality ratings are better utilized.

Another method of predicting the future popularity of content with few votes is to look at the patterns of vote accrual to predict future popularity. “Predicting the Popularity of Online Content” have looked at the early vote patterns of content, and compared that to new content, to predict future success [3].

5 Proposed Solution

This paper proposes predicting the future popularity of products based on existing ratings through linear prediction. By predicting the trend of future ratings, and accounting for that trend, products can be more accurately compared to one another.

The ideal scenario is a comparison between two products: *product* A with a high mean rating but a sharp downward trend, and *product B*, one with a lower mean rating and an upward trend. Ranking by mean would not account for the indication that the mean rating for *product B* will soon be higher.

To improve product ranking, we propose to predict future product ratings and include those values in the calculation of the product’s mean rating, such that the resulting mean is closer to the future mean than the current mean. For example, if a product currently has a mean rating of 4, which drops to 3 as more reviews accumulate, then a mean derived using predicted values should move the mean closer to 3 to be an improvement.

6 System

For our set of experiments, we used a dataset of 5.8 million Amazon product ratings from May 31st of 1996 to May 29th of 2006. The dataset was provided by Nitin Jindal and Bing Liu of the Social Media and Data Mining (SMDM) Lab at the University of Illinois at Chicago [6].

The dataset was corrected for minor errors and imported into a MySQL database using a set of Perl scripts. Corrections to the dataset and import scripts have been published to [7]. Each table in the dataset has a corresponding Perl script to parse the file, and insert that data into a corresponding MySQL database table, named in the form *TXT-to-MySQL_*.pl* in [7].

Scripts to convert elements from their string form, such as “Saturday, December 29, 2007,” to a easier to process format, such as “2007-12-29,” are named in the form *Variable_*.pl* in [7].

Perl scripts were written to generate signals, named in the form *Variable_linear_*.pl* in [7], and a Matlab script was written to predict future ratings based on the generated signal signal, named in the form *predict_rating.m* in [7].

7 Experiments

Products were chosen by their number of ratings, the proximity of their last rating to the end date of the dataset, and the average number of ratings per day. The first set of products was chosen to run to at least 9 days before the end of the dataset, contain ratings spanning over 200 days, and have an average of 2 or more ratings per day.

The first method of signal generation used is a signal composed of 200 days, of the mean rating for those 200 days. The first number in this signal was the mean of all the ratings written on the 200th day before the final date of the experiment, followed by the mean of all the ratings written on the 199th day before the final date of the experiment. This produces an extremely erratic signal which LPC predicted poorly. To smooth the signal, a running average was used, such that the signal advanced by one day for each element, but each element contains an average of the past three, ten, or thirty days. Many of the signals contain gaps in the records, where a product received no ratings for a short time period, which were replaced by a gradation between the two closest days which had ratings. The problem of more accurate interpolation of the signal is outside of the scope of this paper. To overcome this problem completely, a second method was derived to produce a continuous incrementing data set for analysis.

The second method uses 200 elements, each of which is the average of 10 ratings, incremented by 2 ratings for each element. The first element is the average of ratings $1 - 10$, the second of ratings $3 - 12$, and the third of ratings $5 - 14$, etc. This has the advantage of removing gaps in the data to fill, because as long as a product had over 210 ratings, it would be able to generate a signal of 200 elements. Additionally, it widens the range of products eligible for prediction, including those which had many ratings but a lower daily average of ratings.

8 Results and Evaluation

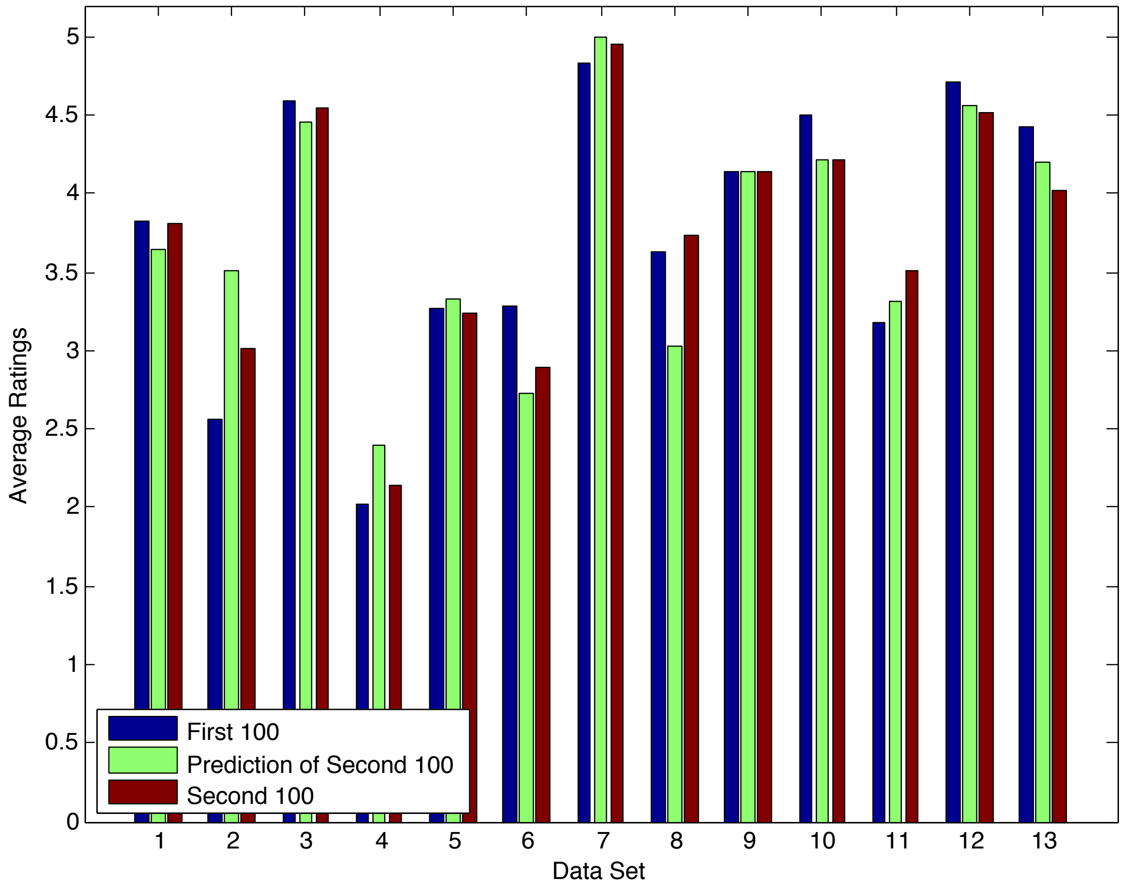
The first method of signal generation, an averaging of daily reviews, was not generating successful predictions. Considering that this may be caused by the gaps in the data or the way they were filled, a second signal generation method, which generates data by increments of reviews, has been developed and immediately yielded better results.

