



Universiteit Leiden

Opleiding Informatica

Rating-inference

Predicting the rating of online consumer reviews

Name: Thomas Prikkel
Date: 07/08/2015
1st supervisor: Michael Lew
2nd supervisor: TBD

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Online consumer reviews are a popular way to express ones opinion about a product. The opinions of others influence the purchase intentions of a customer. This paper studies the problem of predicting the human rating of a product based on a written online consumer review. This problem is called the rating-inference problem. We first improved the baseline algorithm to handle negation and by implementing feature selection. We also propose a novel extension to the Naive Bayes algorithm, which shows slight improvements in accuracy over the existing Naive Bayes algorithm.

Contents

1	Introduction	3
2	Related work	4
3	Algorithms	5
3.1	Baseline algorithm	5
3.2	Improved algorithm	7
3.2.1	Negation	7
3.2.2	Multinomial document model	7
3.2.3	Feature Selection	7
3.3	Extension	8
4	Experiment	8
4.1	Data set	9
4.2	Performance measures	9
4.2.1	Accuracy	9
4.2.2	F1-score	9
4.2.3	Mean absolute error and mean square error	10
4.3	Method	10
4.4	Results	11
5	Conclusion	15
6	Future work	15

1 Introduction

The opinion of other people has always been important when making decisions. Due to the Internet, the opinions of people not only spread among friends but also broadcasted in the form of online consumer reviews. Posting online reviews is a popular way to express feelings about a product.

Many studies have been conducted to research the impact of online word of mouth on purchase intentions. In a survey about online consumer reviews among 2000 U.S. Internet users. More than three-quarters of the users in nearly every category reported that the review had a significant influence on their purchase[1]. Zhu and Zhang studied the impact of online consumer reviews on sales in the gaming industry[19]. Their results indicate that on average, one point increase in average rating is associated with 4% increase in game sales. In addition, they found that for less popular games online consumer reviews are more influential. Their results suggest the importance of managing online consumer reviews for companies, especially for their less popular products.

Due to the influence on purchase intentions, analysis of online consumer reviews could produce useful and also actionable data that could be of economic value.

The rise of social media such as blogs and social networks has increased interest in sentiment analysis and opinion mining. Sentiment analysis attempts to identify the subjective sentiment expressed (or implied) in documents, such as consumer product reviews[4]. Companies want to analyze the opinions of the general public and therefore invest in solutions that help turn social media posts and online consumer reviews into actionable data.

This paper focuses on how products are rated by consumers when writing online reviews and how these ratings can be predicted. This problem is called the Rating inference problem. Rating inference is about determining the overall sentiment implied by the user, and map such sentiment onto some fine-grained rating scale. [5, 11] The rating of a review can be seen as the class in which a review could be classified using existing text classification algorithms. Many text classification algorithms use the Bag-of-words model to represent the text, where the frequency of occurrence of each word is used as a feature for training the algorithm. A con of this type of representation is that the order of the words in the text is lost.

We want to research whether online consumer reviews can be classified in the correct classes using Naive Bayes and with what accuracy, therefor the research question of this paper is:

Can online consumer reviews on Amazon.com be classified using Naive Bayes?

The remainder of this paper is organized as follows. Section 2 describes related work about the rating-inference problem. Section 3 describes the baseline algorithm, the improved algorithm and the proposed extension to the improved

algorithm. In section 4, we explain the experiment we did. Which data set and performance measures we used and our methods and the results of the experiment. The conclusion of this paper is in section 5. Section 6 contains the future recommendations.

2 Related work

This section briefly describes previous work about the rating-inference problem in chronological order. In 2002, Pang et al.[12] first addressed the specific classification task of determining what the opinion of a review actually is. They used Naive Bayes, Maximum Entropy and Support Vector Machine algorithms to decide whether a review is 'thumbs up' or 'thumbs down'. The Support Vector Machine scored the best in terms of performance with an accuracy ranging from 72.8% to 82.9%.

In 2005, Pang and Lee[11] gave the problem of determining an author's evaluation with respect to a multi-point scale the name Rating-inference problem. The data set they used contained reviews of four different authors, which they researched separately to factor out the effects of cross-author divergence. For example, the same word (e.g. "good") might indicate different sentiment strengths when written by different users. As figure 1 shows, the Positive Sentence Percentage increases when the rating is higher.

