2017

Impact of reviewer social interaction on online consumer review fraud detection

Kunal Ketan Goswami, *San Jose State University*
Younghee Park, *San Jose State University*
Chungsik Song, *San Jose State University*

Goswami *et al. J Big Data (2017) 4:15*
DOI 10.1186/s40537-017-0075-6

Journal of Big Data

RESEARCH Open Access

Impact of reviewer social interaction on online consumer review fraud detection

Kunal Goswami, Younghee Park* and Chungsik Song

*Correspondence:
younghee.park@sjsu.edu
Department of Computer Engineering, San Jose State University, San Jose, USA

Abstract

Background: Online consumer reviews have become a baseline for new consumers to try out a business or a new product. The reviews provide a quick look into the application and experience of the business/product and market it to new customers. However, some businesses or reviewers use these reviews to spread fake information about the business/product. The fake information can be used to promote a relatively average product/business or can be used to malign their competition. This activity is known as reviewer fraud or opinion spam. The paper proposes a feature set, capturing the user social interaction behavior to identify fraud. The problem being solved is one of the characteristics that lead to fraud rather than detecting fraud.

Methods: Neural network algorithm is used to evaluate the proposed feature set and compare it against the state-of-the-art feature sets in detecting fraud. The feature set considers the user's social interaction on the Yelp platform to determine if the user is committing fraud. The neural network algorithm helps in comparing the feature set with other feature sets used to detect fraud. Any attempt to find the characteristics that lead to fraud has a prerequisite to be good enough to detect fraud as well.

Results: The F1 score obtained using neural networks is on par with all the well-known methods for detecting fraud, a value of 0.95. The effectiveness of the feature set is in rivaling the other approaches to fraud detection.

Conclusions: A user's social interaction on a digital platform such as Yelp is equally important in evaluating the user as social interaction is in real life. The characteristics that lead to fraud can be intuitively captured. The characteristics such as number of friends, number of followers and the number of times the user has provided a review which was helpful to multiple people provide the neural network with a base to form a relationship between opinion fraud and social interaction characteristics.

Keywords: Opinion spam, Neural networks, Social interactive behavior

Background

Online consumer reviews of products and businesses have become increasingly popular and valuable sources for consumers to make a decision. Examples include restaurant reviews on Yelp, product reviews on Amazon, hotel reviews on TripAdvisor, and many others. As online-shopping, e-commerce and social networks are ubiquitous and become an integral part of our daily lives, these reviews have direct influence on product and business sales [7, 26]. Since such user-generated content contains rich information about user experiences and opinions, it is useful for potential customers to make better

decisions about spending their money, and also helps merchants improve their products, services, and marketing. Yelp, one of the leading consumer review web sites, alone contains more than 100 million reviews of businesses, with a market capitalization of roughly four billion dollars.

However, the credibility of these reviews is fundamentally undermined when businesses commit review fraud, creating fake reviews for themselves or their competitors. Recent research has reported that one-third of all consumer reviews on the Internet are estimated to be fake [38]. The financial benefits reaped from such fake reviews have even created a market of paid users. They are paid to counterfeit fake reviews either to fabricate hype to promote a business or to tear down competitive products or businesses. This could be collectively identified as opinion fraud or opinion spam.

Opinion fraud detection problem involves finding a fake review among all the reviews on the site. For each review, we have the text, the author (the reviewer to be precise), the product it was written for, the time stamp of posting, and evaluation. Typically, no user profile is available but additional information for products and for reviews are available depending on site. The problem of opinion fraud detection has been addressed by focusing on the review of text information [10, 31], and by behavioral approaches that utilize the behavior of fake users [18, 19, 24]. The problem has often been approached as a supervised classification problem with two classes, fraud and not-fraud. However, to obtain training data by manually labeling reviews is very hard and ground truth information is often unavailable, which makes training supervised models less attractive for this problem. There has been an increasing interest in the study of unsupervised, general, and network-based approaches, to tackle the opinion fraud detection problem in online review data [1, 16, 24, 30, 36, 39, 42].

Opinion fraud is often conducted by groups of people [36, 41], Latent Collusion Model [41] has been proposed to detect the groups as well as detect fraud in online reviews. This model focuses on finding the unique behavioral patterns in the reviews to spot fraudulent users, collaborating for opinion spam. Along with other recent studies [1, 36], this work serves as an inspiration for us to pursue user behavior for fraud detection.