Figure 6–18 on pages 14–20 show the results of a LPC prediction with 100 coefficients, tested on elements 1-100 of the generated signal, and predicting elements 101-200. These individual predictions are summarized by figure 1 on page 8. Figures 2–4 on pages 9–11 show a similar summarization for LPC prediction for the same data sets under a number of conditions.

Figures 1–4 show the process of tweaking the conditions of the linear prediction towards significant improvement over the testing data. This concludes in the testing of the promising conditions from figure 4 with unrelated data in figure 5.

Figure 1 shows the mean of the training data (elements 1-100), testing data (elements 101-200), and prediction (prediction of 101-200) on those 13 products. The sum of differences between the mean of the training data and the mean of the testing data is 2.491. The difference between the mean of the testing

Figure 1: 100 Coefficients, Predicting Elements 100–200.



data and the mean of the predicted data is 2.456. This is a 1.42% improvement of the mean of the predicted data over the mean of the training training, in comparison to the mean of the testing data.

Figure 2: 92 Coefficients, Predicting Elements 100–171.

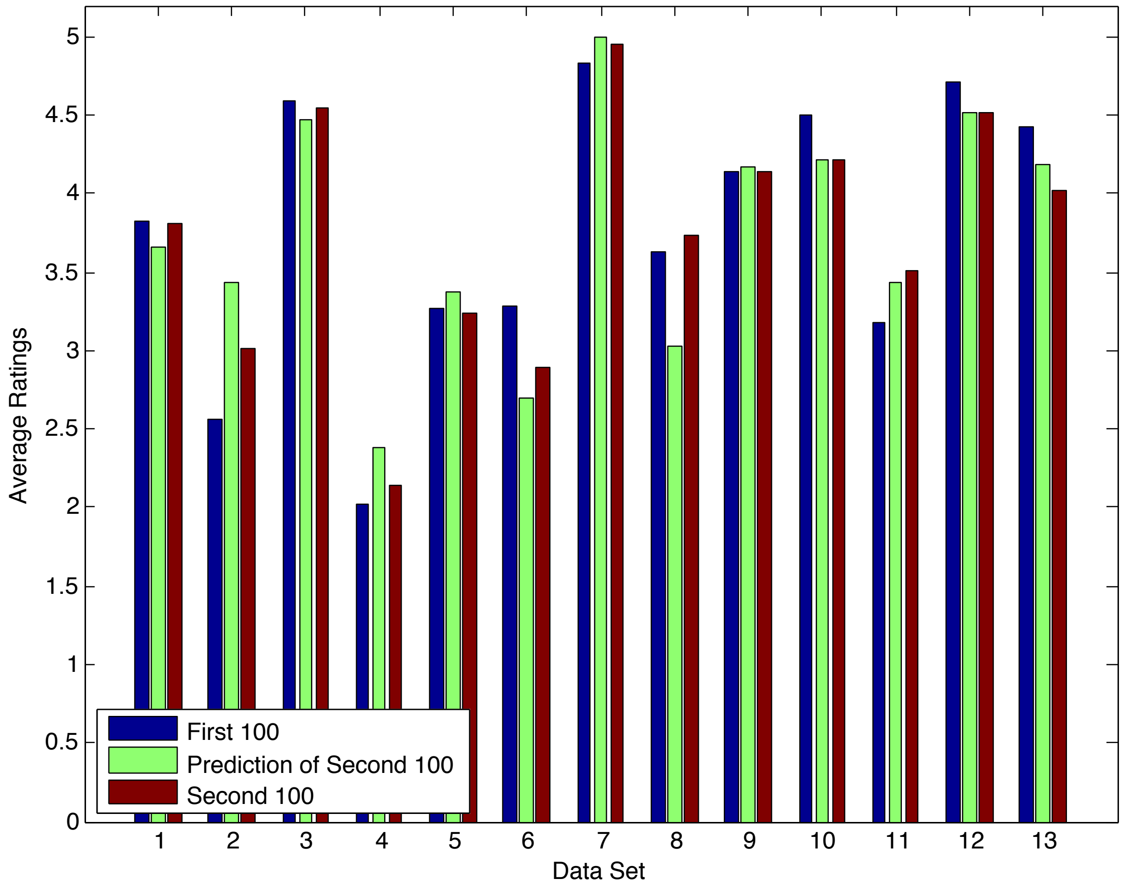


Figure 2 shows the mean of the training data, reduced testing data, and new prediction. The testing data includes the same elements 1–100. The a testing data has been reduced to elements 101-171. The prediction has also been reduced to elements 101–171, and uses 92 LPC coefficients instead of the 100 used previously. In this example, the difference between the mean of the testing data and the mean of the predicted data peaks at a 9.939% improvement.

Figure 3: 100 Coefficients, Mean of Elements 79–123.