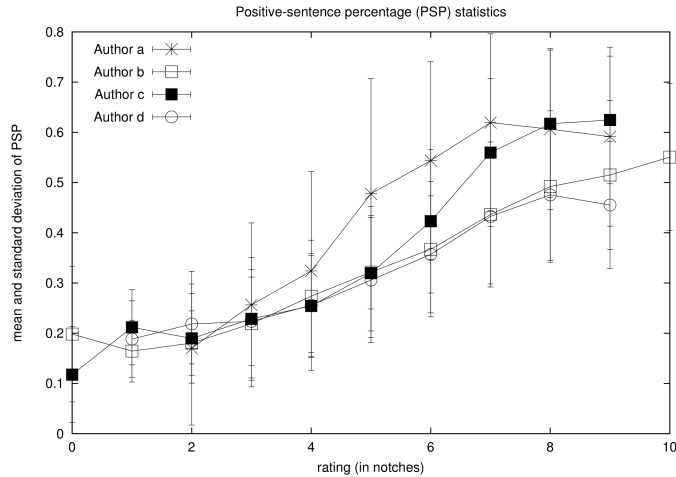


Figure 1: Cross-author divergence based on Positive Sentence Percentage[11]

They first studied the ability of humans to discern relative differences in opinion to set a reasonable classification granularity. Based on their findings then defined a three-class problem and a four-class problem, where the reviews would be classified into three and four classes, respectively. Results show that the algorithms provide a significant increase of accuracy over the simple baseline

of predicting the majority class, although the improvements are smaller for the four-class problem.

In 2006, Goldberg and Zhu[4] present a graph-based semi-supervised learning algorithm to address the sentiment analysis task of rating inference. Their paper describes an approach to the rating-inference problem when labeled data is scarce. The approach starts by creating a graph on both labeled and unlabeled data to encode certain assumptions for this task. They then solved an optimization problem to obtain a smooth rating function over the whole graph. Their algorithm performed significantly better when small amounts of labeled data is available but did not do as well with large labeled set sizes.

In 2008, Shimada and Endo[13] wrote a paper about a variant of the rating-inference problem where the document is not represented by just one rating, but a product is rated on multiple criteria. They compared a support vector machine, a linear support vector regression, a maximum entropy modeling algorithm and a similarity measure. The linear support vector regression performed best based on mean square error.

In 2011, Zhai et al.[18] studied the problem of grouping feature expressions. Feature expressions are words or phrases that express the same feature. For example, "Resolution" and "Number of pixels" both express the same feature of a screen. The proposed algorithm is superior to 13 baselines, which represent various current state-of-the-art solution for this class of problems.

In 2015, Liu et al.[8] propose a semantic-based approach for aspect level opinion mining. This type of rating-inference problem consists of two fundamental sub-tasks: aspect extraction (identify specific aspects of the product from reviews), and aspect rating estimation (offer a numerical rating for each aspect). The proposed approach performs well with an average deviation of around 1 star between the human rating and the estimated rating.

In 2015, Tang et al.[14] present a neural network method to solve the rating-inference problem. Neural network methods for sentiment prediction typically only capture the semantics of texts, but ignore the user who expresses the sentiment. With the proposed method Tang et al. address the issue of cross-author divergence by taking user information into account. The proposed method shows superior performances over several strong baseline methods.

3 Algorithms

3.1 Baseline algorithm

The baseline algorithm is based on Bayes' Theorem, using the Bernoulli document model as a representation for the reviews[10]. In the Bernoulli document model, a review is represented by a feature vector with binary elements taking

value 1 if the corresponding word is present in the review and 0 if the word is not present. The algorithm will classify review R , whose class is indicated by C , by finding the maximum posterior probability $P(C | R)$ using Bayes' Theorem:

$$P(C | R) = \frac{P(R | C)P(C)}{P(R)} \propto P(R | C)P(C) \quad (1)$$

In order to calculate the maximum posterior probability, the review likelihood $P(R | C)$ and the prior probability of class C need to be calculated. The algorithm uses labeled training data to calculate these probabilities using the feature vectors. Given b_i as the feature vector of a review R_i , the t th dimension of this vector, written b_{it} , corresponds to word w_t in vocabulary V . If we make the naive Bayes assumption, that the probability of each word occurring in the document is independent of the occurrences of the other words, then we can write the review likelihood $P(R_i | C)$ in terms of the individual word probabilities $P(w_t | C)$:

$$P(R_i | C) = \prod_{n=1}^{|V|} [b_{it}P(w_t | C) + (1 - b_{it})(1 - P(w_t | C))] \quad (2)$$

