

# Exploiting Ordinality in Predicting Star Reviews



Alim Virani

UBC - Computer Science  
alim.virani@gmail.com

Chris Cameron

UBC - Computer Science  
cchris13@cs.ubc.ca

## Abstract

Automatically evaluating the sentiment of reviews is becoming increasingly important due to internet growth and increasing customer and business use. We hope to address the question of what is the best model for classifying a review's text to its labels. We propose using a classifier that combines metric labelling and ordinal regression.  results showed that metric labeling was not improved by combining it with ordinal regression. Moreover, our results indicate that a one-vs-all classification approach may be  best way to classify reviews.

## 1 Introduction

Platforms for online reviewing are widespread. Reviews inform potential customers about product quality and help businesses to manage reputation and customer satisfaction. Due to the large sizes of review corpora, there is interest in developing automatic ways to extract and aggregate review information. In particular, businesses are interested in automatically aggregating positive and negative sentiment across their corpora of reviews. To facilitate this, often review systems allow customers to submit a rating- often on a 5-star scale- along with a text review.

We address the classification problem of inferring star ratings from review text. This task is useful because an effective classifier could provide rating suggestions for new reviews being authored and label archived text-only review systems. Our goal is develop a predictive model that improves upon the state-of-the-art.

It's important to define the metric that we are using to judge the performance of a classifier. For review classification, error is typically measured

using rank loss as below and does not use 0-1 loss.

$$Error = \frac{1}{N} \sum_i^N |\hat{y}_i - y_i|$$

Intuitively, this means misclassifying a 1 star as a 5 star should be treated as much worse than misclassifying it as a 2 star.

One effective model for star-rating classification (in terms of rank loss) is *metric labelling*, developed by (Pang and Lee, 2005). *Metric labelling* is tailored to rating classification as it exploits label ordering and captures the scale of the labels. The authors are essentially penalizing misclassification according to some measure of distance between labels. This measure of distance was also tailored to the classification problem because it measured distance in terms of polarity (positive vs negative) of sentiment.

*Metric Labelling* captures the distance between labels by selecting the label for a point that minimize its absolute difference in rating over its  $k$  nearest neighbours weighted by a polarity similarity measure. The label  $\ell_x$  of test point  $x$  is found by the minimization of equation (1) below.

$$-\pi(x, \ell_x) + \alpha \sum_{y \in \mathcal{N}_k} d(\ell_x, \ell_y) sim(x, y) \quad (1)$$

where:

$d$  is treated as  $|\ell_x - \ell_y|$

$\pi(x, \ell_x)$  is a prior preference function that is fit with a one vs. all classifier

$sim$  is a polarity similarity metric

This model showed promising results which makes sense since it was tailored specifically towards this classification task. However, we saw some limitations with using a one vs. all classifier as a prior. A one vs. all classifier (e.g. multinomial logistics regression) is naive since it assumes independence between classes. However, we saw

these labels as very much dependent: A 5-star review is inherently more similar to a 4-star review than a 1-star review. Also, these classifiers would treat misclassifying a 1-star review as 2,3,4 or 5 as equally bad. We propose improving *metric labelling* by replacing its prior with a more appropriate model. We should note that other authors, such as (Gupta et al., 2010) and (Ghazvinian, 2010) have had success using one-vs-all models (see related works).

Regression doesn't have the problem of assuming label independence since it treats labels as points on the real number line. However, regression has numerous problems. Regression (especially highly linear models) would be incapable of capturing a non-uniform ordinal scale between rating labels. Consider: Is a 1-star restaurant really twice as "good" as 2 star restaurants? While regression can capture the fact that a 5-star is more like a 4-star than a 1-star (i.e. the order) it further assumes that 1-star is as similar to a 2 star as a 4-star is to a 5-star (i.e. the scale). That is, it correctly captures the ordinal nature of the scale but it goes further and assumes a cardinal scale which may not be warranted.

Also, regression projects onto the real numbers and penalizes predictions outside of the rating scale in training. Classifying a 1-star as a 5-star intuitively seems much worse than classifying a 5-star as a 10. Yet, a regression model would not distinguish between these errors.

