Evaluating The Power of Senitment Analysis and Opinion Mining On User Feedback

Ali Hussain, 20511790
Patrick Tardif, 20527590

05 December 2016

1. Abstract

With the introduction of the internet came an absolute wealth of information at our fingertips. People can shop online, download apps, play games and then share their experiences with the rest of us. One of the drawbacks of having so much information, however, is the difficulty of trying to account for every single person's opinion. Some mediums, such as the app store or Amazon, allow users to leave their feedback in the form of "stars" but this isn't always available nor convenient. We want to try and take advantage of natural language processing to overcome this limitation; we hope to use the power of sentiment analysis to accurately quantify a user's opinion of anything online and create a means to easily analyze feedback.

2. Introduction

Big data and natural language processing have been buzzwords within the tech industry recently and its with reason. The ability to analyze trends in users' thoughts over such a massive scale is such a powerful tool with an infinite amount of possibilities and applications. Sentiment analysis may be one topic of NLP but its importance is not overstated; it conveys a ton of the problems and makes up a huge chunk of the research in the field of NLP (Lui 2012, 10).

Throughout our project we experimented with a few different methods to try and extract sentiment from text. This involves comparing certain models, such as linear vs quadratic discriminatory analysis, and taking advantage of pre-existing technologies such as latent Dirichlet allocation as described by Blei et al (David Blei and Jordan 2003). Overall, we attempt to classify a piece of text by training a model to vectorize it and then proceeding to apply a simple linear or quadratic classifier.

To test our hypothesis, we took advantage of a publicly available data set of reviews originating from Amazon's food products (Lui (2015)). One of the hardships of the internet is the amount of slang and sarcasm that makes processing tough and inaccurate. However, product reviews are generally a lot more structured and literal (Lui 2012, 16) and using this dataset gave us a huge advantage in that regard.

3. Methods

Text filtering

The most important form of dimensionality reduction for textual analysis is text filtering. Filtering allows for the removal of words with little to no predictive power over the desired response but add to the dimensionality of the data set and cause problems with clustering. There were four filters used to reduce the dimensionality of the data set.

The first removed common words provided known as 'stop words'. This filter removed a short list of common words with very little impact on the meaning of the text, yet can create unwanted clusters when applying clustering techniques. These are words such as "I", "the", "are", and other similar words which can be found in NLTK's stopword package.

The second filter was to remove all tokens with a length of 2 or smaller. These were removed as all 2 letter English words provide very little meaning. The set of all 2 letter tokens is significantly larger than the set of all 2 letter words and tokens which aren't words provide no meaning yet can cause problems with clustering techniques and over-fitting.

The third filter is targeted at removing infrequent words. This filter was set to remove all words which appear in less than 30 documents. This number was chosen to be as high as the goal is to capture the sentiment of the review. The elements of the reviews which are specific to the item in question are not of interest as they don't help predict the sentiment of other products. For example, if we are interested in predicting the score for a bottle of orange juice, the word "coffee" is not important.

The final filter removes words which appear too frequently by filtering out words which appear in greater than a certain proportion of documents. In the context of foods reviews where we want to capture the sentiment of the review, this was set high at 0.8 as we didn't want to punish words which could help express the sentiment of the review which appeared commonly. Words such as "good" and "flavour" appear frequently and contribute significantly to the sentiment of the review.

Latent Dirichlet Allocation

Latent Dirichlet Allocation is a two-tiered clustering technique used typically used to group similar documents by clusters known as topics. The model follows the following generative process described by Blei et. al(David Blei and Jordan 2003):

"1. Choose N $\sim$ Poisson(ξ).

2. Choose $\theta \sim$ Dir(α).
3. For each of the N words wn:

(a) Choose a topic zn $\sim$ Multinomial(θ).

(b) Choose a word wn from p(wn |zn,β), a multinomial probability conditioned on the topic zn."

Essentially documents are assumed to be collections of topics drawn from a multinomial with a Dirichlet prior, and topics are assumed to be a collection of words drawn from a multinomial with a Dirichlet prior.