Consumers on Internet are not buyers or shoppers just for convenience. They communicate with each other, forming so called social networks and build up virtual social interactions by sharing their opinion and experience. Many users on Yelp.com, for example, are connecting online, sharing their opinion and experience of local businesses, favorite types of food, getting together in real life and more. This social networking aspect of Yelp is truly what makes it the useful local guide it is today. Most of the reviews that are posted on the site come from a rich network of people who are connecting and expressing their love of local business and stores. And with over 100+ million reviews to date, more and more people continue to join the network to share their experiences in their own communities.

In this paper, we analyze Yelp users with a slightly deeper perspective of social interaction and study impacts of this social interactive behavior on classification of spammers and non-spammers. We consider features such as the total votes a user has received from public, the total number of followers a user has on Yelp, the total number of friends a user has on Yelp, the total number of pictures a user has posted for various businesses. These features represent how socially active a user is, how a user is more likely to have

socially active friends and how a user is more likely to try out new businesses whenever possible. This additional set of social behavioral features about reviewers is used to improve the classification result on real-life opinion spam data.

Yelp uses a filtering algorithm to filter fake or suspicious reviews and puts them in a filtered list. Social interactive behavioral features are analyzed and compared between two user groups: one for a reviewer group whose reviews have been filtered as reviews not thought to be genuine reviews (spam), which we call the "fake review group"; another for the group whose reviews we believe, genuinely evaluate the business or the product (non-spam), which we call the "recommended review group". Among various social interactive features, total number of friends and total number of followers show the most discriminating behaviors to demarcate spammers and non-spammers.

These social behavioral features are used with a neural network to estimate the genuineness of a user. We take Yelp's filtered reviews as a proxy for review fraud and train the neural network for predicting the social activeness. This social activeness is the center of our hypothesis, and a neural network seems to be a good fit for its calculation. Despite its need for a longer training time, a neural network has good convergence rate when it comes to classification problems. We evaluate social activeness of the user as a binary classification output, indicating whether a user is active or inactive. Despite the advantages provided by unsupervised learning, we select a supervised algorithm to evaluate our feature set as we are more focused on determining the characteristics which lead to fraud. As such, the problem has a prerequisite that fraudulent users are known beforehand to determine social interaction features that lead to fraud. We evaluate the features we selected by comparing the performance with other well known feature sets, we inculcate each of our features one after another and observe the changes in the performance metrics such as accuracy, precision, recall and F1 score. With all the given benefits of unsupervised learning, we focus our efforts on supervised learning applied to opinion fraud detection. With supervised learning, we focus on understanding the characteristics of the user which lead to opinion fraud. This perspective has led us to discover this unique social interaction feature set which captures the user-specific information and user's interaction with the platform.

The rest of this paper is organized as follows, section "Related work" presents a literature review of other work carried out in a similar direction. Section "System architecture" provides a detailed overview of our system and discusses the feature sets. Section "Neural networks: an overview" provides a brief explanation on neural networks and how the model is trained, followed by the evaluation of our model in section "Evaluation" and we conclude our work in section "Conclusions".

Related work

Neural networks are widely used for image processing applications such as face recognition and face detection [22, 37, 44]. Also, neural networks are the algorithm of choice when it comes to autonomous driving [34, 35]. They're also used for pattern recognition [33], atmospheric sciences [12], electromagnetism [8], energy systems [20], etc. The popularity of neural networks is due to the fact that they converge on a varied set of problems, also the fact that they're an imitation of how our brain learns new skills, retains that information for future use and applies it when necessary. Such qualities of a neural

network certainly make it an alluring target when it comes to opinion spam detection, which is a crucial problem nowadays, with the rise of the Internet and the number of people voicing their opinions for the general public to know.

Online reviews significantly affect the performance of a business [26], Luca discovers that a one star increase on Yelp.com review can increase the revenue by 5–9% and these ratings are dependent on the specific restaurant in case of chain restaurants. Furthermore his work analyzes how customers use review information and whether or not that use is similar to popular learning models. A customer's rating for a business is dependent on the average rating of the business and not on the number of friends the customer has. Mayzlin et al. [28] study reviews on popular hotel websites and deduce that *neighbor characteristics and own hotel characteristics* affect review manipulation. Anderson [2] presents evidence that a majority of deceptive reviews on private websites are negative with a motive to decrease overall sale of the product. Another study [3] documents that approximately 5% of reviews posted on websites are by users who have no records of purchasing that product or anything on that particular website. A meta-analysis on online reviews [11] focuses on understanding the effect of online reviews on retailers' performance and revenue using sales elasticity. An analysis on online review elasticity [43] present that *volume and valence elasticity is higher* for lesser known products which are sold by a couple of online retailers and that volume elasticity is higher on the product market whereas valence elasticity is higher on community markets. A very recent approach to identify fraud in Yelp reviews addresses the economic incentives for a particular business to conduct review fraud [27], Luca and Zervas with the help of the information provided by Yelp about one of it's sting operations derive that filtered reviews on Yelp tend to lie on extremes and that a restaurant is more likely to commit review fraud when it's reputation is relatively weak, basically when it has less reviews. They also discover that restaurants that have a chain are less likely to be involved in fraudulent activities and increased competition for a restaurant also increases the number of unfavorable fake reviews. However, they do not consider specific features individual to users in order to detect fraud users and are more centered towards determining the economic incentives for the businesses. Such studies bring to light how crucial is review credibility, making review fraud detection necessary to sustain the online consumer market.