The general problem here is that our star rating system isn't actually a discretization of the real number line. Rather, our labels are simply classes, which have an ordinal ranking.

### 1.1 Approach to model ordinal labels

We propose to replace the prior model in *metric labelling* with *ordinal regression*. We hypothesize that ordinal regression improves over both regression and one-vs-all classification as it has the best of both worlds; it models discrete labels and captures order between labels. Ordinal regression is a variant of multinomial logistic regression where the ordinal nature of the dependent variable is explicitly modeled.

This is done by modeling the probabilities that the star rating  $Y$  of a covariate  $x$  belongs to class  $\ell_i$  or any class ranked less than  $i$ . This differs from standard logistic regression that models  $P(Y = \ell_i | X)$  directly.

These probabilities are typically learned using the logit function  $\Theta$ . Where:

$$P(Y \leq \ell_i | x) = \Theta(\alpha_i - \beta^T x)$$

This is very similar to standard logistic regression which uses:

$$P(Y = 1 | x) = \Theta(\beta^T x) \quad (2)$$

However, ordinal regression trains over both  $\beta$  and every  $\alpha_i$  such that  $\alpha_{i-1} \leq \alpha_i \leq \alpha_{i+1}$ . Intuitively,  $\alpha_i$  can be seen as the learned cardinal distance between the ordered labels.

A predicted label  $\hat{Y}$  is classified by:

$$\hat{Y} = \max_j \Theta(\alpha_j - \beta^T x_i) - \Theta(\alpha_{j-1} - \beta^T x_i)$$

(Baccianella et al., 2009) apply a similar ordinal regression model to the one described above to classify ratings of product reviews. The authors use an SVM classifier rather than a logistic classifier in their loss function.

Both (Pang and Lee, 2005) and (Ghazvinian, 2010) were interested in investigating *Ordinal Regression* for review classification in future research. However, (Gupta et al., 2010) claimed that they found that ordinal regression performed poorly.

### 1.2 Related Works

Gupta et al. (2010) approach this problem using regression, one vs. all classification, and ordinal regression. They use a perceptron model to implement *thresholds* in ordinal regression, a neural network for regression, and Logistic Regression for one vs. all classification. They classified overall ratings for restaurant reviews as well as other restaurant aspect ratings. Surprisingly, they found that Logistic Regression performed the best, followed by numeric regression and finally by ordinal regression. They stressed the simplicity of their ordinal regression model and proposed to use more complicated models in future work.

Ghazvinian (2010) approaches the problem by using Logistic Regression. The paper mostly focusses on feature understanding and offering evidence for the importance of certain features. They achieved good performance with their classifier doing as well as humans in terms of precision. The authors cite their machine learning technique as a limiting factor and propose adding metric labeling or some other approach that exploits distance between labels.

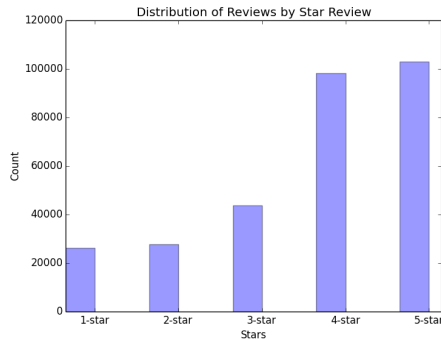


Figure 1: Distribution of reviews by star review.

## 2 Methodology

### 2.1 Approach

We want to evaluate if using an ordinal regression classifier, as a prior for metric labeling will improve over the previous metric labeling approach. To do this we first compared standard ordinal regression and standard metric labelling against ordinal regression combined with metric labelling on the same data set. We also wanted to benchmark the performance of our hybrid model by comparing against some baseline predictive models. The models we tested included logistic regression, linear regression, and random forest regression.

Except for ordinal regression, which we partially implemented ourselves, we used the python sklearn library.

### 2.2 Corpus

We tested the various models on the Yelp Data set. This data set had over 300 thousand establishment reviews labelled with a user star rating from 1 to 5.

Since we are trying to train a classifier that maps lexical features to star labels, we removed reviews where we think we can not extract sufficient informative features to identify sentiment. We chose to remove some 36 thousand reviews under a threshold of 160 characters. This filtering method was supported in (Ghazvinian, [2011](#)), where they filtered reviews less than 100 characters.