The implementation of Latent Dirichlet Allocation used is online-LDA implemented in genism. Online-LDA allows for the model to be updated in steps using online variational inference(Mathew Hoffman and Bach 2010). This implementation was chosen as it is fast and allows for the model to be updated as needed.

We make use of Latent Dirichlet allocation as a form of dimensionality reduction. We do this by, for each document, creating a feature vector where each element represents that topics weight. Therefore, each element is either 0 or in (0,1] (we make this distinction due to the sparsity of the vector) with all elements summing to 1 as these represent multinomial probabilities. This was found to be a reasonable approach by Blei et al where they found that when doing classification using linear classifiers this form of dimensionality reduction often improved performance.(David Blei and Jordan 2003)

Classifciation Through Discriminant Analysis

Given computational complexity of methods such as support vector machine and neural networks, and the resulting computation time on a single machine using the CPU we chose to consider simpler classifiers. We considered using discriminant analysis as a classifier. Discriminant analysis is the process of using methods to discriminate between different groups of data. One of these methods is linear discriminant analysis (LDA) where the discrimination is done by finding linear combinations of features to distinguish the classes. As noted in Linear Discriminant Analysis - Bit by Bit, "the goal of an LDA is to project a feature space (a dataset n-dimensional samples) onto a smaller subspace k while maintaining the class-discriminatory information" (Raschka (2014)). However, LDA requires the assumption of equal variances between the different classes. To overcome this, a modification of LDA is used called quadratic discriminant analysis (QDA) (ttnphns 2013) where the assumption is no longer necessary.

Model Validation

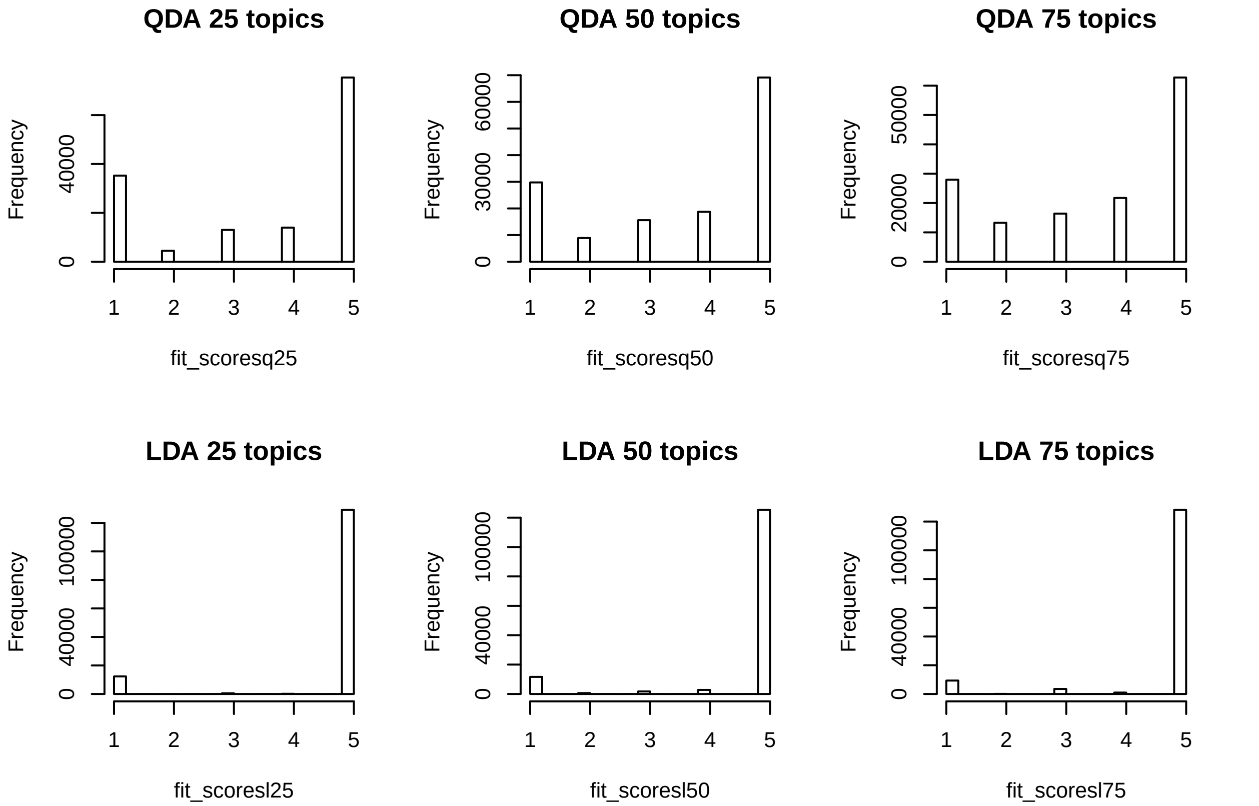
To estimate the accuracy of the model, the data set was split into two subsets by generating two random samples without replacement. One set was to be used to train the model and was approximately 75% of the original data set, while the remaining 25% was used to validate the model as a test set.