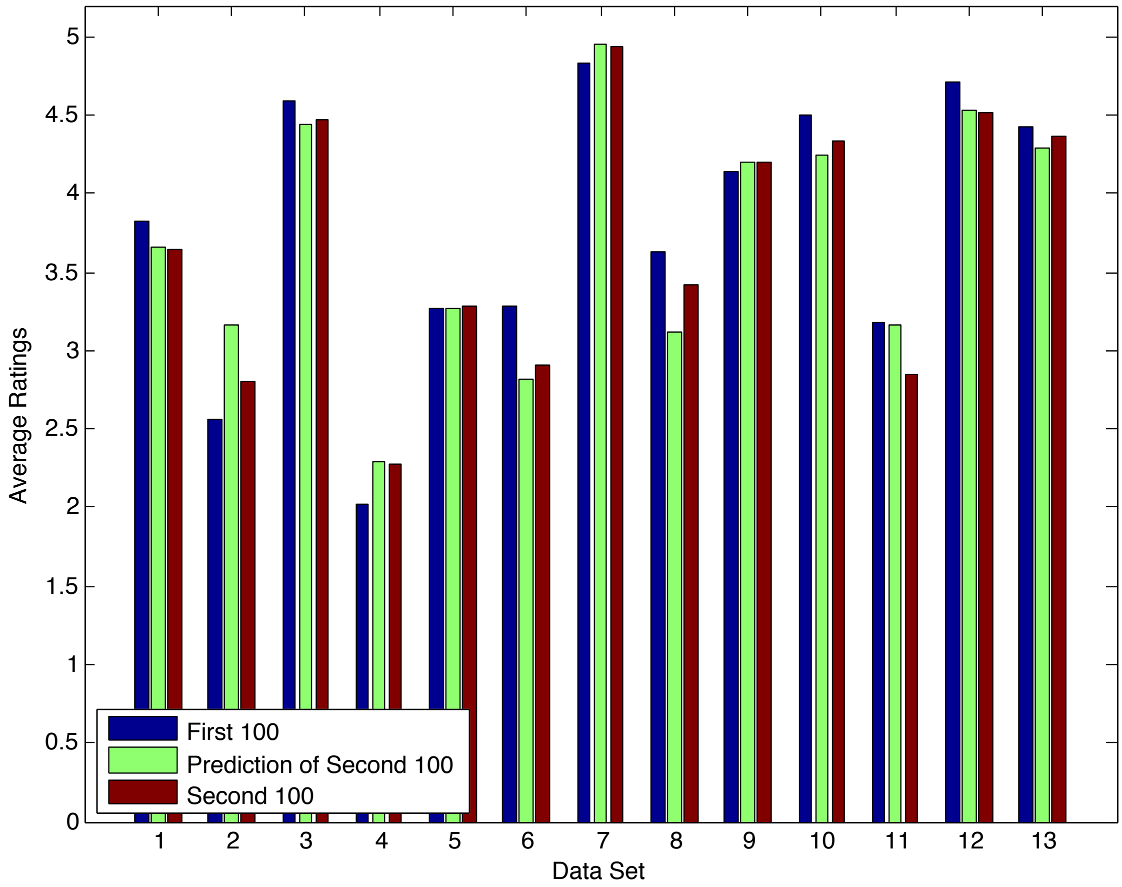


Figure 3 shows the mean of the training data, shifted testing data, and prediction. In this figure, elements within the training data were incorporated into the testing and prediction. This range of 79–123 yielded an improvement of 43% over the mean of the testing data alone.

Figure 4: 70 Coefficients, Mean of Elements 80–123.

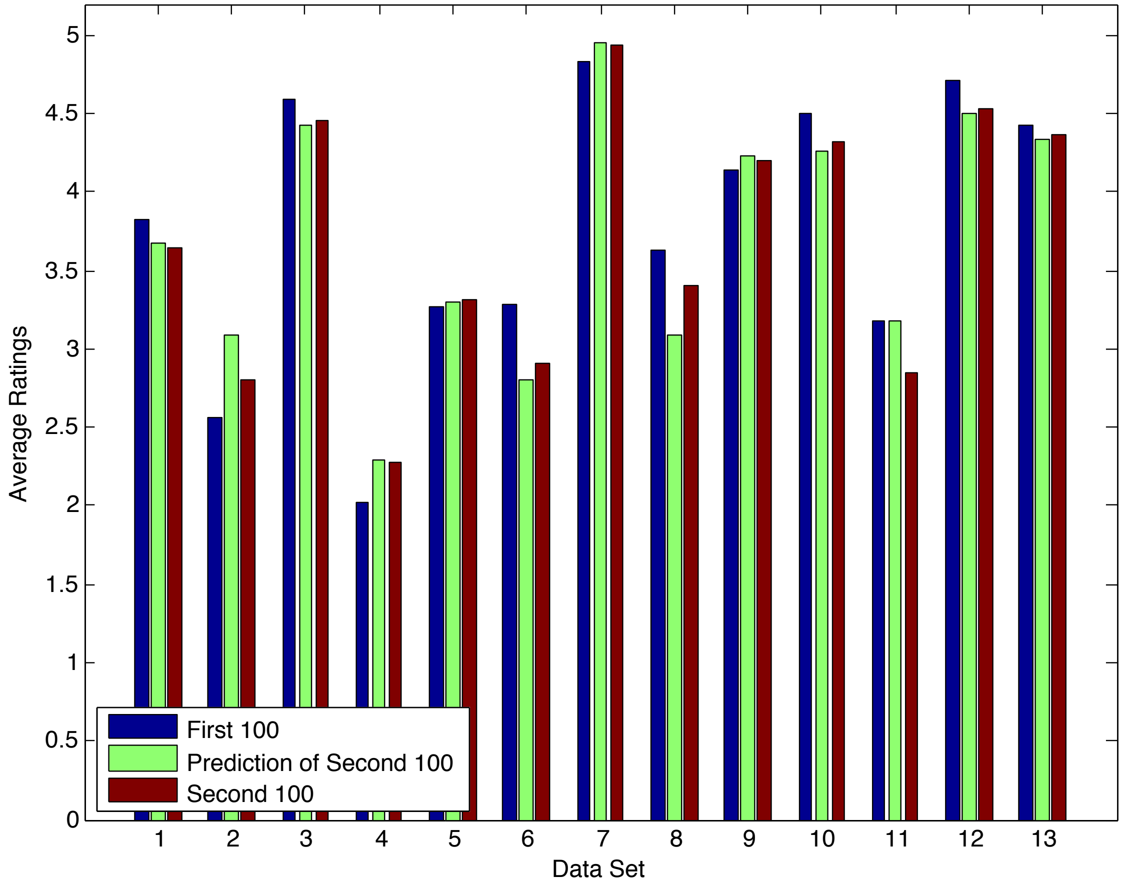


Figure 4 shows the mean of the training data, shifted testing data, and prediction. This prediction uses 70 coefficients, and predicts on the range 80–123. This prediction is an improvement of 44.1% over the mean of the testing data alone.

This set of coefficients and range is promising, but the improvement may simply be a result of checking for the best possible conditions for this data. The conditions must then be validated against unrelated data.

In order to validate these conditions against unrelated data, the same coefficients and range as used in figure 4 were applied to a second unrelated dataset of 40 products, with the criteria of over 400 ratings per product.

Figure 5: 70 Coefficients, Mean of Elements 80–123.

