# Sentiment Classification on Product Reviews

Ishan Kumar
School of Computer Science and Engineering
Lovely Professional University

Manish Prajapati
School of Computer Science and Engineering
Lovely Professional University

**Abstract:** ***The concept of product review concept, application problem, has recently become very popular in mining and language research. Here, we want to learn the consistency between Amazon product reviews and rating of products offered to consumers. Machine learning algorithms has been used including Naive Bayes analysis, Vector Support machines, the local K- method and deep neural networks such as Recurrent Neural Network (RNN). By comparing these results, we can gain a better understanding of these algorithms. They can also serve as add-ons to other forms of fraud detection.***

**Keywords: Sentiment, NLP.**

## I. Introduction

In recent years we have seen an increasing number of research efforts expanding our understanding of textual resources. As we can see from the statistics from the web of information, the papers published in neuroscience have been on the rise over the years. One of the findings of this study is called sentiment analysis or sentiment classification, that is, given a large amount of text, we can study people's ideas, appraisals, behaviours, and feelings in organizations, people, stories, events, topics, and their own attributes. The use of this method is different.

For example, businesses are always seeking to gain public or consumer opinions and feelings about their products and services. Potential customers also want to know the ideasfeelings of existing users before using the service or purchasing a product. Finally, researchers [2] use this information to conduct a comprehensive market study of consumer perceptions and ideas, which may lead to better forecasts of the stock market.

However, to say this, finding and viewing opinion sites on the Web and disseminating the information contained in them remains a daunting task due to the proliferation of various sites. Each site generally contains a large volume of text whose ideas are not readily available in detailed submissions and long blogs. The average reader will have difficulty identifying the relevant sites and summarizing the information and details in it [5]. Besides, teaching a computer to recognize hacking is a complex and challenging task given now, a computer cannot think like humans. The purpose of this paper is to compare and distinguish positive and negative customer reviews with different products and to create a supervised learning model and to eliminate a large number of reviews. Our data contains customer reviews and ratings, which we found in the consumer reviews of Amazon products. We extracted the features of our data and created a specific supervised model based on that. These cameras include not only traditional algorithms such as unlimited beaches, vector-based virtual machines, close K neighbours, but also deep learning metrics such as Recurrent Neural Networks and Deep RNN. We have compared the accuracy of these models and gained a better understanding of the attitudes recorded in the products.

## II. A Brief Literature Survey

To date, there have been many research papers correlated to product reviews, detailed analyses or data mining. For example, Xu Yun [8] el al of Stanford University used existing learning algorithms such as the perceptron algorithm, trivial reserves and vector support machine to predict the ratio of reviews to the Yelp rating dataset. They typically performed cross validation using 70% of the data used as training data and 30% of the data as test data. Here, the author used various classifiers to figure out accuracy and recall values.

In article [3], Maria Soledad Elli and Yi-Fan represented the emotions from the reviews and analysed the construct-up effect business model. Authors say the tool gives them excellent accuracy. They have used mainly Multinomial Naive Bayesian (MNB) and support vector machines as major attackers.

Callen Rain [6] suggested extending the present work in the area of natural language correction. Naive Bayesian and the list of decisions used to classify a given classification as positive or negative.

The neural networks are deeply learned and popular in the area of sentiment analysis. Ronan Colbert [1] et al used the available network of semantic role labelling task with the aim of avoiding large-scale task-related engineering. On the other hand, in paper

[7], the authors suggested using neural recursive networks to achieve a better understanding of the dynamics of tasks such as sentiment analysis.

In this paper, we want to use traditional algorithms including Naive Bayesian, K nearest neighbour, Supporting Vector Machine and deep learning algorithms. By comparing the accuracy of these models, we would like to gain a better understanding of how these algorithms work in tasks such as sensory analysis.

## III. Dataset and Features

### Data Pre-processing

Our dataset comes from Consumer Reviews of Amazon Products[1]. This dataset has 34660 data points in total. Each example includes the type, name of the product as well as the text review and the rating of the product. To better utilize the data, first we extract the rating and review column since these two are the essential part of this project. Then, we found that there are some data points which have no ratings when we went through the data. After eliminating those examples, we have 34627 data points in total.

Besides, to have a brief overview of the dataset, we have plot the distribution of the ratings. In Figure 1, it shows that we have 5 classes - rating 1 to 5 as well as the distribution among them. Also, these five classes are actually imbalanced as class 1 and class 2 have small amount of data while class 5 has more than 20000 reviews. Here is one sample from our dataset:

Here, we will illustrate how we convert a review text into an input vector, and we simply take the rate of a review as its label.