4. Results

Model Accuracy

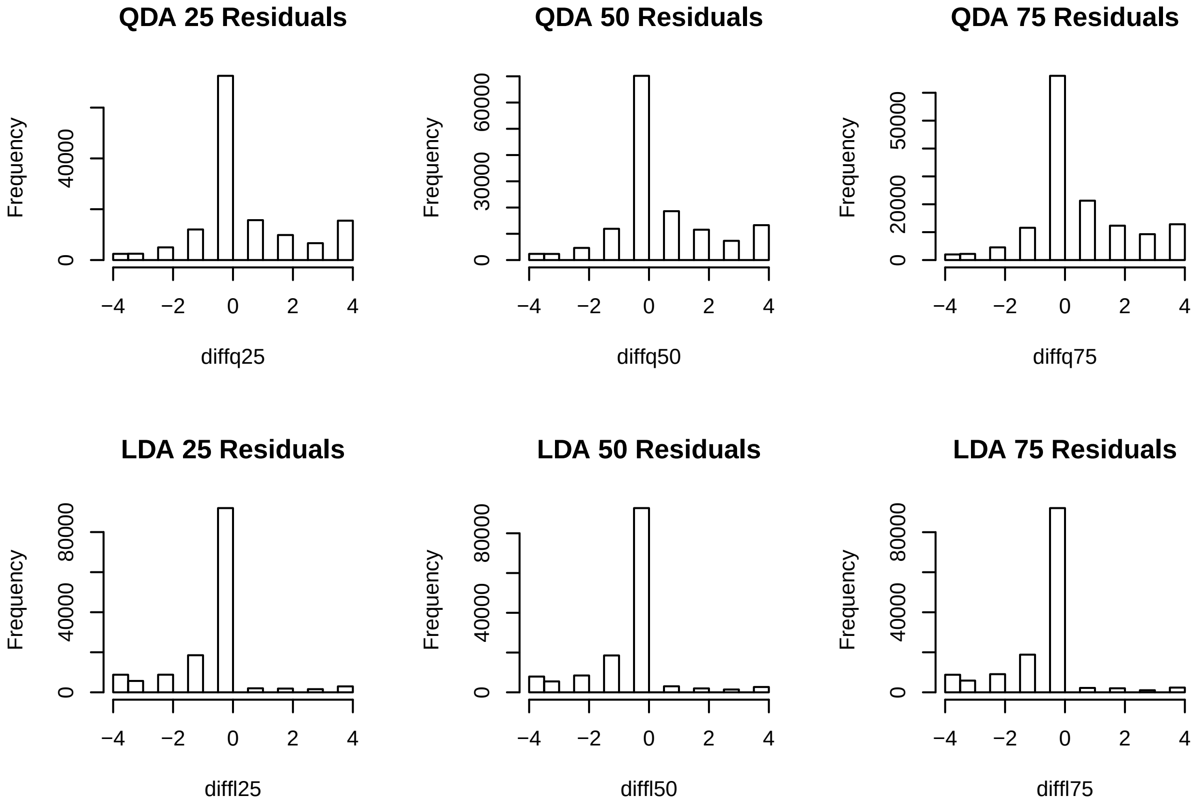
After filtering the dictionary, we had 16811 unique tokens down from the initial 211311. Using this dictionary, we trained three different Latent Diricihlet allocation models. We trained one with 25, 50 and 75 as the choice of topic number. This topic number being the number of dimension that our end feature vector will have. The actual topics can be found in the appendix.