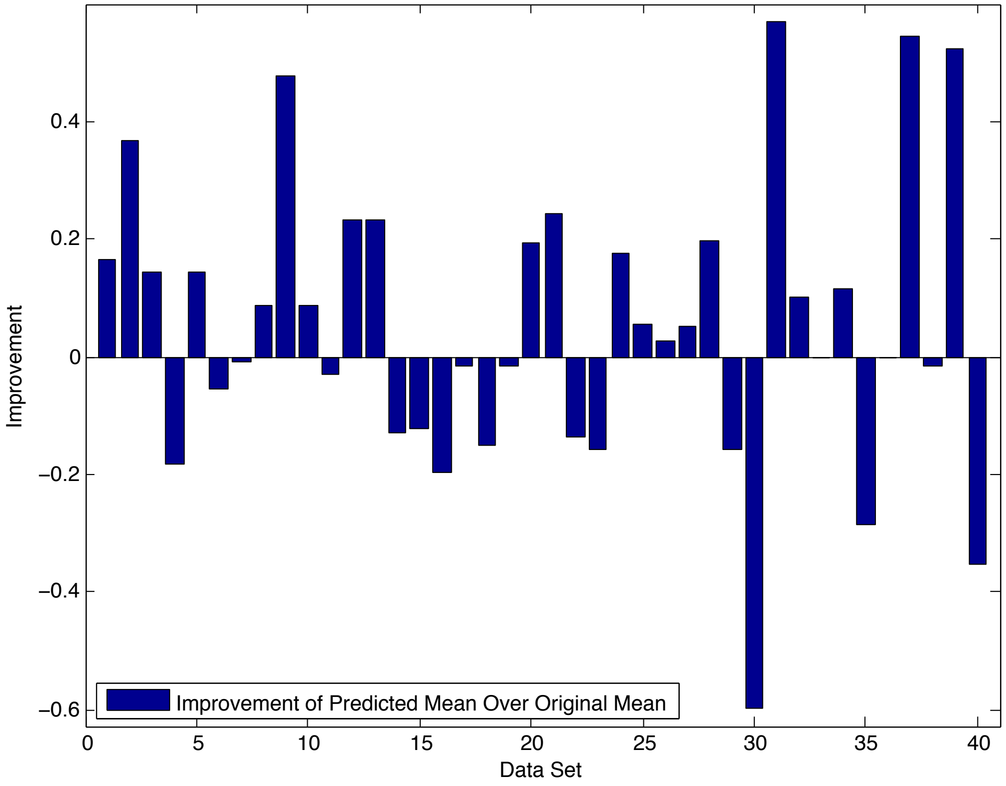


Figure 5 shows the improvement of the mean of the prediction over the mean of the testing data. Like figure 4, this prediction uses 70 coefficients, and predicts on the range 80–123. The prediction conditions continued to yield positive results even on unrelated data, with an improvement of 22.6%. However, the prediction performed better than the original data for only 55% of the 40 products. The 22.6% improvement on unrelated data represents the significant promise of this method.

9 Conclusion

Linear prediction using LPC was able to yield overall positive results. It was a significant improvement for many products, but in others the prediction was worse than the current mean of the ratings. LPC across any product data sets could not yield reliable improvement on a consistent basis.

Without improvements, LPC would not be useful in a production environment; where reliability is a considerable factor. Any LPC method would need to yield more successful predictions on a consistent basis to become an accepted method.

However, there is still much which can be done to improve the results. Significant improvements can be made through changes to both the method of signal generation and the use of LPC. Gaps in signal generation could be improved through more advanced interpolation techniques. Additionally, prediction methods other than LPC, including nonlinear regression, may yield better results.

Overall the development of robust methods for the ranking of rated content will only grow in importance, and we hope this work proves useful in that pursuit.

Figure 6: Amazon Product #006073132X. Mean of 10 ratings at 2 review increments.

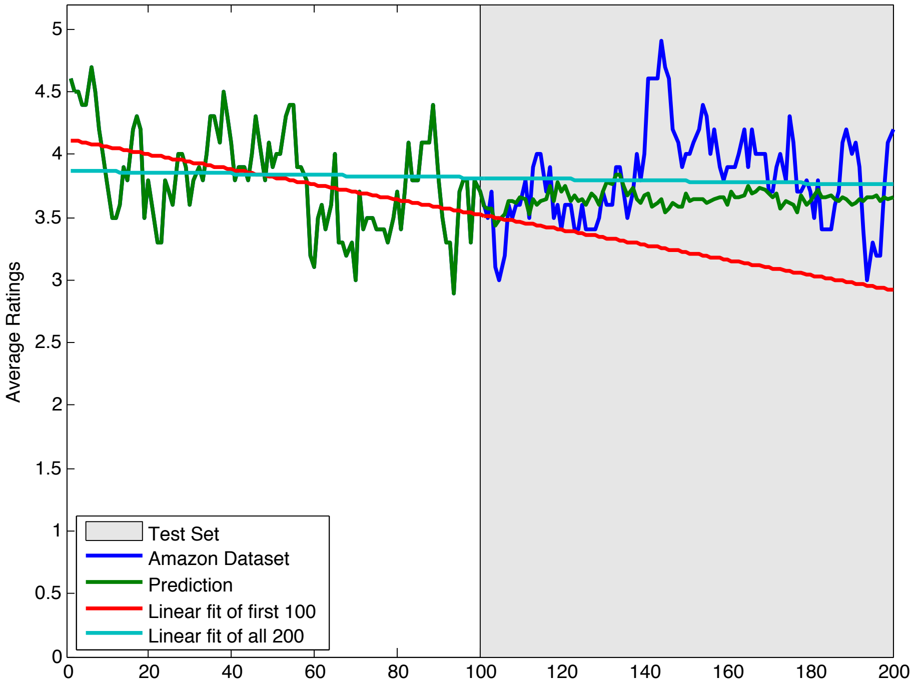


Figure 7: Amazon Product #0060761288. Mean of 10 ratings at 2 review increments.

