Determing Trustworthiness in E-Commerce Customer Reviews

Dhruv Gupta
Department of Mathematics
and Computing
Indian Institute of Technology Patna
Patna, India
dhruv.mc12@iitp.ac.in

Asif Ekbal
Department of Computer Science
and Engineering
Indian Institute of Technology Patna
Patna, India
asif@iitp.ac.in

Abstract

In this paper, we delve into opinion mining and sentiment analysis of customer reviews posted on online e–Commerce portals such as *Amazon.com*. Specifically, we look at novel ways of automatic labelling of data for customer reviews by looking at the number of helpful votes and subsequently determine hidden factors that can explain why a customer review is more helpful or trustworthy in contrast to others. We further utilize the factors identified by Multiple Factor Analysis to training Logistic Regression and Support Vector Machine (SVM) models for classifying reviews into trustworthy and non-trustworthy. Experiments show the effectiveness of our proposed approach.

1 Introduction

As the e–Commerce business grows, more and more customer express their experience with the products bought online. There is no doubt about the ever growing number of customer reviews and the fact that the reviews can influence the potential buyer's decision. It has become exceedingly pertinent to develop algorithms that allow the e–Commerce industry to predict the potential trustworthiness of the customer reviews. In this direction, we attempt to utilize the concept of **Sentiment Analysis** to determine whether a customer review can be trusted or not.

Customer reviews on a online portal or social media platform such as Facebook[1], Twitter[2], Pinterest [3] etc. serve as advice for potential customers of service or products on whether the particular service suits their needs. Some statistics posed by Pang et al. (Pang and Lee, 2007) puts the scope of importance of these customer reviews.

Following are some insights as quoted from (Pang and Lee, 2007):

- 81% of Internet users consult the web at least once before buying a product (Pang and Lee, 2007).
- 73% to 87% Internet users confirm that customer reviews about services relating to medical, travel etc have influenced their decision on availing the service (Pang and Lee, 2007).
- For a 5–star rated product customers are willing to shell out 20% to 99% more as compared to a 4–star rated product (Pang and Lee, 2007). The variability arises from the variety of services and products on offer (Pang and Lee, 2007)

As is evident from the above insights, we can conclude that customer reviews pay a pivotal role in a "strategic" customers decision to purchase high value products or services. However as the number of users entering the sphere of Web 2.0 grows, we see a surge in the amount of reviews that users are able to write across social media platforms'(Pang and Lee, 2007). In such a case, a customer can only look at a few customer reviews before s/he makes a final decision to purchase a product. Thus, there arises a need to automatize the process of identifying the sentiment, opinion and the subjectivity hidden amongst these reviews. So, that a crisp report on how a product or service fares amongst the online blogosphere and social media can be reported to the Internet user, allowing her to make informed decision on whether to purchase a particular service or product.

Due to growing number of online reviews for any product or service, a lot of attention has

[1] www.facebook.com
[2] www.twitter.com
[3] www.pinterest.com

D S Sharma, R Sangal and J D Pawar. Proc. of the 11th Intl. Conference on Natural Language Processing, pages 196–205,
Goa, India. December 2014. ©2014 NLP Association of India (NLPAI)

recently been focused on classifying the customer reviews based on their subjectivity of opinions (Jindal and Liu, 2008). Jindal et al. (Jindal and Liu, 2008) point out that little attention has been paid to classifying whether a customer review can be trusted or not. The motivation the authors point out is that fake reviews may be created to increase the popularity of a particular product in the market. This can create a illusion in the mind of the potential customer that a product s/he is about to purchase is worth amount to be paid. Hence, it becomes extremely important to check such kind of non–trustworthy reviews.

Our contribution to the detection of opinion spam analysis is two-fold. First, for labeling of the Amazon customer review dataset[4] (McAuley and Leskovec, 2013), we utilized the "helpfulness" votes as a measure to determine if a customer review was trustworthy or not. This method of labeling the corpora is akin to utilizing a crowd–sourcing platform to determine the trustworthiness of customer review. Unlike prior approaches such as (Lim et al., 2010), where the authors employ human evaluators for determining whether a customer review is trustworthy or not; we have taken a very simple yet intuitive approach to label a Web–scale web corpora.

Secondly, we hypothesize that by employing Multiple Factor Analysis (Abdi1 and Valentin, 2007) in generating principal components for the features identified for each customer review, we can enhance the performance of the machine learning algorithms. We hypothesize that this can be attributed to the fact that the data points are projected in a lower dimensional feature sub–space, where the data is scaled on the most pertinent dimensions.

2 Related Work

2.1 Detecting Opinion Spam

Jindal et al. (Jindal and Liu, 2008) utilize a logistic regression approach to identify Type 2 (reviews on brands only) and Type 3 (non-reviews) by utilizing a variety of features describing different aspects of the product, reviewer and the review itself. The authors' objective lies in training a supervised classifier that can tell whether a review posted is a spam or genuine. The authors also mention that the use of other supervised learning methods such as support vector machines (SVM) (Cortes and Vapnik, 1995), naïve Bayes classifier etc. do not qualify to logistic regression in terms of quality of results obtained.