We then trained 6 different classifiers; 3 linear discriminant analysis and 3 quadratic discriminant analysis. The following histograms show the distribution of each of the models.



From these we can see that QDA has a greater distinction between classes. Classes 1, 2, 3, 4 appear much more frequently with QDA than with LDA. With LDA class 5 appears much too frequently but this results in high accuracy because of how frequent 5 is relative to the other numbers.

The following histograms represent the residuals. We consdier these and not the absolute residuals as given the context the sign of the residual provides a significant amount of information and the residuals for LDA are not



symmetric.

From these residuals we can see that QDA has fairly symmetric residuals. The class is too large as often as it is too small. For LDA we see that the residuals are often negative because our estimate is often too large due to how often our estimate is 5. In both instances we can see we aren't frequently that far away from the correct class as the frequency of the residuals gets smaller the the larger the residuals are.

QDA.25	LDA.25	QDA.50	LDA.50	QDA.75	LDA.75
0.51	0.647	0.494	0.652	0.465	0.648

This table shows the success rates for each model. From this we can see that LDA has a higher success rate than QDA even though QDA has a better distribution. This is because guessing 5 each time has good performance in and of itself. If you guessed just 5 each time you would have approximately 0.63 as a success rate and this is essentially what LDA is approaching.

Model Choice

The following table outlines the distributions of the predicted class versus the actual class. From this table, we see that QDA gives a better estimate in terms of class proportions and that LDA gets close to predicting 5 for nearly every case.

	1	2	3	4	5
QDA.25	0.25	0.03	0.09	0.10	0.53
LDA.25	0.09	0.00	0.00	0.00	0.91
QDA.50	0.21	0.06	0.11	0.13	0.49
LDA.50	0.08	0.00	0.01	0.02	0.88