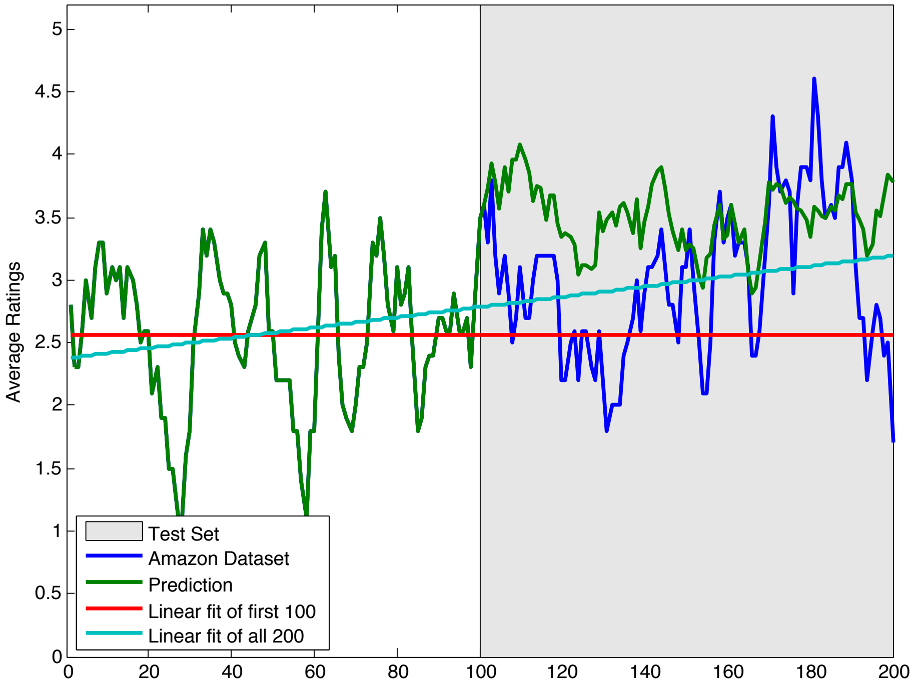


Figure 8: Amazon Product #0060817089. Mean of 10 ratings at 2 review increments.

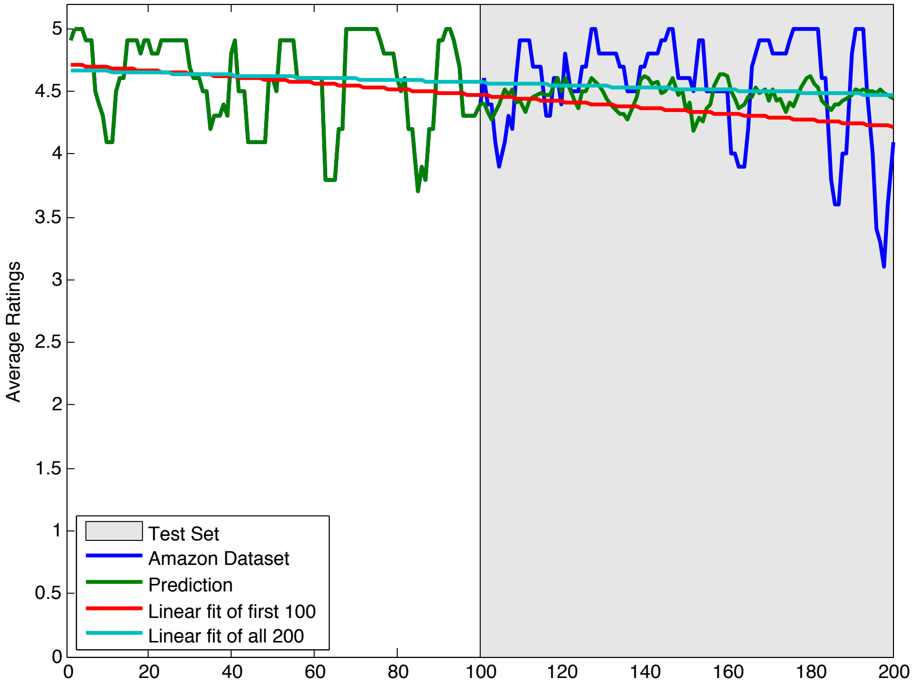


Figure 9: Amazon Product #0975599518. Mean of 10 ratings at 2 review increments.

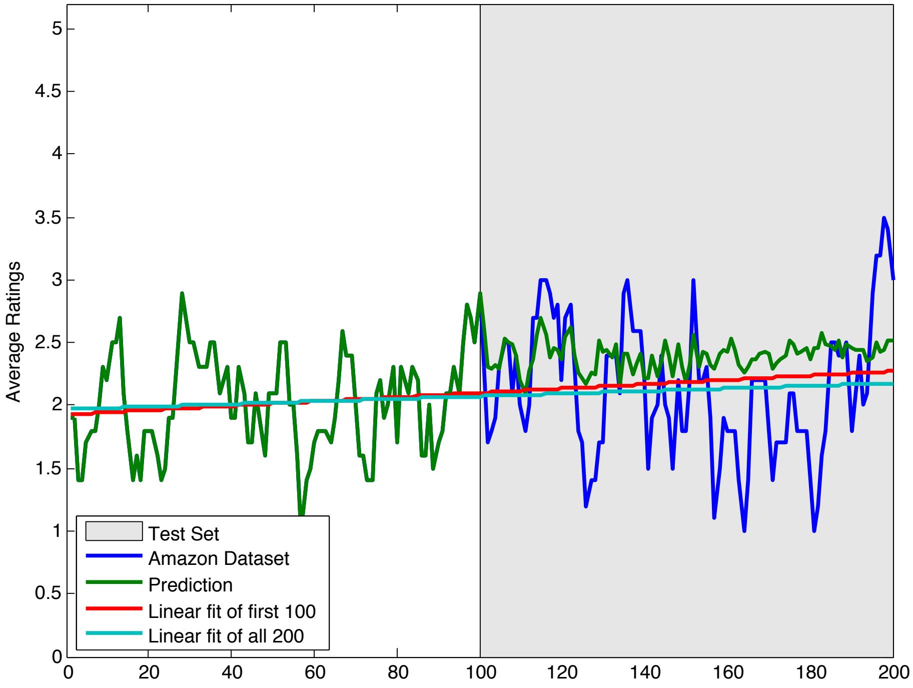


Figure 10: Amazon Product #0316011770. Mean of 10 ratings at 2 review increments.

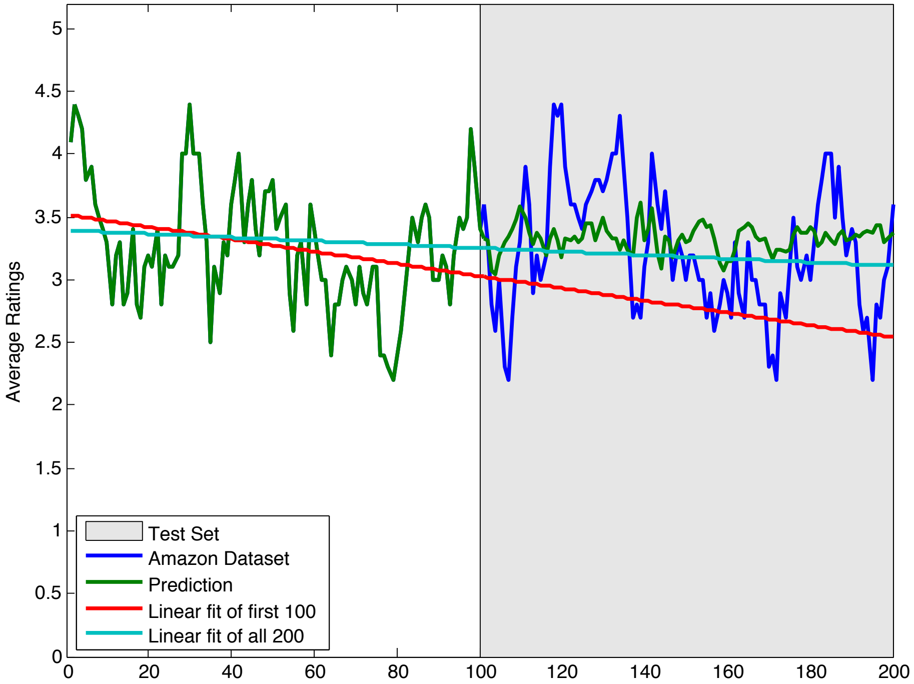


Figure 11: Amazon Product #B00008OWZG. Mean of 10 ratings at 2 review increments.