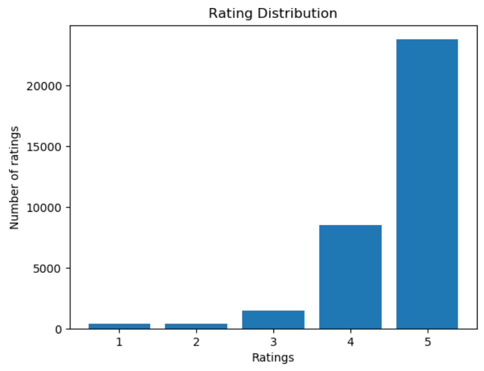

**Figure 1:** Rating Distribution

### Data Resampling

Due to the imbalance of our dataset, we have tried data resampling in some of our experiments. Data resampling is a popular way of dealing with imbalanced data. For this project, we tried to supplement the details of classes 1,2 and 3 by including those samples because the three classes have a smaller sample than the other two. Therefore, the initial revisions of labels 1,2 and 3 have been repeated 15 times in our training set. However, since there are so many repetitive samples in the training set, it is easy for the model to over fit.

### Features

We tried two types of symbols in the project. The first type is the traditional method. Basically, we create a dictionary based on common words and an index of each word. We limit the word dictionary to 6 occurrences and end up collecting 4223 words across our dataset. We then convert each update into a vote, where each value represents how many times the name reflects. In this case, we actually tried to change the length and length of the dictionary. What we found was that the increase in dictionary length did not have more effect than accuracy.

Another type of feature we used is the 50-d glove[2]dictionary which was pertained on Wikipedia. For this part, we basically want to take advantage of the meanings of each word. In this case, we represent each review by the mean vector of 50-d glove vectors of all individual words making up the review. As we will see in our result, because of the way we represent each review, the features got weakened and the distance between different reviews actually is not that accurate.

## IV. Methods

For classification problems,Naive Bayes is one of the bestfamiliar generative learning algorithms. This algorithm considers that $x^j$*is* conditionally independent given y, and it is known as Naive Bayes assumption.

$$p(x_1, ..., x_k|y) = \prod_{i=1}^{k} p(x_i|y)$$

We also incorporated Laplace Smoothing in our model to make it work better. The prediction of an example is given by the formula below:

$$\hat{y}^{(i)} = \arg\max_j \prod_{i=1}^{k} p(x_i|y=j)\phi(j)$$

K-nearest Neighbor(KNN) classification method which does not require parameters. It has been extensively used in recent times. When creating a prediction, this method first look for the $K = n$, which isclosest neighbors of the input. Then, it will pass on the most of that $n$ neighbor's class. The distance in between the neighbors is Euclidean distance, capable of measuring the similarity among each data point [4].

$$\hat{f}(x) = \frac{1}{K} \sum_{x \in N_K(x)} y_i$$

The equation above shows the mathematical representation of KNN algorithm. The general idea of KNN is that if the inputs are similar to each other, then the output would be the same. In this project, we have tuned the number of nearest neighbors *K* among 4,5 and 6.

Linear SVM is a method that separates the labeled datasets by creating a classifier (a vector).Given two different types of points, circles and x's, in space, it tries to extend a small distance from one point to another. In other words, expand the line.

Short-Term Memory (LSTM) is a unit of the Custom Neural Network (RNN). A typical unit of LSTM is a cell, an input gate, an output gate and a forgotten gate. The cell remembers values in times of conflict and three gates control information flow in and out of cell.LSTM networks are well suited for classifying, processing and making predictions based on time series data, since there can be unknown time limits between key events in the time series.

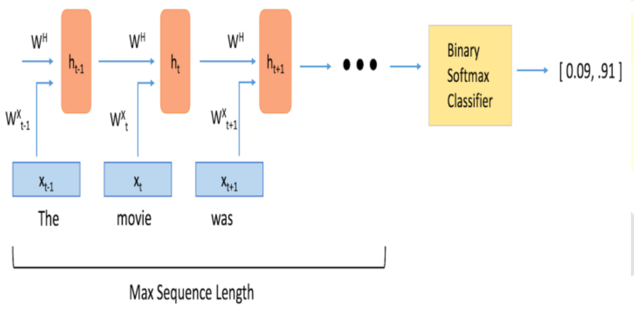

Figure 2: LSTM Configuration

For this method, we have also tried to input the original text with glove embedding. We found that there are 33629 reviews' length less than 100 words, which is about 97.1 percent of the whole dataset. Therefore, the max text length has been set to 100. Then, all the review text data has been padding to 100 word length and the words which are after 100 have been removed. Because when implementing this method, the input shape of the data should be the same. After that, each word is shown by glove word vector as a input of the neural network.