As mentioned above, the labeled training data is used to calculate prior class probability and word probability. In order to calculate the word probability we define $P(c_j | R_i)$ as follows:

$$P(c_j | R_i) = \begin{cases} 1, & \text{if } R_i \text{ is in class } c_j. \\ 0, & \text{if } R_i \text{ is not in class } c_j. \end{cases} \quad (3)$$

We use $P(c_j | R_i)$ to estimate the probability of word w_t in class c_j :

$$P(w_t | c_j) = \frac{\sum_{i=1}^{|R|} b_{it}P(c_j | R_i)}{\sum_{i=1}^{|R|} P(c_j | R_i)} \quad (4)$$

The prior class probability is the fraction of reviews belonging to a certain class in the labeled training data, which we calculate for every class using:

$$P(c_j) = \frac{\sum_{i=1}^{|R|} P(c_j | R_i)}{|R|} \quad (5)$$

Using (1), (2), (3) and (5), the maximum posterior probability is calculated and the algorithm classifies the review in the most probable class.

3.2 Improved algorithm

In this section the improvements made to the baseline algorithm are described.

3.2.1 Negation

The baseline algorithm does not model the contextual effect of negation by only storing a negation word as a word in the feature vector. In order to improve the algorithm we adapted a technique by Das and Chen[3]. Every word between a negation word ("Not", "Doesn't", "Couldn't", etc.) and the first following punctuation mark was prefixed with NOT_ and being stored as a separate word in the feature vector. For example, "I didn't like this product." would be transformed into "I didn't NOT_like NOT_this NOT_product.". This technique creates a larger feature vector. Therefore, it does have an impact on the performance of the algorithm.

3.2.2 Multinomial document model

The baseline algorithm represents the review as a binary feature vector where only the presence or absence of a word is captured. The Multinomial document model also captures the frequency of a word. b_{it} represents the frequency of the w_t in review R_i . For the algorithm to take the frequency into account, some of the formulas had to be redefined. The review likelihood $P(R_i | C)$ can be formulated as:

$$P(R_i | C) = \prod_{n=1}^{|V|} P(w_t | C)^{b_{it}} \quad (6)$$

Using $P(c_j | R_i)$ as defined in (3), the probability of word w_t in class c_j can be calculated as the relative frequency of w_t in reviews of class c_j with respect to the total number of words in reviews of that class:

$$P(w_t | c_j) = \frac{\sum_{i=1}^{|R|} b_{it} P(c_j | R_i)}{\sum_{s=1}^{|V|} \sum_{i=1}^{|R|} b_{is} P(c_j | R_i)} \quad (7)$$

By using (1), (6), (7) and (5) the maximum posterior probability is calculated.

3.2.3 Feature Selection

In order to decrease the size of the vocabulary we used feature selection. We used a technique called Document Frequency Thresholding[17, 20]. The document

frequency of a word is the number of reviews a word occurs in. The document frequency of each unique word in the training set is computed and the words with a document frequency lower than the predetermined threshold are removed from the vocabulary. Although this feature selection technique does not have accuracy optimization as the main goal. It could increase accuracy due to rare terms that produce noise are removed from the vocabulary.

3.3 Extension

Since the classes of this problem are ordinal, we tested a new extension to the algorithm. When calculating the maximum posterior probability of a class, we wanted surrounding classes of this class influence the outcome of this calculation. Therefore, based on the Multinomial document model we tested an alternative method to calculate the review likelihood ($P(R_i | C)$) to include the probabilities of the surrounding classes. For the extension, we redefine formula (6) as:

$$P(R_i | C_j) = \prod_{n=1}^{|V|} (0.8 * P(w_t | C_j) + 0.1 * P(w_t | C_j - 1) + 0.1 * P(w_t | C_j + 1))^{b_{it}} \quad (8)$$

This formula works for class 2 through 4 since these have two surrounding classes. Class 1 and 5 only have one surrounding class. Therefore, for class 1 and 5 we formulas (9) and (10), respectfully.

$$P(R_i | C_j) = \prod_{n=1}^{|V|} (0.8 * P(w_t | C_j) + 0.2 * P(w_t | C_j + 1))^{b_{it}} \quad (9)$$

$$P(R_i | C_j) = \prod_{n=1}^{|V|} (0.8 * P(w_t | C_j) + 0.2 * P(w_t | C_j - 1))^{b_{it}} \quad (10)$$

4 Experiment

In this section, we explain our data set and methods for the experiment. Then we discuss the experimental results.