Jindal et al. (Jindal et al., 2010) study a constrained problem of identifying set of rules to predict whether a review is an anomaly with respect to the others. They mine these rules in a similar fashion to Association Rule mining, wherein they chalk out a separate definition of support and confidence to take into unexpectedness.

Lim et al. (Lim et al., 2010) present approaches to deal with review spammer behavior. They construct a linear regression model(Sharma, 1995) that takes into account features such as rating spamming, review text spamming, single product group multiple high ratings, single product group multiple low ratings etc. Their approach involves supervised machine learning algorithms, hence to acquire the labels for their dataset they took help of human evaluators.

Ghose et al. (Ghose and Ipeirotis, 2007) also train linear regression models to find whether a review is helpful or not. Also they extend their regression model to study the effect of various features on sales rank of products featured on Amazon.com. In addition to the features used by prior approaches (Jindal and Liu, 2008; Lim et al., 2010), the authors (Ghose and Ipeirotis, 2007) take into account review *subjectivity & objectivity* for the aforementioned objectives.

In our work we used Multiple Factor Analysis as a pre-processing step, and have been able to take into account the hierarchy and grouping of feature space whilst training machine learning algorithms. This can assist us in training of machine learning algorithms as learner do not take into consideration the grouping / hierarchy of features identified. Also, utilizing MFA as a pre-processing step we are able to scale continuous as well as nominal attributes on the most pertinent principal components identified by MFA. Having the feature vectors scaled, aids in improving the performance of the supervised machine learning algorithm. As we can see when comparing the AUC values of the baseline model and our proposed model (MFA used as pre-processing of SVM), we see that our proposed approach does better job in identifying the trustworthy reviews.

[4]https://snap.stanford.edu/data/web-Amazon.html

2.2 Multiple Factor Analysis

Pages et al. (Escofier and Pagès, 1994) in their work outline the Multiple Factor Analysis method for data containing groups of variables. They also outline methods for its application to sensory data. The authors outline solutions to solve the problem of introducing multiple groups of variables at once as active variables. This allows for weighing the groups of variables in a balanced manner and subsequently provides better insights into the data at hand. Abdi et al. (Abdi et al., 2013) work out a mathematical treatment behind Multiple Factor Analysis. The authors also outline several new algorithmic extensions of MFA. Some of the extensions at length they discuss are Hierarchical Multiple Factor Analysis (HMFA), Dual Multiple Factor Analysis, Procrustes Multiple Factor Analysis, Multiple Factor Analysis for Qualitative Data, and Multiple Factor Analysis Barycentric Discriminant Analysis (MUFABADA). HMFA entails hierarchically applying the MFA normalization; this assists in analysis of data sets that contain a hierarchical structure of variables (Abdi et al., 2013). Dual MFA is utilized in the cases where data contains in which the observations are partitioned (Abdi et al., 2013). Procrustes MFA is an extension that is useful for data in which we have several Euclidean distance matrices capturing the same observations (Abdi et al., 2013). MFA for qualitative data is an extension for qualitative data in the same vein as MFA is an extension of PCA for multi–block data (Abdi et al., 2013). A similar extension of PCA for qualitative data is Multiple Correspondence Analysis (MCA) (Abdi et al., 2013). MUFABADA is an approach of utilizing MFA in "multiblock barycentric discriminant analysis framework" (Abdi et al., 2013). MUFABADA addresses the problem of putting the observations in groups which have described these observations in different tables (Abdi et al., 2013).

3 Technical Background

In this section we familiarize the reader with the two pivotal approaches used for determing the trustworthiness of customer reviews on e–Commerce websites. We take the help of detailed literature available in the area of Factor Analysis (Escofier and Pagès, 1994), (Abdi et al., 2013), Logistic Regression (Sharma, 1995) and Support Vector Machine (Cortes and Vapnik, 1995).

3.1 Sentiment Analysis

Sentiment conveys humans emotions or opinions in a given piece of text. Sentiment Analysis as pointed out by Pang et al. (Pang and Lee, 2007) and Turney (Turney, 2002) is an attempt to identify the *subjectivity* or *sentiment polarity* of given piece of text. This is done by leveraging Natural Language Processing (NLP) techniques and trying to model a computation algorithm to identify the same automatically for a given piece of input text (Pang and Lee, 2007; Turney, 2002). Pang et al. (Pang and Lee, 2007) points out that the term "Sentiment Analysis" is more popular amongst the NLP community. "Opinion Mining" also conveys aggregating the subjectivity associated with the item (product) features being discussed in the text (Pang and Lee, 2007). For example, in the example below, "amazing" associates a certain opinion about the product camera, and the sentiment "really great" conveys the subjectivity about the "zoom" feature of the camera. Pang et al. (Pang and Lee, 2007) also point out that, the term opinion mining is favored amongst the information retrieval community.