As shown in Figure 2.2, there was a large class label asymmetry in the data set. To simplify analysis, we downsampled our data to 20,000 reviews per star label.

It may have been important to run some tests with human annotators to justify the scale of the

reviews. Due to time constraints and resource limitations we were not able to do this.

### 2.3 Feature Engineering

Our feature selection was guided by the relevant literature in review classification. Unfortunately, we have left certain features out due to time constraints and implementation difficulty. However, note that our approach measures relative performance of models and we have no reason to believe that more comprehensive features would substantially improve one model more than another.

Feature engineering was predominately done with the NLTK and SKlearn libraries.

#### 2.3.1 Metric Labelling

In their metric labelling approach, Pang and Lee (2005) use a single feature to capture review sentiment. They build a positive sentence classifier based on large annotated corpus and measure positive sentence percentage (PSP). PSP is defined as the number of positive classified sentences over subjective sentences. We didn't have access to a positive sentence corpus, so we decided to develop our own method to calculate PSP. We also did not do any objective sentence removal that was mentioned in the literature.

Based on the literature, using individual word polarity is beneficial towards sentiment analysis (Ghazvinian, [2011](#)). However it can be important to recognize valence shifter to avoid classifying phrase like *not good* as positive, which we did not do. We decided to create a PSP metric by defining a positive sentence by number of positive and negative polarity words weighted by their strength. A sentence is positive if :

$$\left( \sum_{i \in \text{words}} \text{polarity}_i \times \text{strength}_i \right) > 0$$

where:

$\text{polarity}_i = 1$  if positive and  $-1$  if negative  
 $\text{strength}_i = 2$  if Strong and  $0.5$  if Weak

We determine word polarity from the Harvard General Inquirer dataset, which is widely used in NLP. The dataset contains  $\approx 2000$  polarity words and identifies them as positive/negative and strong/weak.

To justify this metric as being an informative feature, we created boxplots of the aggregate positive/negative score for a review against its star rating as shown in Figure 2. There is a lot of variability in the data but there is clear correlation be-

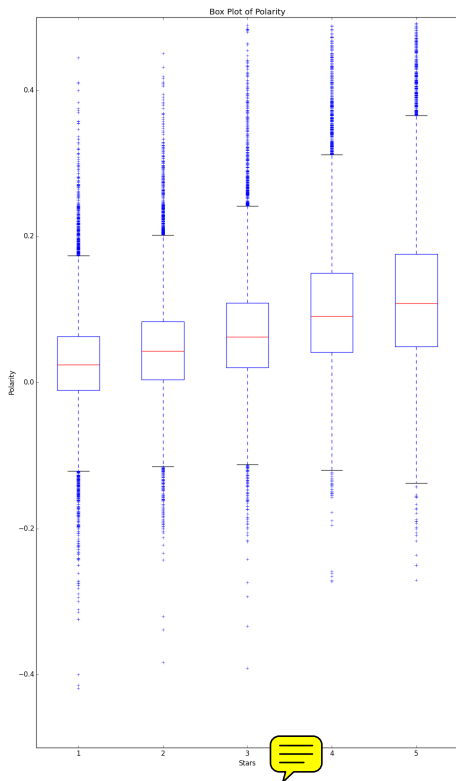


Figure 2: Boxplots of aggregate review polarity against star rating

tween increasing star rating and increasing aggregate positive polarity.

Many of the words in the polarity dictionary have more than one sense. To disambiguate word sense, we used the most commonly used sense.

### 2.3.2 Bag of Words

Based on the literature, we decided to combine a bag-of-words model with various other features based on word polarity. We tested models with bag of words alone, polarity alone, and the two in combination.

We built the bag-of-words features using standard techniques. After tokenization, we applied POS tags and used a Wordnet based lemmatization. We then stemmed words using the Snowball stemmer. We experimented with both td-idf and regular counts and settled on the former. We also tested the inclusion of exclamation marks. The size of our final feature vector was 100,000.

Our initial feature vector performed poorly across all models because our feature matrix was

overly sparse. To mitigate this we only retained ngrams that appeared with a high enough frequency. This technique is similar to the approach suggested by (Gupta et al., 2010). We also improved performance by removing common english stop words and corpus stop words, which are words that appear with overly high frequency in the corpus.