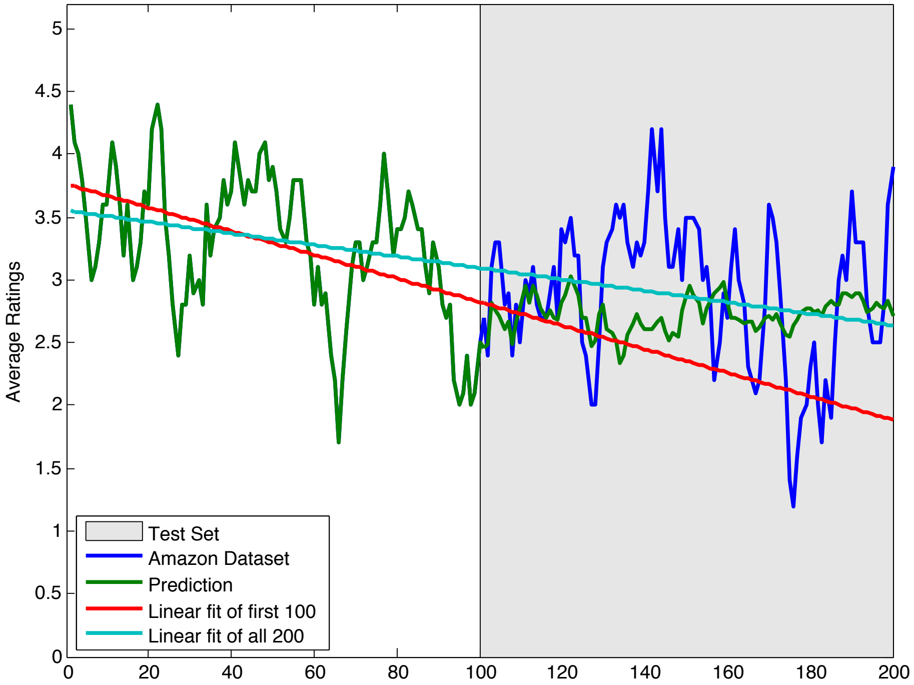


Figure 12: Amazon Product #B0000AQS0F. Mean of 10 ratings at 2 review increments.

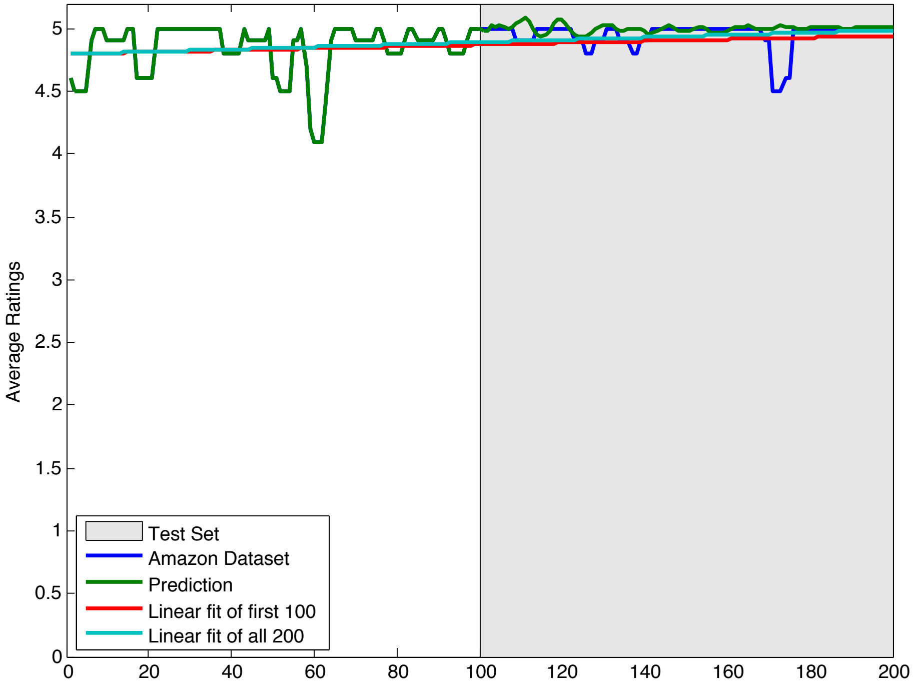


Figure 13: Amazon Product #B0006399FS. Mean of 10 ratings at 2 review increments.

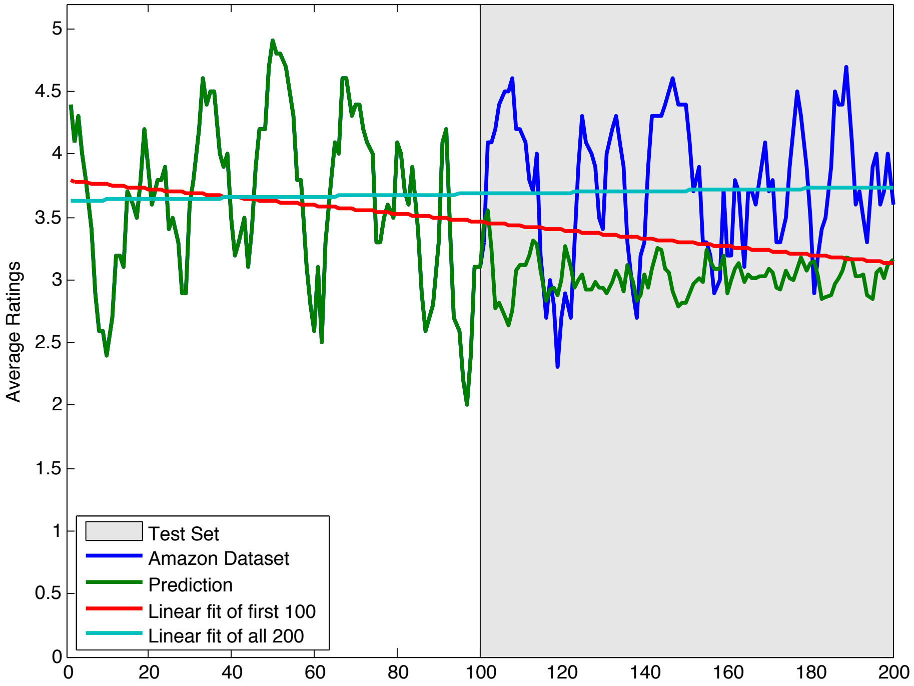


Figure 14: Amazon Product #B0006L16N8. Mean of 10 ratings at 2 review increments.