4.1 Data set

A data set from Amazon.com is used. The data set contains 147.3 million reviews spanning May 1996 through July 2014[9]. The data set is split into 24 separate product categories. For the purpose of this research we will look at four product categories: *Electronics*, *Office products*, *Sports and Outdoors* and *Automotive*. Each review in the data set contains a number of headers and a text body. The headers include the rating, the helpfulness rating of the review, a product ID, a reviewer ID, a summary of the review, the name of the reviewer and the date and time of the review. The rating is a positive integer ranging from 1 to 5. Figure 2 shows the ratings distribution of the data. More than 50% of the reviews have the rating of 5.

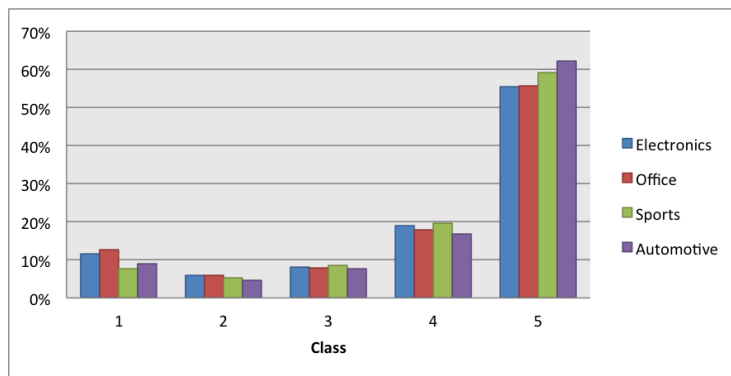


Figure 2: Five-class ratings distribution

4.2 Performance measures

In order to compare the algorithm, performance measures are used. The classification results of the algorithms can be displayed in a confusion matrix. Using the confusion matrix the following metrics can be calculated to compare the classifier.

4.2.1 Accuracy

The accuracy is the percentage of correctly classified reviews. Although this performance measure does not take into account how much a classifier is off when a review is misclassified, it is a good measure to start comparing with.

4.2.2 F1-score

The first performance measure we used to compare the classifiers is the F1-score.[16] The F1-score is the harmonic mean of the precision and recall. Using

the F1-score the performance of the classifier for each individual class can be measured.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Where the precision and recall can be calculated for each class using the amount of True Positives, False Positives and False Negatives.

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

4.2.3 Mean absolute error and mean square error

The Mean absolute error(MAE) and mean square error(MSE) are used to measure the performance of ordinal classifiers. [2] These performance measures take into account the amount of classes the predicted class is off of the actual class.

$$MAE = \frac{1}{N} \sum_{r=1}^K \sum_{c=1}^K n_{r,c} |r - c|$$

$$MSE = \frac{1}{N} \sum_{r=1}^K \sum_{c=1}^K n_{r,c} (r - c)^2$$

With $n_{r,c}$ representing the number of reviews from the r th class predicted as being from c th class.

4.3 Method

We will look at two different problems,(1) a five-class problem and (2) a three-class problem. The five-class problem will classify based on the original five stars rating. To create the three-class problem from our data we transformed our five stars rating to three classes. The original rating of a review d , denoted by $r(d)$, was transformed into $r'(d)$ as follows:

$$r'(d) = \begin{cases} 1, & \text{if } r(d) = 1. \\ 2, & \text{if } 2 \leq r(d) \leq 4. \\ 3, & \text{if } r(d) = 5. \end{cases}$$

For each of the four product categories, *Electronics*, *Office products*, *Sports and Outdoors* and *Automotive*, we ran the algorithm on different data set sizes. We use 5-fold cross-validation on the data set to test the algorithms on both problems. 80% of the reviews are used to train the algorithm and 20% to test, these sets of reviews do not overlap. Also the data set of 20000 reviews used to test the algorithm does not overlap with the data set of 10000 reviews. The performance measures for both algorithms are compared. To optimize feature selection we ran the algorithm using multiple thresholds to determine the optimal parameter, this is the threshold we used to compute the results of the improved algorithm.