> "Today I bought this *amazing* **camera** ! The *zoom* of this camera is **really great**."

In this work our proposed algorithm for sentiment analysis is based on logistic regression (Sharma, 1995) and support vector machine (Cortes and Vapnik, 1995). We make use of the implementations as available in Rinker et al. (Rinker, 2013).

3.2 Multiple Factor Analysis

Multiple Factor Analysis is one of the principal component methods that takes into account groupings of variables or attributes when describing the observations (Escofier and Pagès, 1994). MFA is multi–step process whereby in the intial step, a Principal Component Analysis (Jolliffe, 2005) is performed on each group of attributes followed by normalization of the data with eigenvalues (first singular value) found by PCA. Next, the normalized dataset is combined into a unique matrix to be further evaluated by doing a global PCA. MFA is most suitable to be applied to set of observations when the attributes describing it vary in nature i.e. they can be nominal and/or continuous in nature (Escofier and Pagès, 1994). The key distinguishing aspects of MFA (Abdi et al., 2013) are its

ability (i). to consider groupings of different variabales describing the same set of observations, (ii). to determine "compromise factor scores" or "common factor scores" and (iii). to project the individuals (observations) onto these "compromise factor scores".

We shall adopt the notations used by Abdi et al. (Abdi et al., 2013) to illustrate the Multiple Factor Analysis algorithm. A matrix, $\mathbf{M}$, conveys, the dataset on which we wish to apply the MFA algorithm. The rows of $\mathbf{M_{(i,)}}$ conveys the vector of observations while the columns $\mathbf{M_{(,j)}}$ convey the value of a particular feature / attribute for all the observations. A single element of $\mathbf{M}$ is denoted by $\mathbf{M_{(i,j)}}$. Groupings of attributes in $\mathbf{M}$ are considered as *sub–matrices* $\mathbf{M_{[i]}}$. A congregation of sub–matrices is represented as $\mathbf{T} = [\mathbf{M_{[1]}}\mathbf{M_{[2]}} \dots \mathbf{M_{[i]}}]$. The matrices, $\mathbf{M^T}$ and $\mathbf{M^{-1}}$ denote the transpose and the inverse of $\mathbf{M}$ respectively. To obtain a column vector of the diagonal elements of matrix $\mathbf{M}$ we use the operator **diag**. The operator **diag** does the opposite in case it is applied to a vector i.e. it transforms the vector into a diagonal matrix. The identity matrix is denoted by $\mathbf{I}$. The vector $\mathbf{1}$ represents a vector of ones (Abdi et al., 2013). The dimension is specified by the index when referring to a vector of ones $\mathbf{1}$ (Abdi et al., 2013).

The procedure to carry out MFA can be decomposed into three steps (Abdi et al., 2013). The first steps entails performing a principal component analysis (PCA) of each individual grouping of attributes, and observing the eigenvalues (first singular value) obtained for each grouping of attributes (Abdi et al., 2013). The second step includes combining of all the groupings of attributes after they have been divided by their respective eigenvalues, and running a non–normalized PCA on this data set (Abdi et al., 2013). The final step involves projecting the groupings of attributes on the "common space" (Abdi et al., 2013).

The first step in Multiple Factor Analysis utilizes the Singular Value Decomposition of the data matrix to perform PCA (Abdi et al., 2013). Mathematically, it can be described as (Abdi et al., 2013)

$$\mathbf{X_{[k]}} = \mathbf{U_{[k]}}\mathbf{\Gamma_{[k]}}\mathbf{V_{[k]}^T} \tag{1}$$

with

$$\mathbf{U_{[k]}^T}\mathbf{U_{[k]}} = \mathbf{V_{[k]}^T}\mathbf{V_{[k]}} = \mathbf{I} \tag{2}$$

Now we need to obtain the first singular values for each grouping of the attributes. For each grouping of attributes we get the first singular values by taking the inverse of the square of the first diagonal element of $\mathbf{\Gamma_{[k]}}$ i.e. (Abdi et al., 2013)

$$\mathbf{diag}(\mathbf{\Gamma_{[k]}}) = [\gamma_{(1,k)}, \gamma_{(2,k)}, \dots \gamma_{(n,k)}] \tag{3}$$

$$\alpha_k = \frac{1}{\gamma^2_{(1,k)}} = \gamma^{-2}_{(1,k)} \tag{4}$$

All the singular values of each grouping of data are stored in a matrix $\mathbf{A}$ computed as follows (Abdi et al., 2013)

$$\mathbf{A} = \mathbf{diag}\Big[\alpha_1\mathbf{1_{[1]}^T}, \alpha_2\mathbf{1_2^T}, \dots, \alpha_K\mathbf{1_{[K]}^T}\Big] \tag{5}$$