Our polarity features included number of words, number of sentences and features that represented the intensity and polarity of words and sentences in the text (using the Harvard General Inquirer dataset).

## 2.4 Approached to Training Models

### 2.5 Baselines

We used L2 regularization for all three linear models. We used cross validation to set the lambda parameters for logistic and linear regression. Note that this was 2 step cross validation: The cross validation for parameter selection was done using the 9 folds of any given iteration.

### 2.6 Metric Labelling

Implemented (Yang and Lee, 2005).  $\alpha$  had to be found through cross-validation. KNN is extremely slow in high dimensions  $O(kMND)$  where  $M$  is size of the test set,  $N$  is size of training set, and  $D$  is dimensionality of the feature space. In 1 dimension this would become  $O(kMN)$  but this is still extremely slow given we might have  $N=90,000$  and  $M=10,000$  for a given fold. However, in 1 dimension we can first sort the training data, create a binary tree over it, and find the  $k$  nearest neighbours of any test point in  $O(k \log N)$ . Thus, with this change the algorithm runs in  $O(kM \log n)$ . With this change, metric labelling performs as fast as the others. Note that we fixed  $k$  to 5. We found through experimentation that this value worked well. That being said, a major improvement to our project would be to set  $k$  using cross-validation.

## 3 Evaluation

### 3.1 Evaluation Procedure

#### 3.1.1 Performance Metric

We wanted to compare results from both real values and nominal labels. As such, we need to choose an evaluation metric that could work for both and capture what we consider good performance.

Since these star ratings are labels, we decided to

round any real valued results to their nearest integer. We also projected results to a 1 to 5 scale.

$$\text{Stars} = \max(1, \min(5, \text{round}(\text{Prediction})))$$

However, we also wanted to capture the idea that classifying a 1-star as a 5-star is significantly worse than classifying it as a 2-star. As such, we used rank loss to measure performance.

### 3.1.2 Cross Validation

To measure the generalized error of each model, we used 10-fold cross validation.

## 3.2 Evaluation Results

Our results as seen in Table 1 show that metric labeling was not improved by replacing its one-vs-all classifier with an ordinal classifier. Surprisingly, ordinal regression and ordinal regression combined with metric labelling did worse than both logistic and linear regression. Furthermore, our one-vs-all classifier (logistic regression) did the best overall with a mean average error of 0.489, beating both the (real-valued) regression models and our ordinal regression model by a wide margin. Referring to Table 3, we can see that logistic regression had a better f1-score than our hybrid approach for every label.

Except for logistic regression, the addition of polarity features to the bag of words model reduced mean average error. Looking at Table 3, we can see that predicted scores are concentrated around 2.5 for our hybrid model. We can surmise, this is directly due to using scale in our loss function. If our classifier cannot find a signal, we would expect it to choose 3 stars since that would have the lowest expected loss. That table also shows that logistic regression performs best on 1-star and 5-star reviews.

Note that we did not train Random Forests Regression with a bag of words since it is generally not recommended to do so.

## 4 Analysis

Our results are completely counter to our expectations. Given the apparent ordinal nature of the data, we expected the use of an ordinal classifier to have a clear advantage over the use of a one-vs-all classifier.

One major weakness in our project is that we haven't formally investigated this counter-intuitive result. A good future direction for this project

Model	Features		
	BOW	BOW+Polarity	Polarity
Random Forest	N/A	N/A	1.016
Ridge Regression	0.910	0.815	0.988
Ordinal Logistic	1.058	0.976	1.076
Ordinal + Metric	1.058	0.975	1.070
Logistic	0.489	0.559	1.145

Table 1: Mean Average Error for different Machine Learning models and different feature sets.