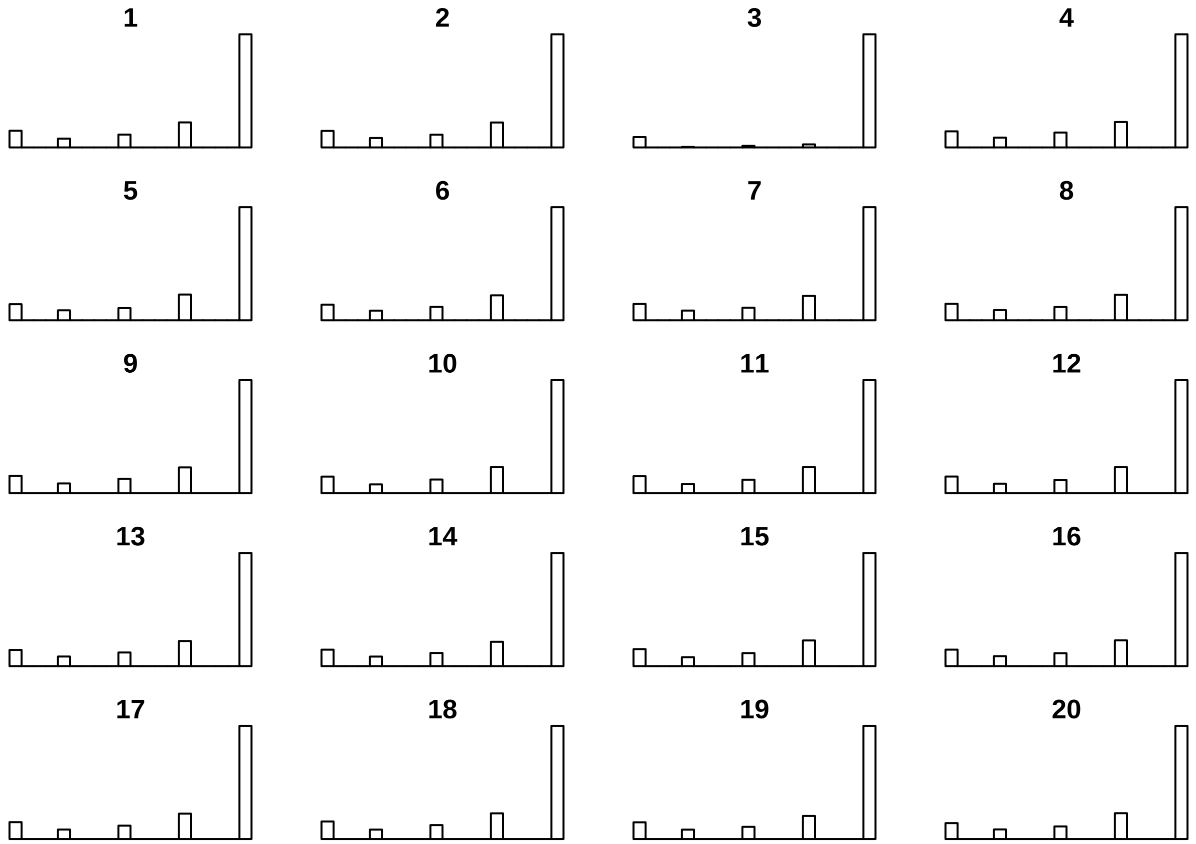
	1	2	3	4	5
QDA.75	0.20	0.09	0.12	0.15	0.44
LDA.75	0.07	0.00	0.02	0.01	0.90
Data	0.09	0.05	0.07	0.14	0.64

Therefore despite the fact that LDA has higher performance, it might make more sense to use QDA because it follows a distribution which is closer to the actual distribution of the data.

In order to verify this, we conducted a lineup test where we randomly plot the predictions against random samples from our data to see if the predicted data can be chosen our of the samples of real data.

Lineup for LDA50:

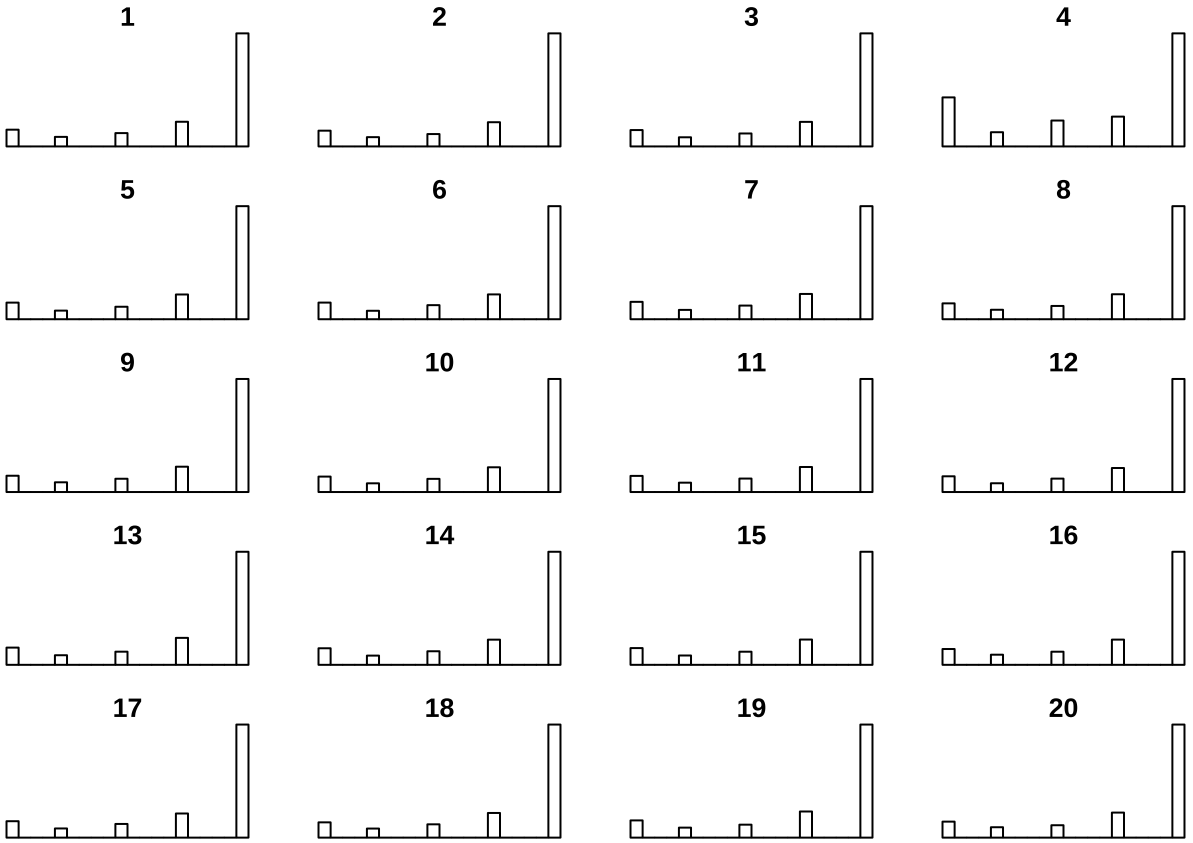
```r
lineup(test50, fitlda50)
```



```
## $trueLoc
## [1] "log(2.95147905179353e+88, base=20) - 65"
```