Jindal and Liu [17] with their work on review fraud detection mark the beginning of a different class of problem solving, detecting patterns in online reviews to classify them as genuine/fake. Considering duplicate reviews as spam reviews for training their classifier, they experiment with Naive Bayes classifier obtaining a precision of 4–5% and logistic regression on online reviews from amazon.com. One of their most important insight was the fact that fraud detection not only depends on the features from the review but also from the features obtained from the reviewers. A detailed approach in [18] identifies opinion spam as *untruthful reviews, reviews on brand instead of product and advertisements* and uses duplicate reviews as spam for classification purposes as well. Jindal et al. [19] further take the approach of fake review detection by considering it similar to web search result spam detection, analyzing reviews from amazon.com to obtain unusual patterns which indicate fraud behavior. They took up a heuristic approach to define the rules that indicate spam behavior in the users, for e.g. if a particular user has been posting only negative reviews for a particular product or brand, that user is considered

suspicious and possibly a spammer. They define a set of expected and unexpected rules, their expectations being no relation between the data attributes and classes upon which stronger co-relations are investigated.

Scoring reviews and reviewers [25] is another approach to spam detection wherein Lim et al have considered the average review content similarity of amazon.com reviews and rating spamming by detecting multiple reviews with the same rating from the reviewer. Once such groups are identified, they can be targeted further to identify fake reviewers. Extracting review specific and reviewer specific features [23] such as content similarity, sentiment analysis, rating deviation and brand deviation is a more robust way to identify fake reviews. Li et al. construct a supervised classifier on top of these features to provide a semi-supervised solution to a majority of unlabeled data reaching upto an F1 score of 0.631.

By focusing on detecting spammer groups which target specific products, Mukherjee et al. [29] focus on frequent pattern mining and features of reviewers as individuals and as part of a group to calculate spam indicators and based on that detect spam. A detailed approach [30] which explains the frequent itemset mining used to detect fake review groups and explains the individual and member specific features. Ott et al. [32] take up the approach with supervised models of Naive Bayesian classifier to fraud detection in online reviews. They use n-grams to generate features from reviews and obtain ground truth labels using Amazon Mechanical Turk, delivering an accuracy of about 90%. However, on a platform such as Yelp, review text alone does not suffice in providing information about it being fake or genuine with the growing skills of fraudsters to generate reviews resembling genuine reviews.

With review graphs [39] the method to detect spam reviewers using a heterogeneous graph of product, review and the reviewer proposed by Wang et al turned the direction towards detecting relationships between these three nodes with parameters such as *trustworthiness of the reviewers, reliability of the stores and honesty of reviews* using the data from resellerratings.com. Akoglu et al. [1] also utilize a heterogeneous graph to determine fake reviewers on Yelp.com, making the framework independent of the domain under consideration they do not utilize review specific features. An extension to their work [36] focuses on the review specific features to improve the performance of the framework. A recent work by Ye and Akoglu [42] focuses on obtaining network footprint score to determine fake reviewers from genuine ones, they identify spam as a group activity instead of an individual activity and classifies spammers based on hierarchical clustering and a network footprint score. This score relies on the fact that real world networks are self similar [4, 6] and on the diversity in a review network. Their approach might be extended with review specific features which provide sentimental analysis and other statistics from the text.

These studies on opinion spam detection certainly point us towards the importance of this problem. However, we identify this problem with a different perspective. It is indeed necessary to determine which reviews are fraudulent reviews, but even more so it is important to identify the characteristics which lead to fraud. With that motivation in mind, we approached this problem in order to identify the distinctive user-specific features that lead us to discover fraud, or opinion spam in other words.