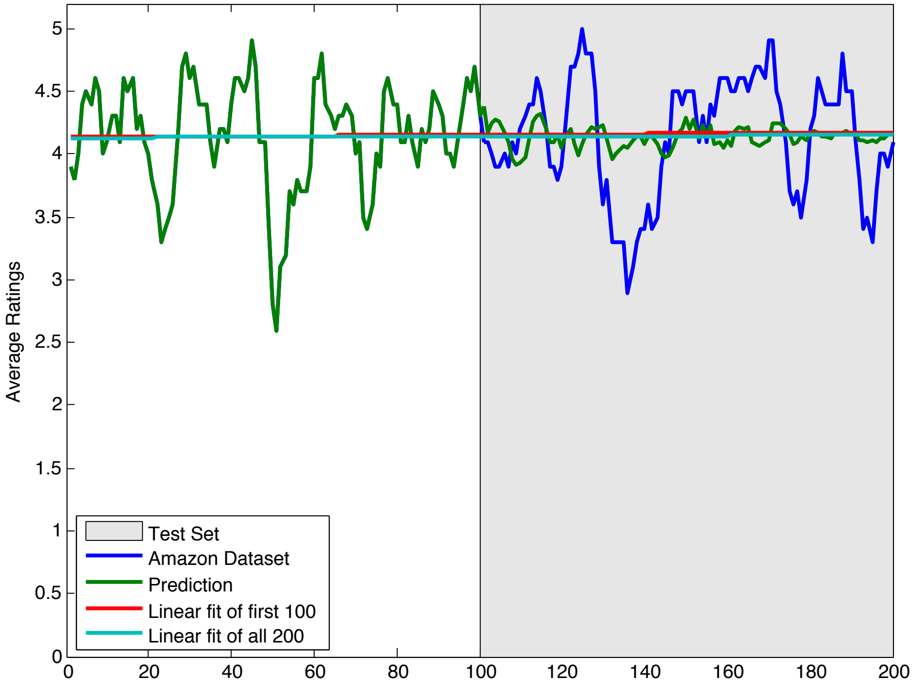


Figure 15: Amazon Product #B0007GAERG. Mean of 10 ratings at 2 review increments.

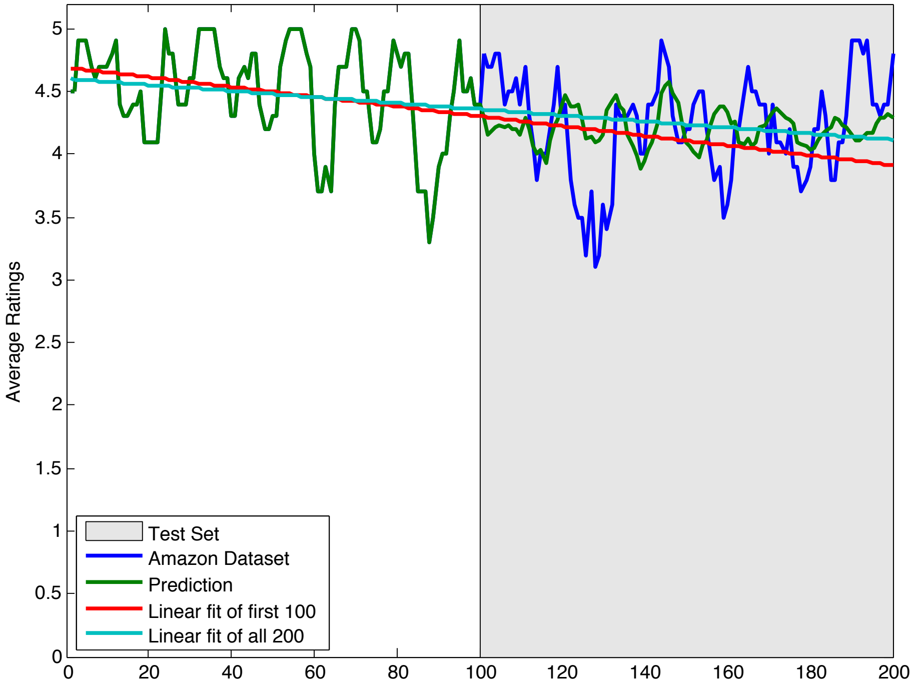


Figure 16: Amazon Product #037582670X. Mean of 10 ratings at 2 review increments.

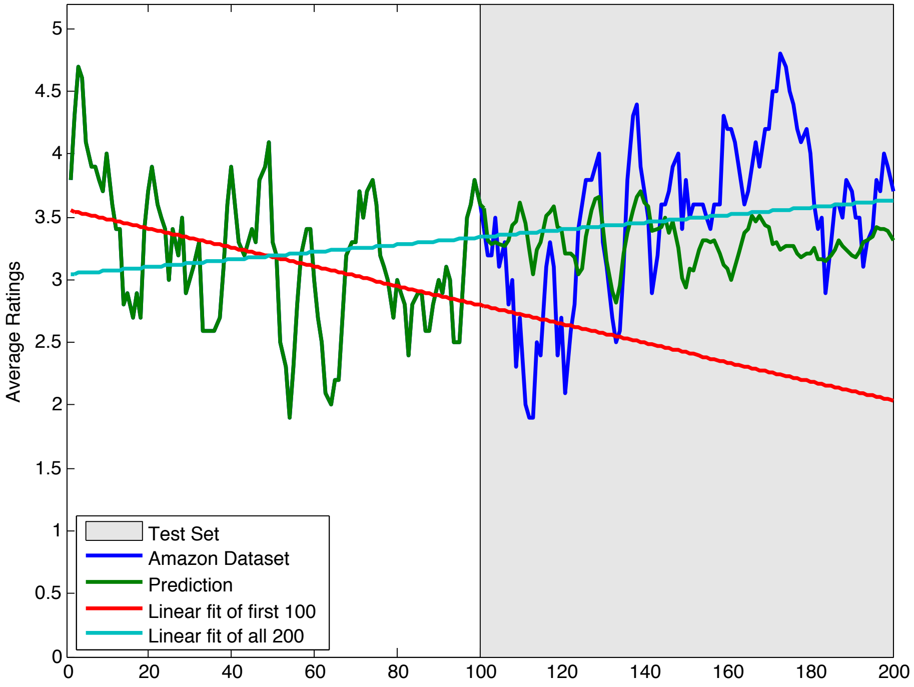


Figure 17: Amazon Product #B000BW7QWW. Mean of 10 ratings at 2 review increments.