4.4 Results

In this section we will present the results from our experiment. First we determined the optimal threshold. The results per threshold can be found in Figure 3. We picked 60 to be the threshold in our experiment, since this is the first local maximum and to avoid losing many important features we did not select a higher threshold.

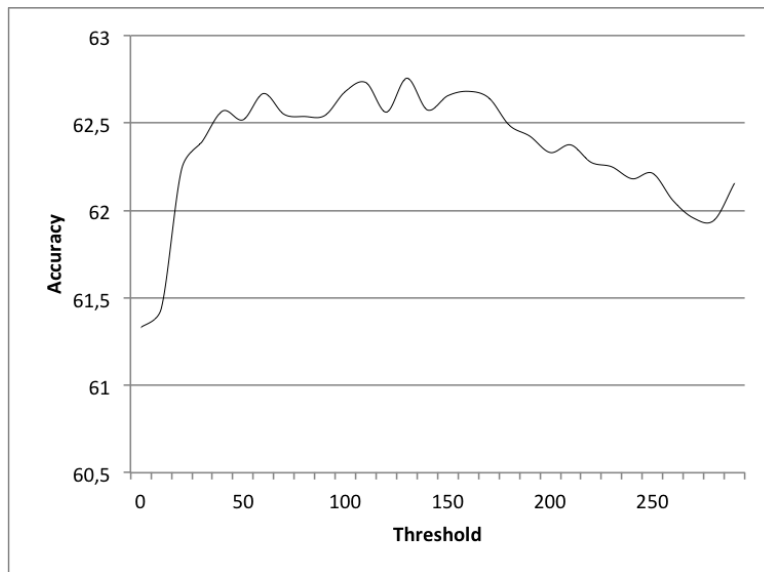


Figure 3: Average accuracy per threshold

Table 1 shows the difference in vocabulary size and execution time, based on tests using 20000 reviews. Implementing document frequency thresholding decreased the average vocabulary size with 33%. This decrease in vocabulary size also leads to a significant decrease in execution time.

	Vocabulary size	Execution time (s)
Without feature selection	327084	206,922
With feature selection	219959	8,464

Table 1: Feature selection performance and vocabulary comparison

Tables 2 through 5 display the MAE and MSE of the algorithms. The results are displayed separate for the five-class problem and three-class problem. These results do not include the proposed extension, which is shown separately in Table 9. Shown in bold is the highest value for each of the performance measures. The improved algorithm performs better than the baseline algorithm in both the 3-class and 5-class problem. Since the data is not evenly distributed the majority has a relatively high accuracy (53,7% - 67,7%). Therefore, the majority also outperforms the baseline algorithm. For the 5-class problem with 20000 reviews in the data set the MAE and MSE of the improved algorithm are the lowest.

	majority			baseline			improved		
	MAE	MSE	%	MAE	MSE	%	MAE	MSE	%
Electronics	1,063	3,187	53,745	1,474	4,752	44,040	0,778	2,800	56,425
Office	1,015	3,051	56,360	1,324	4,299	48,130	0,735	1,863	60,600
Sports	0,645	1,740	67,695	1,151	3,713	55,385	0,607	1,476	65,545
Automotive	0,848	2,441	61,019	1,266	4,046	50,515	0,641	1,543	63,655

Table 2: 5-class problem results with 20000 reviews in the data set

	majority			baseline			improved		
	MAE	MSE	%	MAE	MSE	%	MAE	MSE	%
Electronics	0,597	0,865	53,745	0,713	1,068	46,370	0,413	0,533	64,710
Office	0,564	0,818	56,360	0,644	0,981	50,990	0,390	0,502	66,555
Sports	0,386	0,512	67,695	0,576	0,879	57,540	0,336	0,423	70,730
Automotive	0,486	0,679	61,019	0,623	0,938	53,399	0,346	0,430	69,655

Table 3: 3-class problem results with 20000 reviews in the data set

	majority			baseline			improved		
	MAE	MSE	%	MAE	MSE	%	MAE	MSE	%
Electronics	0,831	2,222	58,240	1,717	5,723	38,320	0,866	2,298	55,500
Office	0,989	2,951	56,700	1,474	4,830	45,270	0,764	1,934	59,380
Sports	0,807	2,216	60,830	1,458	4,774	45,760	0,655	1,533	61,590
Automotive	0,829	2,330	61,120	1,412	4,587	46,770	0,673	1,580	61,420