Next, a Generalized SVD is performed on X (Abdi et al., 2013). Mathematically, it can be expressed as (Abdi et al., 2013)

$$\mathbf{X} = \mathbf{P}\mathbf{\Delta}\mathbf{Q^T} \tag{6}$$

with

$$\mathbf{P^T}\mathbf{M}\mathbf{P} = \mathbf{Q^T}\mathbf{A}\mathbf{Q} = \mathbf{I} \tag{7}$$

In equation (6) the column vectors of matrix $\mathbf{P}$ and $\mathbf{Q}$ describe a principal component. The factor scores are stored in a matrix $\mathbf{F}$ and the loadings in $\mathbf{Q}$ by the following simple mathematical manipulation (Abdi et al., 2013)

$$\mathbf{X} = \mathbf{F}\mathbf{Q^T} \tag{8}$$

with

$$\mathbf{F} = \mathbf{P}\mathbf{\Delta} \tag{9}$$

3.3 Logistic Regression

Suppose, we want to predict the outcome variable using k independent variables, X_i, we can leverage an *logistic regression model*. Mathematically, we can describe it as (Sharma, 1995)

$$ln\Big(\frac{p}{1-p}\Big) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k \tag{10}$$

Equivalently, the equation (10) can be re–written in the following form (Sharma, 1995)

$$ln\Big(\frac{p}{1-p}\Big) = \beta_0 + \sum_i \beta_i X_i \tag{11}$$

$$p = \frac{1}{1 + e^{-\beta_0 + \sum_i \beta_i X_i}} \tag{12}$$

3.4 Support Vector Machines

Support Vector Machines (SVM) illustrated by Cortes et al. (Cortes and Vapnik, 1995), are a class of large–margin classifier. Support Vector Machines as outlined earlier are a class of classifiers that determine a decision surface that is furtherest from any data–point (Manning et al., 2008), and *Support Vectors* are the subset of data–points that define the location of the decision surface. The SVM model for two-way classification problem is given in equation (13) (Manning et al., 2008). Where, $\vec{w}$, represents a weight vector and the intercept term b is used to define the decision hyperplane (Manning et al., 2008). A non–zero α_i points to the fact that the i^{th} data–point viz. $\vec{x}_i$ is a support vector (Manning et al., 2008).

$$f(\vec{x}) = \text{sign}\left(\sum_i \alpha_i y_i \vec{x}_i^T \vec{x} + b \right) \tag{13}$$

By utilizing a *kernel trick*, we can provide a mapping of data points from lower dimension to higher dimensions (Manning et al., 2008). In this work we analyze with different kernel functions including linear kernel and polynomial kernel. From, equation (13), the kernel function is a simple dot product of data point vectors ie. $K(\vec{x}_i, \vec{x}_J) = \vec{x}_i^T \vec{x}_j$ (Manning et al., 2008). This simple substitution transforms the equation into equation given below (Manning et al., 2008):

$$f(\vec{x}) = \text{sign}\left(\sum_i \alpha_i y_i K(\vec{x}_i, \vec{x}) + b \right) \tag{14}$$

To transform the data points in lower dimensional space to a higher dimensional space, a transformation of the format $\Phi : \vec{x} \mapsto \phi(\vec{x})$ can be used (Manning et al., 2008). This then implies the simple dot product that becomes the following *Kernel Function* (Manning et al., 2008):

$$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i)^T \phi(\vec{x}_j) \tag{15}$$

A kernel function must satisfy *Mercer's condition* viz. it must be "continuous, symmetric and have a positive definite gram matrix" (Manning et al., 2008). If a kernel function does not satisfy such conditions then Quadratic Programming may not yield a answer for the optimization problem posed earlier for SVM's (Manning et al., 2008). Some kernel functions that satisfy Mercer's conditions and are popularly used are indicated below (Manning et al., 2008; Hornik et al., 2006):

- Linear Kernel Function

$$K(\vec{x}, \vec{z}) = (1 + \vec{x}^T \vec{z}) \tag{16}$$

- Polynomial Kernel Function

$$K(\vec{x}, \vec{z}) = (1 + \vec{x}^T \vec{x})^d \tag{17}$$

- Hyperbolic Tangent Kernel – where, C, is scaling constant and b, is a offset

$$K(\vec{x}, \vec{z}) = \tanh\left(C \cdot \vec{x}^T \vec{z} + b \right) \tag{18}$$

- The Bessel function of the first kind kernel

$$K(\vec{x}, \vec{z}) = \frac{\text{Bessel}^n_{(\nu+1)}(\sigma \|\vec{x} - \vec{z}\|)}{(\|\vec{x} - \vec{z}\|)^{-n(\nu+1)}} \tag{19}$$

- The Laplace Radial Basis Function (RBF) kernel

$$K(\vec{x}, \vec{z}) = exp\left(-\sigma \|\vec{x} - \vec{z}\| \right) \tag{20}$$