		Predicted				
		1	2	3	4	5
True	1	1249	18468	241	0	0
	2	209	18298	1401	6	0
	3	28	14194	5726	96	0
	4	6	7538	11886	616	0
	5	2	5741	13011	1268	0

		Predicted				
		1	2	3	4	5
True	1	15693	3476	521	132	254
	2	5315	10170	1400	7	315
	3	4090	3914	10389	3452	993
	4	484	689	3768	9546	5551
	5	514	252	689	4326	14239

Table 2: Top: Confusion matrix for Metric Labelling; Bottom: Confusion Matrix for Logistic Regression

Stars	Error Measure		
	Recall	Precision	F-score
1	0.06	0.84	0.12
2	0.92	0.28	0.43
3	0.29	0.18	0.22
4	0.03	0.31	0.06
5	0	0	0

Stars	Error Measure		
	Recall	Precision	F-score
1	0.78	0.60	0.68
2	0.59	0.55	0.57
3	0.45	0.62	0.52
4	0.48	0.55	0.51
5	0.71	0.67	0.69

Table 3: Top: Error Measures for Metric Labeling; Bottom: Error Measures for Logistic Regression

Method Name	Best Result (MAE)
Logistic (Us)	0.489
Logisitc (Gupta et al., 2010)	0.637
Logisitc (Ghazvinian, 2010)	0.477

Table 4: Mean average error comparison for best models in literature.

would be to formally analyze what about this data set (and other’s like it) allow it to work well with a one-vs-all classifier.

That being said, we have some preliminary ideas to explain this result. The texts of different labels may be correlated but that correlation may not follow the rating ordering. While we can find a correlation between a label and the sentiment expressed in its text, we may not be able to extrapolate correlations for other labels through the ordering. In other words, the labels may not differ in degree (cardinal or ordinal), but in kind. Consider that dependencies could go counter to order: Is the text of a 2-star label really more similar to the text of 3-star label in every relevant respect as it is a 4-star label.

After a qualitative analysis of the data we’ve found the following general trend:

2-star and 4-star reviewers tended to express mixed sentiment. These reviewers often focused on justifying why they didn’t give the extreme value and as such spent almost an equal amount of time focusing on mitigating (abasing and redeeming respectively) factors.

We found that a 3-star review often actively expressed neutral sentiment and focused on what made the establishment unremarkable.

Finally we found that 1-star and 5-star reviews tended to be the longest and most passionate. The texts of both were similar in that they used copious amounts of superlatives.

Given the above trend, modeling a signal (predominately sentiment) from the text of these reviews as something that is ordered between star values may have been the wrong approach. The one-vs-all approach may not have been at a disadvantage as we thought there are dependencies that also go counter to the ordinal scale. For example, it is not clear that the texts of 4-star reviews are more similar to 5-star reviews as 2-star reviews.

Thus our contribution is a useful negative result along with supporting other authors’ results. For one, as shown in Table 1, logistic regression

works well. Secondly, we corroborated (Gupta et al., 2010) finding that ordinal regression classifiers do not work well.

#### 4.1 Future Investigation

To support our explanation for the success of logistic regression ~~regression~~ over ordinal regression, we want to experimentally investigate dependencies beyond ordering. Specifically, it would be beneficial to future research to identify concrete similarities between labels that go counter to order. For example, one simple question would be how would our performance change if we combined one and five star reviews. This change in performance would indicate how and if there are strong correlations between the labels’ texts.

### 5 Conclusions

We wanted to address the question of what is the best review classifier. Our approach was to improve metric labeling by replacing its one one-vs-all classifier with ordinal regression. We hypothesized that combining metric labeling with ordinal regression would improve metric labeling and possibly yield state-of-the-art results. We tested this hybrid method against several other models. We found that ordinal regression did not improve metric labeling and in general performed poorly. Moreover, we found- counter to our expectations- that logistic regression, which is a one vs all classifier performed the best by a wide margin. Logistic regression achieved a mean average error of 0.489 that was comparable to the best results found in the literature. Therefore, our project has a useful negative result and confirmation of the results of other authors.

### References

- Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2009. Multi-facet rating of product reviews. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval, ECIR ’09*, pages 461–472, Berlin, Heidelberg. Springer-Verlag.
- Amir Ghazvinian. Star quality: Sentiment categorization of restaurant reviews.
- Narendra Gupta, Giuseppe Di Fabrizio, and Patrick Haffner. 2010. Capturing the stars: predicting ratings for service and product reviews. In *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*, pages 36–43. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. 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. Association for Computational Linguistics.