Lineup for QDA50:

```r
lineup(test50, fitqda50)
```



```
## $trueLoc
## [1] "log(1.92592994438724e+78, base=25) - 52"
```

From this we can see that both QDA and LDA can be picked out of the actual data with relative ease though LDA is much more obvious. LDA and QDA being suspects 5 and 7 respectively. QDA does seem to be follow a distribution reasonably appropriate.

5. Discussion

In conclusion, we've managed to form two models (QDA50 and LDA50) with 49.4% and 65.2% accuracy respectively. Even though the accuracy rates aren't as high as we've hoped, looking how our model fits with the real data gives us confidence that we're on the right track. Improvements can be made in how we're training our data. We decided to use QDA/LDA because of the computational cost that comes with a 500,000 element dataset. Using support vector machines (SVM) or majorization-minimization (MM) algorithms could've yielded better results but would've been too expensive. Some alternatives would've included training on a smaller dataset to get a model within the time limitation or changing how we treated certain words and punished others to improve LDA. Additionally the feature space could be improved by optimizing the number of topics and filtering to better capture the sentiment of the reviews. Also, as noted in the introduction, the dataset we're using uses very structured and literal English. Attempting to run this model on most internet postings would most likely result in very inaccurate results and would require a lot of fine tuning.

References

David Blei, Andrew Ng, and Michael Jordan. 2003. “Journal of Machine Learning Research 2003, Latent Dirichlet Allocation.” https://www.cs.princeton.edu/~blei/papers/BleiNgJordan2003.pdf.

Lui, Bing. 2012. “Sentiment Analysis and Opinion Mining.” https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf.

———. 2015. “Amazon Fine Food Reviews.” https://www.kaggle.com/snap/amazon-fine-food-reviews.

Mathew Hoffman, David Blei, and Francis Bach. 2010. “Online Learning for Latent Dirichlet Allocation.” https://www.cs.princeton.edu/~blei/papers/HoffmanBleiBach2010b.pdf.

Raschka, Sebastian. 2014. “Linear Discriminant Analysis - Bit by Bit.” http://sebastianraschka.com/Articles/2014_python_lda.html.

ttnphns. 2013. “StackOverflow: Three Versions of Discriminant Analysis: Differences and How to Use Them.” http://stats.stackexchange.com/a/71571.