Table 4: 5-class problem results with 10000 reviews in the data set

	majority			baseline			improved		
	MAE	MSE	%	MAE	MSE	%	MAE	MSE	%
Electronics	0,495	0,650	58,240	0,822	1,296	38,320	0,455	0,612	62,410
Office	0,555	0,798	56,700	0,721	1,106	47,200	0,400	0,505	65,290
Sports	0,472	0,631	60,830	0,709	1,091	48,250	0,363	0,446	67,780
Automotive	0,475	0,679	61,200	0,678	1,034	50,040	0,357	0,428	67,880

Table 5: 3-class problem results with 10000 reviews in the data set

Figure 4 shows us that, in the four product categories we tested the algorithms on, the average accuracy of the improved algorithm is higher than the accuracy of the baseline algorithm. However, it is only better than the majority baseline 3 out of 4 times.

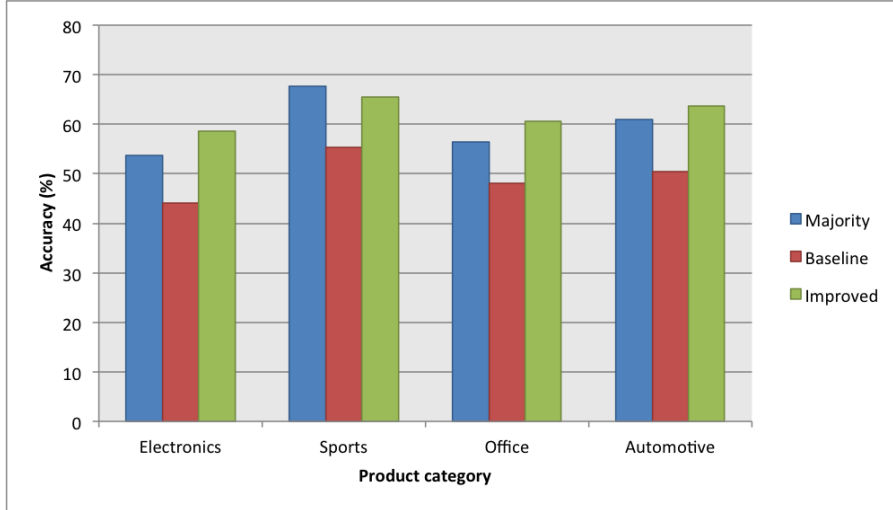


Figure 4: Algorithm accuracy per product category

Tables 6 and 7 show the F1 score per class for each of the product categories. The improved algorithm has a higher F1-score for every class in each product category. The F1-scores for class 1 and class 5 are significantly higher than the scores of class 2 through 4. Reason for these differences in F1-score could be that the data set is not evenly distributed. Only 5% of the reviews in the data set are in class 2, on average. Which means out of 16000 reviews in one fold, only 800 reviews can be used to train the algorithm to predict this class. Identical for class 3 and 4, with 8% and 18% of the reviews.

	Electronics		Sports		Office		Automotive	
	Baseline	Improved	Baseline	Improved	Baseline	Improved	Baseline	Improved
1	0,306	0,541	0,192	0,524	0,325	0,544	0,248	0,446
2	0,037	0,143	0,015	0,152	0,022	0,197	0,031	0,173
3	0,020	0,148	0,054	0,204	0,054	0,192	0,039	0,176
4	0,102	0,299	0,108	0,305	0,104	0,251	0,128	0,311
5	0,639	0,763	0,746	0,803	0,649	0,782	0,702	0,815

Table 6: F1 scores for the 5-class problem with 20000 reviews in the data set

	Electronics		Sports		Office		Automotive	
	Baseline	Improved	Baseline	Improved	Baseline	Improved	Baseline	Improved
1	0,297	0,545	0,189	0,455	0,319	0,565	0,321	0,520
2	0,220	0,461	0,234	0,474	0,253	0,453	0,439	0,502
3	0,640	0,768	0,748	0,820	0,682	0,784	0,800	0,807

Table 7: F1 scores for the 3-class problem with 20000 reviews in the data set

Figure 5 shows us the F1-score per class of the baseline and the improved algorithm. For every class we see an increase in F1-score, which means the improved algorithm performs predicts every class more accurately than the baseline.