- The ANOVA radial basis kernel

$$K(\vec{x}, \vec{z}) = \left(\sum_{k=1}^{n} exp\Big(-\sigma (\vec{x}^k - \vec{z}^k)^2 \Big) \right)^d \tag{21}$$

- Gaussian Radial Basis Kernel Function

$$K(\vec{x}, \vec{z}) = exp\left(-\frac{(\vec{x} - \vec{z})^2}{2\sigma^2} \right) \tag{22}$$

4 Identifying Trustworthy Customer Reviews

4.1 Approach

For identifying trustworthy customer reviews, we train a supervised machine learning algorithm in two steps. First we identify and implement the useful features keeping in view their effectiveness. In addition we also utilized features leveraged in prior approaches (Jindal and Liu, 2008). We classify these features as per (Jindal and Liu, 2008)

into *Review Centric*, *Reviewer Centric*, and *Product Centric*. The features are tabulated in Table 1. Thereafter we perform Multiple Factor Analysis on the feature vectors that were generated by applying the features as described in Table 1. The result of MFA gives us with orthnormal independent dimensions (principal components) upon which we project the individuals (observations).

As a baseline model we omit MFA as a preprocessing step for the classifiers taken into consideration. Using this naïve approach we wanted to see to what degree MFA assists in the classification task of detecting trustworthy and non–trustworthy reviews. To label each individual observation as trustworthy or not we take up an intutive idea that takes into account the number of helpful feedbacks obtained for each of the customer reviews. We label a customer review as helpful if the total percentage of helpful feedbacks for that review is greater than equal to 75 %.

4.2 Feature Extraction

In this section we explain the various features that we take into account for the Amazon customer review data set[5] (McAuley and Leskovec, 2013). We leverage several features identified by prior approaches such as (Jindal and Liu, 2008) for the task of review spam analysis. We also augment them with several additional characteristics. The feature set utilized by us is tabulated in Table 1. We explain the intuition behind these features in the following subsections.

4.2.1 Review Centric Feature Group

Review centric features, as the name suggests are measures extracted from the review written by the online users. We have also taken into account the titles of the reviews that the users assign. Below we describe the explanations of the features listed in Table 1.

- **Length of the review title** and **length of review body** measure in the number of words captured in the body and title of the customer review. As indicated in (Jindal and Liu, 2008), Jindal et al. predict that long customer reviews tend to be more helpful; which we have also found in a case study we conducted utilizing MFA.

[5] https://snap.stanford.edu/data/web-Amazon.html

Feature Group	Feature
Review Centric	Length of Review Title
	Length of Review Body
	Contextual sentiment polarity of review body
	Contextual sentiment polarity of review summary
	Cosine similarity between review and the title
	Percentage of numericals
	Percentage of capitals
	Percentage of all capitals
	Rating of review
	Deviation of rating
	(Flag) Review is good
	(Flag) Review is average
	(Flag) Review is bad
	Time of postings of customer review
Reviewer Centric	Ratio of reviews written by the reviewer
	(Flag) Reviewer gives only good rating
	(Flag) Reviewer gives only average rating
	(Flag) Reviewer gives only bad rating
	(Flag) Reviewer gives both good and bad rating
	(Flag) Reviewer gives both good and average rating
	(Flag) Reviewer gives both bad and average rating
Product Centric	Price
	Average Rating

Table 1: Groupings of features used for MFA

- **Contextual sentiment polarity** is computed for both the customer review body and the review title. For the purpose of identifying the sentiment of given piece of text, the authors of (Rinker, 2013) utilize a "sentiment dictionary" (Hu and Liu, 2004) that is used to label the words carrying opinions. After identifying the "polarized" words (i.e. the words carrying sentiment), a "context cluster" is created by taking into account words surrounding the "polarized" cluster (Rinker, 2013). The "context cluster" is denoted by x_i^T and the words are considered as valence shifters (Rinker, 2013). The words in x_i^T are further labeled as –"neutral (x_i^0), negator (x_i^N), amplifier(x_i^a), or de–amplifier (x_i^d)" (Rinker, 2013). Each opinionated word is then assigned a weight, w, based on the amount of "valence shifters" as well as its position (Rinker, 2013). The final sentiment score is determined following the approach as described in Rinker et al. in (Rinker, 2013). It first adds the "context clusters" together. Next, the sum is divided by the square root of the number of words. $\delta = \frac{x_i^T}{\sqrt{n}}$

By capturing these feature we wanted to see if there was an agreement with the rating indicated by the user in the reviews. We also wanted to capture the true sentiment of the user writing the reviews and if this was predictor of whether a review was trustworthy or not.