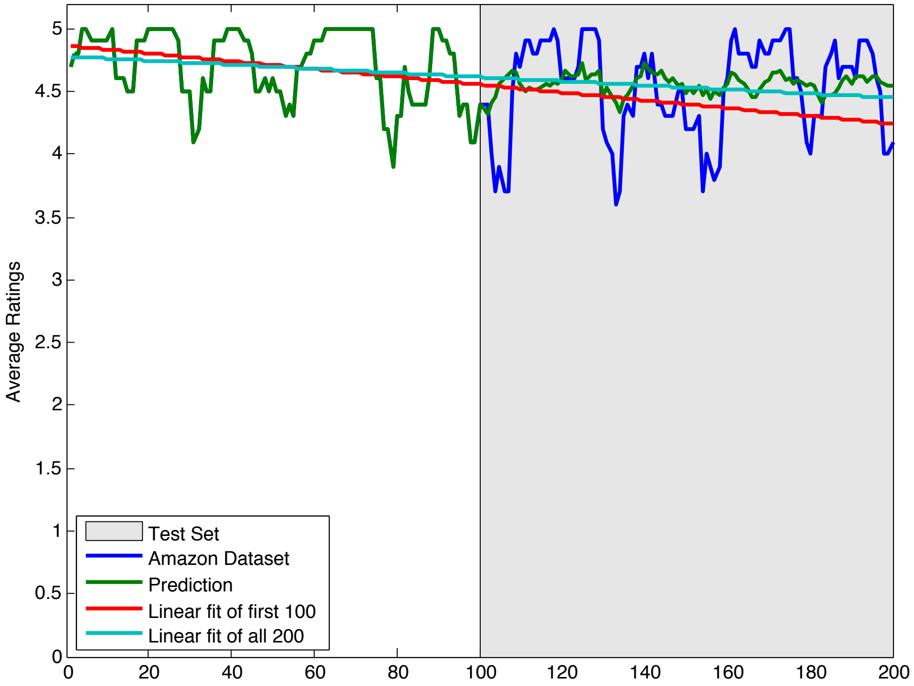
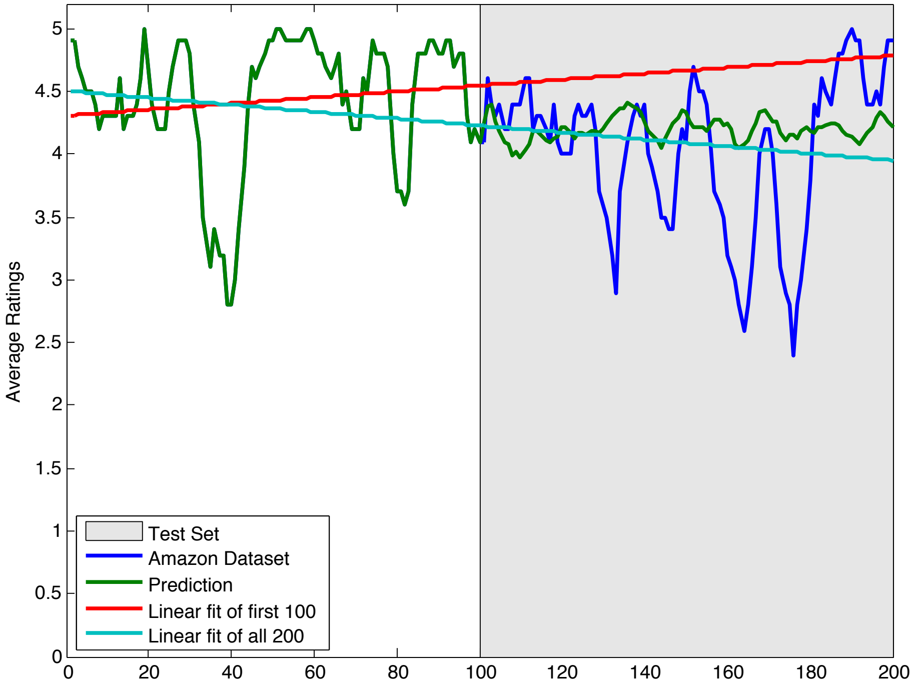


Figure 18: Amazon Product #0439784549. Mean of 10 ratings at 2 review increments.



References

[1] M. Wesch. (2007, Mar.) The machine is us/ing us. [Online]. Available: http://www.youtube.com/watch?v=NLlGopyXT_g

[2] R. C. Zhang and T. Tran, “Helpful or unhelpful: A linear approach for ranking product reviews.” *JOURNAL OF ELECTRONIC COMMERCE RESEARCH*, vol. 11, no. 3, pp. 220–230, 2010. [Online]. Available: http://www.csulb.edu/journals/jecr/issues/20103/paper3.pdf

[3] G. Szabó and B. A. Huberman, “Predicting the popularity of online content,” *CoRR*, vol. abs/0811.0405, 2008.

[4] C.-F. Hsu, E. Khabiri, and J. Caverlee, “Ranking comments on the social web,” in *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04*, ser. CSE ’09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 90–97. [Online]. Available: http://dx.doi.org/10.1109/CSE.2009.109

[5] L.-L. Jiang, M. Medo, J. R. Wakeling, Y.-C. Zhang, and T. Zhou, “Building reputation systems for better ranking,” *CoRR*, vol. abs/1001.2186, 2010.

[6] N. Jindal and B. Liu, “Opinion spam and analysis,” in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, ser. WSDM ’08. New York, NY, USA: ACM, 2008, pp. 219–230. [Online]. Available: http://doi.acm.org/10.1145/1341531.1341560

[7] R. Kuykendall. (2012, Dec.) Amazon data processing scripts. [Online]. Available: https://github.com/rkuykendall/Amazon-Data-Processing