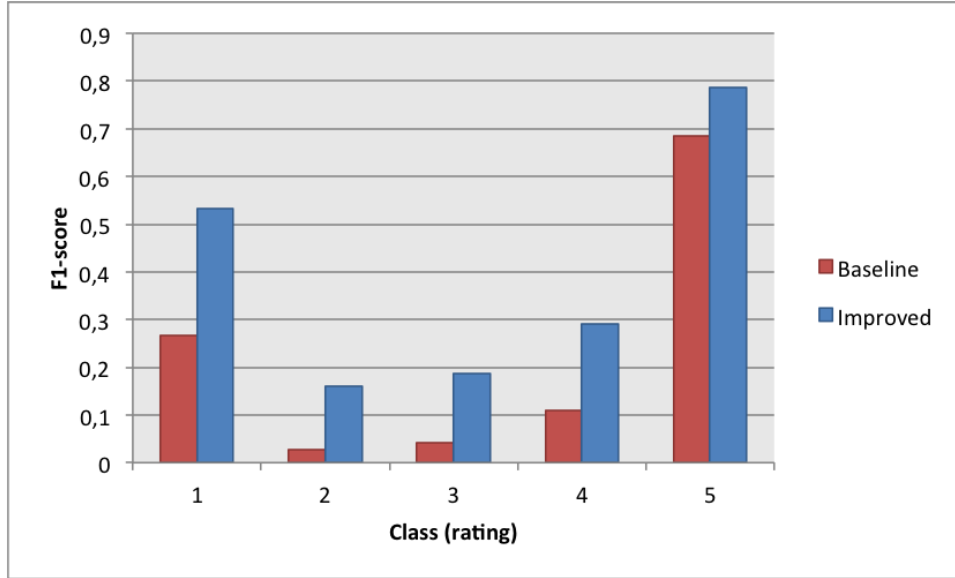


Figure 5: Average F1-scores per class for the baseline and improved algorithms

Table 8 shows the results of the proposed extension next to the results of the improved algorithm. Although the results show an increase in accuracy, the MSE of the algorithm with the extension is always lower.

	Improved			Improved + Extension		
	MAE	MSE	%	MAE	MSE	%
Electronics	0,778	1,990	58,605	0,780	2,047	59,895
Office	0,735	1,863	60,600	0,728	1,879	62,050
Sports	0,607	1,476	65,545	0,591	1,495	68,075
Automotive	0,641	1,543	63,655	0,638	1,596	65,315

Table 8: Improved + Extension results tested on 5-class problem with 20000 reviews

Figure 6 shows the F1-scores of the improved algorithm and F1-scores of the proposed extension. The F1-scores from the proposed extension are higher for class 1 and 5. For class 2 through 4 the F1-scores of the extension are lower.

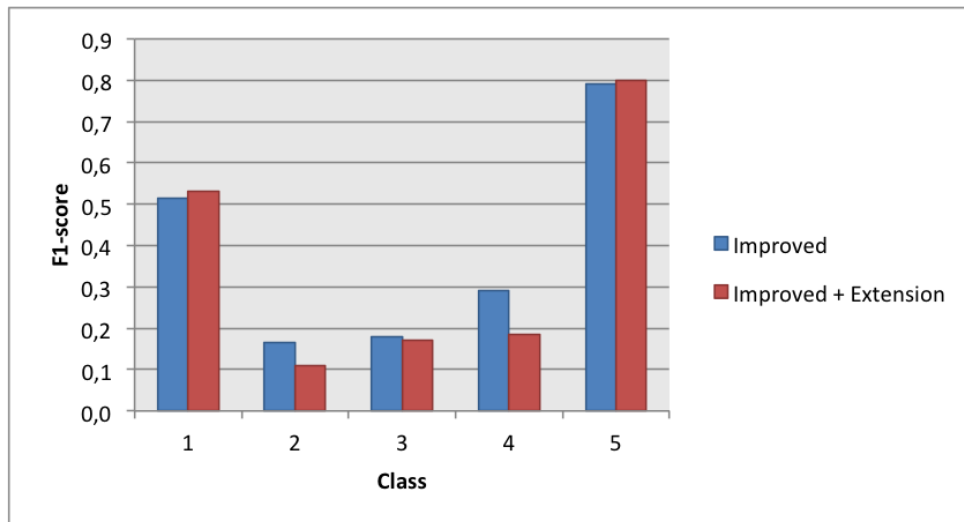


Figure 6: Average F1-scores per class for the Improved and Improved + Extension algorithms

5 Conclusion

As shown in the previous section, the improved algorithm outperforms the baseline. The improved algorithm has the lowest MAE and MSE for both the 5-class and 3-class problem when 20000 reviews are in the data set used to run the algorithm. The extension shows an increase in accuracy but a decrease in MSE when compared to the improved algorithm. The average F1-score of the extension is lower than the average F1-score of the improved algorithm.