- **Cosine similarity** between the review and the product title captures whether the reviewer has included a lot of technical details about the product or not. We wanted to see if a review, rich in technical description, is trustworthy or not. Also it is used to find the similarity between two reviews with the intuition that fake or genuine reviews have certain similarities in textual contents.
- **Percentage of numericals**, as outlined by Jindal et al. (Jindal and Liu, 2008) indicates a very technically oriented review. Percentage of capitalized letter, and all capitalized characters are indicators of not well crafted reviews (Jindal and Liu, 2008). We wanted to see if there was any relation between customer reviews containing a lot of technical details and a trustworthy review. Also, we wanted to see how not well drafted reviews correlate with trustworthy reviews.
- **Rating of review**, **deviation of rating**, **review is good, average, and bad** are all review rating related features. We wanted to see how the ratings and their associated flags correspond with trustworthy reviews. It may often be the case that rating which are exceptionally high do not necessarily correspond to trustworthy reviews or a thoughtful average rating may correspond to a helpful review.
- Further, we take into account the **time of postings of customer review** to see if the time a customer review was posted could be potentially linked to whether a customer review was trustworthy or not. For example, a review which was posted very late could potentially be of no help to the reader of the review.

4.2.2 Reviewer Centric Feature Group

Reviewer centric features were designed to capture various attributes related to the user writing the reviews, as the motivation given by Jindal et al. (Jindal and Liu, 2008). Given below are the detailed explanations of the features in this group :

- **Ratio of reviews written by reviewer** checks whether a user who is extremely vocal, writes trustworthy reviews or not.
- There are other features which are defined based on the observations such as reviewer gives only **good, average, and bad or a combination** of such ratings. These features try to capture if a variation in review rating given by the user is indicator of trustworthy reviews (Jindal and Liu, 2008). Also these are the good indicators of the biased reviews that users write in favor of a particular brand.

4.2.3 Product Centric Feature Group

The last feature group is product centric feature group, which captures two features *price* and *average rating of the product*. Following set of features are included under this set of features :

- As mentioned by Jindal et al. (Jindal and Liu, 2008), we wanted see if **price** point of a product (cheap or expensive) could influence the reviewer in writing less trustworthy reviews.
- Similarly we wanted to see if the **average rating of the product** could evoke the same response.

5 Results and Analysis

We consider customer reviews from each of the product categories in the Amazon customer review data set[6] (McAuley and Leskovec, 2013). The training set was constructed by taking 80% customer reviews and holding out 20% as a test set. The datasets were generated using the sampling without replacement method. Please note that rather than manually labeling the reviews for supervised classification, we utilize the "helpfulness" votes as a measure to determine if a customer review was trustworthy or not. This method of labeling the corpora is akin to utilizing a crowd–sourcing platform to determine the trustworthiness of customer review. This seems to be very simple yet intuitive approach to label a Web–scale corpora.

We present results for the various classifiers considered for the task of determining trustworthy customer reviews. For training of the Logistic Regression (LR) model we utilize the R statistical computing programming language. The SVM models for the various kernel functions were created using the algorithmic implementation by Karatzoglou A. et al. (Karatzoglou et al., 2004) in the R statistical computing programming language. We performed MFA using the algorithmic implementation given by Husson et al. (Husson et al., 2014).

For the baselines we measure the performance of the classifiers trained without the MFA as a preprocessing step. Results for the LR model and

[6] https://snap.stanford.edu/data/web-Amazon.html

SVM models are presented in Table 2 and Table 3, respectively.

Product Categories	AUC Value @ k			
	2500	5000	7500	10000
Cell Phones & Accessories	0.72	0.74	0.75	0.77
Software	0.68	0.71	0.71	0.72
Clothing & Accessories	0.78	0.79	0.78	0.80
Amazon Instant Video	0.69	0.70	0.71	NA†
Video Games	0.74	0.73	0.76	0.77
Home & Kitchen	0.76	NA†	NA†	NA†
Electronics	0.72	0.75	0.75	0.77

†k number of customer reviews not available for this category

Table 2: AUC values for the ROC curves obtained from Liner Regression model trained for various product categories at *www.Amazon.com*

Thereafter we integrate MFA and LR model together (named as MFALR), and its results are reported in Table 4. We perform a similar exercise with MFA and SVM classifier (MFASVM) and obtain the AUC values for different kernel functions described in Section 3.4. The results are reported in Table 5.

5.1 Analysis

Comparing the baseline logistic regression model versus the MFARA, we see that the proposed approach attains performance increments for all the product categories. For some product categories such as *Clothing & Accessories* and *Software* our proposed approach encompassing MFA in the classification task along with LR model achieves impressive accuracies. Considering the baseline SVM model and the MFASVM model, we observe that for all the product categories, utilizing MFA as a pre-processing step aids in classification of trustworthy vs. non–trustworthy reviews.