In this system, we used the LSTM with 128 hidden units and then used a solid net with soft-max as the activation function to predict these 5 classes. The data being trained for 20 epochs in experiments using LSTM. Adamoptimizer is used to optimize the arguments, the learningrate is0.01andthebatchsizeis32. Topreventoverfitting,a dropout rate of 0.2 was set in the LSTM layer.

## V. Result and Discussion

The entire dataset of 34,627 reviews was divided into a training set of size 21000 (60%), a validation set of size 6814 (20%) and a test set of size 6813 (20%).

With 4223-d input features representing review text, we implemented Multinomial Naive Bayes, SVM with Linear Kernel, SVM with RBF Kernel, KNN-4, 5, 6 and LSTM. KNN-5 outperforms the other 2 KNN models and SVM with Linear Kernel slightly outperforms SVM with RBF Kernel. The SVM with Linear Kernel seems to have overfitting problem indicated by the significant gap between training accuracy and test accuracy. LSTM performs best in term of test accuracy among all of them.

With 50-d input features from glove dictionary, we run Gaussian Naive Bayes, SVM with Linear Kernel and KNN- 4, 5 6 and LSTM. KNN-5 outperforms the other 2 KNN models again. Besides, we tried data resampling on LSTM model but unfortunately it did not improve the test accuracy due to overfitting problem. It turns out that LSTM generates best predictions among all models again.

Detailed results of training and test accuracy of all models are listed in Table 1.

Table 1: Performance of different models

| Models | Training Acc. | Test Acc. |
|---|---|---|
| Multinomial NB | 75.1% | 70.6% |
| Linear SVM | 83.4% | 69.6% |
| RBF SVM | 69.7% | 69.2% |
| KNN-4 | 61.7% | 61.7% |
| KNN-5 | 65.5% | 65.4% |
| KNN-6 | 64.9% | 64.6% |
| LSTM | 73.5% | 71.5% |
| Gaussian NB w/ Glove | 52.2% | 52.4% |
| Linear SVM w/ Glove | 68.7% | 68.6% |
| KNN-4 w/ Glove | 58.1% | 57.6% |
| KNN-5 w/ Glove | 62.6% | 62.2% |
| KNN-6 w/ Glove | 61.3% | 61.6% |
| LSTM w/ Glove | 70.1% | 70.2% |
| LSTM w/ Glove(Resample) | 85.6% | 65.6% |

In general, all models perform better with traditional in- put features than with glove input features. Specifically, LSTM generates the most accurate predictions over all other models.

We observed that KNN required much higher computation complexity than Naive Bayes and SVM during train time. As in KNN algorithm, it needs to calculate the distance of all the evaluation data points and all the training data points, which is more time consuming.

In addition, the increase of the dictionary's length did not have too much effect on the accuracy. One explanation is that when we decrease the threshold of the dictionary, the length of dictionary will increase. But the problem is that we only have less than 40,000 reviews. If we think about it, the number of data points is not that significantly larger than the dimension of feature space. So the curse of dimensionality could the issue here.

The result using glove mean is worse than the method of normal word count. The possible reason is that if we use the average, the individual word feature will be weakened, and then the distance between different reviews will be inaccurate.

When it comes to LSTM, the result is a little bit better than other conventional machine method due to the bigger amount of the parameters. We can also see from table 1, after resampling, the training accuracy of LSTM with Glove has reached 85.6 %.

## VI. Conclusion

In summary, we have tried two types of features. For this two type of features, we tried all the algorithms we mentioned in the model part including Naive Bayes, SVM, KNN, LSTM. From the results, we can see that our accuracy on the test set is the best when we use LSTM on the first type of feature. One of the main reasons our accuracy is not high enough is because of the data imbalance. We tried resampling and different weighting techniques that we got from the feedbacks of the audience during the poster session. But that didn't help too much. Another possible solution we haven't tried is to find more data points from other resources. We think that might help us solve the problem of data imbalance.

## References

[1] R. Collobert, J. Weston, L. Bottou, M. Karlen,K. Kavukcuoglu, and P. Kuksa. Natural language pro- cessing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[2] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM, 2003.

[3] M. S. Elli and Y.-F. Wang. Amazon reviews, business analytics with sentiment analysis.

[4] S. Hota and S. Pathak. Knn classifier based approach for multi-class sentiment analysis of twitter data. In *International Journal of Engineering Technology*, pages 1372–1375. SPC, 2018.

[5] B. Liu and L. Zhang. *A Survey of Opinion Mining and Sen- timent Analysis*, pages 415–463. Springer US, Boston, MA, 2012.

[6] C. Rain. Sentiment analysis in amazon reviews using proba-bilistic machine learning. *Swarthmore College*, 2013

[7] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing, pages 1631–1642, 2013.

[8] Y. Xu, X. Wu, and Q. Wang. Sentiment analysis of yelps ratings based on text reviews, 2015.