The main contribution of this work is the proposed extension to the naive bayes algorithm which takes the scores of surrounding classes into account. Although the extension does show an increase in performance based on accuracy, the performance based on MSE decreased.

6 Future work

For future work, it would be interesting to test multiple feature selection methods as seen in [7] to further improve performance. Another topic could be to see how the size of the data set influences the performance of the algorithm. We only showed the difference between 20000 reviews and 10000 reviews. A different part of research dedicated to the rating-inference problem looks at semantic approaches using adjective-noun word pairs to select features. With these types of algorithms the run time is much longer, because all the sentences have to be parsed and parts-of-speech have to be tagged. Research into this area does show promising results.

References

- [1] Online Consumer-Generated Reviews Have Significant Impact on Offline Purchase Behavior. <http://www.comscore.com/dut/Insights/Press-Releases/2007/11/Online-Consumer-Reviews-Impact-Offline-Purchasing-Behavior>.
- [2] J. S. Cardoso and R. Sousa. Measuring the performance of ordinal classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(08):1173–1195, 2011.
- [3] S. Das and M. Chen. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific finance association annual conference (APFA)*, volume 35, page 43. Bangkok, Thailand, 2001.
- [4] A. B. Goldberg and X. Zhu. Seeing Stars when There Aren’T Many Stars: Graph-based Semi-supervised Learning for Sentiment Categorization. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing, TextGraphs-1*, pages 45–52, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.
- [5] C. W. Leung, S. C. Chan, and F.-l. Chung. Integrating collaborative filtering and sentiment analysis: A rating inference approach. In *Proceedings of the ECAI 2006 workshop on recommender systems*, pages 62–66. Citeseer, 2006.
- [6] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, pages 342–351. ACM, 2005.
- [7] M. Liu, X. Lu, and J. Song. A New Feature Selection Method for Text Categorization of Customer Reviews. *Communications in Statistics - Simulation and Computation*, 2014.
- [8] P. Liu, X. Qian, and H. Meng. Seeing Stars from Reviews by a Semantic-based Approach with MapReduce Implementation. Working Paper, Academy of Science and Engineering, USA, Jan. 2015.
- [9] J. McAuley, C. Targett, Q. Shi, and A. v. d. Hengel. Image-based Recommendations on Styles and Substitutes. *arXiv preprint arXiv:1506.04757*, 2015.
- [10] A. McCallum, K. Nigam, and others. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.
- [11] B. Pang and L. Lee. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL ’05*, pages 115–124, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

- [12] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.
- [13] K. Shimada and T. Endo. Seeing Several Stars: A Rating Inference Task for a Document Containing Several Evaluation Criteria. In T. Washio, E. Suzuki, K. M. Ting, and A. Inokuchi, editors, *Advances in Knowledge Discovery and Data Mining*, number 5012 in Lecture Notes in Computer Science, pages 1006–1014. Springer Berlin Heidelberg, 2008.
- [14] D. Tang, B. Qin, T. Liu, and Y. Yang. User modeling with neural network for review rating prediction. In *Proc. IJCAI*, 2015.
- [15] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.
- [16] Y. Yang and X. Liu. A Re-examination of Text Categorization Methods. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 42–49, New York, NY, USA, 1999. ACM.
- [17] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420, 1997.
- [18] Z. Zhai, B. Liu, H. Xu, and P. Jia. Clustering product features for opinion mining. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 347–354. ACM, 2011.
- [19] F. Zhu and X. Zhang. The influence of online consumer reviews on the demand for experience goods: The case of video games. *ICIS 2006 Proceedings*, page 25, 2006.
- [20] J. Zhu, H. Wang, and X. Zhang. Discrimination-Based Feature Selection for Multinomial Naïve Bayes Text Classification. In Y. Matsumoto, R. W. Sproat, K.-F. Wong, and M. Zhang, editors, *Computer Processing of Oriental Languages. Beyond the Orient: The Research Challenges Ahead*, number 4285 in Lecture Notes in Computer Science, pages 149–156. Springer Berlin Heidelberg, 2006.