We have shown that by utilizing an intuitive concept of using helpfulness scores in labeling a web–scale corpora we are able to achieve good classification accuracies in terms of AUC values. This is comparable to the prior approaches that utilized human evaluators (Jindal and Liu, 2008). However it is also to be noted that this direct comparison will not be fair as the experiments reported in (Jindal and Liu, 2008) were carried out in a different setting. Contrasting our approach of utilizing MFA as a pre-processing step for classification task and state-of-the-art approaches leveraging standard stand-alone classifiers, we see that

Product Categories	Kernel	AUC Value @ k			
		2500	5000	7500	10000
Cell Phones & Accessories	Polynomial	0.62	0.64	0.66	0.67
	"Gaussian" RBF	0.55	0.57	0.59	0.61
	Linear	0.64	0.66	0.67	0.69
	Hyperbolic Tangent	0.54	0.55	0.57	0.58
Electronics	Polynomial	0.65	0.67	0.68	0.68
	"Gaussian" RBF	0.50	0.50	0.52	0.53
	Linear	0.49	0.63	0.64	0.65
	Hyperbolic Tangent	0.50	0.50	0.50	0.50
Clothing & Accessories	Polynomial	0.81	0.86	0.86	0.89
	"Gaussian" RBF	0.72	0.73	0.73	0.75
	Linear	0.67	0.69	0.73	0.74
	Hyperbolic Tangent	0.59	0.61	0.62	0.64
Amazon Instant Video	Polynomial	0.49	0.50	0.51	NA †
	"Gaussian" RBF	0.54	0.55	0.56	NA †
	Linear	0.50	0.53	0.55	NA †
	Hyperbolic Tangent	0.47	0.48	0.51	NA †
Software	Polynomial	0.63	0.66	0.65	0.66
	"Gaussian" RBF	0.62	0.63	0.64	0.64
	Linear	0.64	0.67	0.69	0.70
	Hyperbolic Tangent	0.54	0.55	0.57	0.57
Video Games	Polynomial	0.59	0.62	0.64	0.65
	"Gaussian" RBF	0.54	0.54	0.55	0.57
	Linear	0.62	0.65	0.67	0.70
	Hyperbolic Tangent	0.48	0.49	0.55	0.57
Home & Kitchen	Polynomial	0.51	NA †	NA †	NA †
	"Gaussian" RBF	0.50	NA †	NA †	NA †
	Linear	0.63	NA †	NA †	NA †
	Hyperbolic tangent	0.50	NA †	NA †	NA †

†k number of customer reviews not available for this category

Table 3: AUC values for the ROC curves obtained from SVM model trained for various product categories at *www.Amazon.com*

our approach is promising with respect to the accuracy values as reported by Jindal et al. (Jindal and Liu, 2008). Ghose et al. (Ghose and Ipeirotis, 2007) reported the performance for regression model trained on data set comprising of a subset of electronic categories such as *Audio-Video* and *Digital Camera*. Our proposed approach encompassing MFA performs at an impressive rate in comparison to the approach presented by Ghose et al. (Ghose and Ipeirotis, 2007).

The key advantages of our proposed approach are as follows:

1. Using Multiple Factor Analysis as a pre-processing step we have been able to take into account the hierarchy and grouping of feature space whilst training machine learning algorithms. MFA is able to assign first singular value corresponding to each grouping of features. This can assist us in training of machine learning algorithms as learner do not take into consideration the grouping / hierarchy of features identified.
2. Also, utilizing MFA as a pre-processing step

Product Categories	AUC Value @ k			
	2500	5000	7500	10000
	2500	5000	7500	10000
Cell Phones & Accessories	0.73	0.74	0.77	0.79
Software	0.70	0.71	0.72	0.74
Clothing & Accessories	0.78	0.78	0.80	0.81
Amazon Instant Video	0.69	0.71	0.73	NA†
Video Games	0.76	0.73	0.78	0.78
Home & Kitchen	0.78	NA†	NA†	NA†
Electronics	0.74	0.75	0.77	0.79

†k number of customer reviews not available for this category

Table 4: AUC values for the ROC curves obtained from MFA + logistic regression model trained for various product categories at *www.Amazon.com*

Product Categories	Kernel	AUC Value @ k			
		2500	5000	7500	10000
Cell Phones & Accessories	Polynomial	0.66	0.67	0.69	0.69
	"Gaussian" RBF	0.56	0.56	0.58	0.59
	Linear	0.64	0.67	0.69	0.71
	Hyperbolic Tangent	0.56	0.56	0.58	0.59
Electronics	Polynomial	0.67	0.69	0.69	0.70
	"Gaussian" RBF	0.68	0.69	0.71	0.71
	Linear	0.65	0.69	0.72	0.73
	Hyperbolic Tangent	0.54	0.56	0.57	0.57
Clothing & Accessories	Polynomial	0.82	0.84	0.85	0.86
	"Gaussian" RBF	0.73	0.75	0.75	0.77
	Linear	0.71	0.73	0.75	0.77
	Hyperbolic Tangent	0.62	0.63	0.64	0.65
Amazon Instant Video	Polynomial	0.50	0.50	0.52	NA †
	"Gaussian" RBF	0.54	0.56	0.58	NA †
	Linear	0.52	0.55	0.57	NA †
	Hyperbolic Tangent	0.48	0.50	0.50	NA †
Software	Polynomial	0.68	0.69	0.69	0.71
	"Gaussian" RBF	0.64	0.65	0.66	0.68
	Linear	0.65	0.68	0.71	0.72
	Hyperbolic Tangent	0.54	0.56	0.56	0.57
Video Games	Polynomial	0.65	0.68	0.68	0.69
	"Gaussian" RBF	0.54	0.56	0.58	0.59
	Linear	0.64	0.66	0.68	0.71
	Hyperbolic Tangent	0.54	0.56	0.57	0.59
Home & Kitchen	Polynomial	0.60	NA †	NA †	NA †
	"Gaussian" RBF	0.70	NA †	NA †	NA †
	Linear	0.67	NA †	NA †	NA †
	Hyperbolic tangent	0.54	NA †	NA †	NA †

†k number of customer reviews not available for this category

Table 5: AUC values for the ROC curves obtained from MFA + SVM model trained for various product categories at *www.Amazon.com*

we are able to scale continuous as well as nominal attributes on the most pertinent principal components identified by MFA. Having the feature vectors scaled, aids in improving the performance of the supervised machine learning algorithms. As we can see when comparing the AUC values of the baseline model and our proposed MFASVM model (MFA used as pre-processing of SVM), we see that our proposed approach does better job in identifying the trustworthy reviews.

6 Conclusions

In this paper we propose a novel method to classify a online review into trustworthy or non–trustworthy. Our results show that the proposed approach of taking into account MFA as pre-processing step for classification task increases the performance of the classifiers. We have also shown how the performance of classifier improves with the increment in the size of the training data. We additionally see that our proposed approach utilizing MFA achieves results comparable to the state–of–the–art–systems.

In the present work we utilize the concept of helpfullness votes to prepare the datasets for the experiments. This process contributes to false positive and false negative examples in the training data. Some amount of manual labeling could be useful to tackle this problem. As part of our future work we would like to see how the concept of classifier ensemble can help in classification of trustworthy reviews. Jindal et al. (Jindal and Liu, 2008), indicate that certain machine learning approaches such as SVM and Bayesian are not up to par in classification of review spam. We hypothesize that, by employing ensemble learning we can further improve the perofrmance. We also plan to use the concept of active learning method for creating the data automatically. We would also focus to identify more features for the target problem.

References

Hervé Abdi, Lynne J Williams, and Domininique Valentin. 2013. Multiple factor analysis: principal component analysis for multitable and multiblock data sets. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(2):149–179.

Herv Abdi1 and Dominique Valentin. 2007. *Multiple Factor Analysis*. Neil Salkind (Ed.), Encyclopedia of Measurement and Statistics.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Brigitte Escofier and Jérôme Pagès. 1994. Multiple factor analysis (afmult package). *Computational statistics & data analysis*, 18(1):121–140.

Anindya Ghose and Panagiotis G. Ipeirotis. 2007. Designing novel review ranking systems: predicting the usefulness and impact of reviews. In *ICEC*, pages 303–310.

Kurt Hornik, David Meyer, and Alexandros Karatzoglou. 2006. Support vector machines in r. *Journal of statistical software*, 15(9):1–28.

Minqing Hu and Bing Liu. 2004. Mining opinion features in customer reviews. In Deborah L. McGuinness and George Ferguson, editors, *AAAI*, pages 755–760. AAAI Press / The MIT Press.

Francois Husson, Julie Josse, Sebastien Le, and Jeremy Mazet, 2014. *FactoMineR: Multivariate Exploratory Data Analysis and Data Mining with R*. R package version 1.26.

Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *WSDM*, pages 219–230.

Nitin Jindal, Bing Liu, and Ee-Peng Lim. 2010. Finding unusual review patterns using unexpected rules. In *CIKM*, pages 1549–1552.

Ian Jolliffe. 2005. *Principal component analysis*. Wiley Online Library.

Alexandros Karatzoglou, Alex Smola, Kurt Hornik, and Achim Zeileis. 2004. kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20.

Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. 2010. Detecting product review spammers using rating behaviors. In *CIKM*, pages 939–948.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge University Press.

Julian J. McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In Qiang Yang, Irwin King, Qing Li, Pearl Pu, and George Karypis, editors, *RecSys*, pages 165–172. ACM.

Bo Pang and Lillian Lee. 2007. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Tyler W. Rinker, 2013. *qdap: Quantitative Discourse Analysis Package*. University at Buffalo/SUNY, Buffalo, New York. version 1.3.5.

Subhash Sharma. 1995. *Applied multivariate techniques*. John Wiley & Sons, Inc.

Peter D. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *ACL*, pages 417–424. ACL.