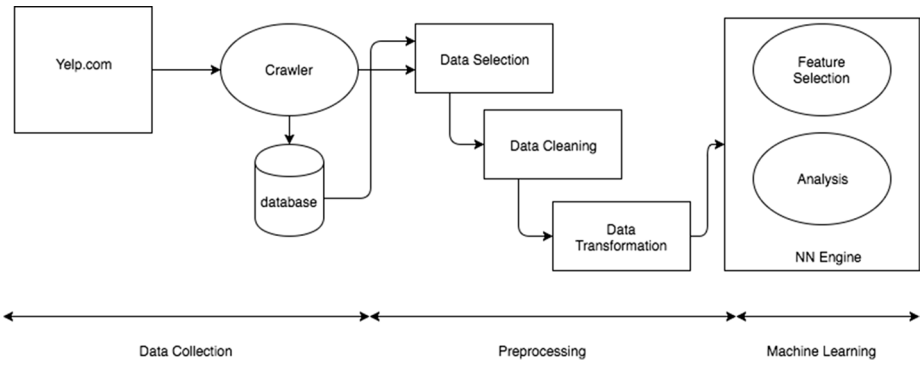

Fig. 1 System architecture for our framework

System architecture

The architecture diagram for our framework is shown in Fig. 1. Our framework consists of three major modules, a data collection module, which gathers the data for the purpose of our study. Data preprocessing module which cleans and transforms the collected data into a suitable form for the third module which is our machine learning engine. In this section we discuss the underlying processes of each of our modules and then take a look over individual features making up our social interaction feature set.

Data collection

As the name suggests, this module performs tasks related to gathering information, from which we derive insights at the end of our data mining cycle. For the purpose of our study, we designed a web crawler using the JSoup framework which starts with the home page of the Yelp website and extracts all links from that page. Once the links have been extracted, the crawler determines whether the link leads to a user page on Yelp or a business page on Yelp and accordingly parses it. Once parsed, the information is stored in a MySQL database for further access.

Data preprocessing

The data that has been collected from the website is not consistent in all cases, there are missing values for particular fields. Especially when a user has deactivated the account on Yelp or the user has been removed from the websites for violating the code of conduct, the reviews written by such users still exist on the business page and while computing we add null values to take care of such reviews. So basically, the users would have a default value of 0 for the respective features that we consider for evaluating whether or not they are fraudulent.

The machine learning algorithm in order to train upon the data needs it to be presented in a particular format, data transformation comes into picture for achieving the desired format. Particularly, in our case the machine learning algorithm is a neural network which accepts m inputs each of which contains n features and then trains upon it. These features contain precomputed values by combining some information from the user page on Yelp.

Machine learning engine

This module from our framework deals with the various feature sets under consideration and analyzes them in order to obtain insights from it. We consider various combinations of all the features that we gather from the website, the details of these combinations are discussed in further sections. These combinations were used with the neural network to analyze the feature specific impact on the overall performance of the algorithm. As we mentioned earlier, the convergence rate of neural network with classification makes it an attractive choice for our research. We also study the results to obtain insights about the importance of the respective feature in classification. Let us now discuss the user specific social behavioral features we consider for our study,

Social interaction features

In this section we discuss about meticulously evaluated user specific features which we categorize as user social interaction behavior features. Social interaction, in a broader sense indicates the level of activity of the person in a society. In the case of a website such as Yelp, this level of activity can be measured using the number of friends the user has, the interaction with the fellow users in terms of votes, compliments and followers, along with the total number of pictures posted by the user in the reviews for the respective business. We believe that these features define the activeness of a user on Yelp and the activeness in the community for discovering and reviewing various businesses. Furthermore, we investigate whether these features demarcate spammers and non-spammers. This section provides the intuition behind the selection of each feature for the proposed feature set, the reason behind the importance of the feature and its place collectively with the other features.

- Number of friends: Yelp allows users to invite their friends to join Yelp or to make new friends on Yelp. Friendship is a mutual relationship, which means that when a user adds another user as a friend, the first user will automatically be added as a friend of the second user. Users in the fake review group show the characteristics of one-time users [21] in a social network who have never made social connection with other users in the network.
- Number of votes from others: Each review is voted by viewers that corresponds to correct information on the businesses review page. Viewers have choices for the review being useful, funny or cool. Voting helps others determine whether or not people should try the business and says "Yes, the review is true to its word" or "it may be okay, but it got some funny things said within it" or "it got things I didnt even know about the location". A genuine user is more likely to have a decent number of votes indicating support from fellow users and general public which agrees with the user's opinion about a particular place.
- Photo count: Yelp added a feature to its service that highlights rich content added to the review. Not only are consumers able to write reviews about their experiences with local business, they are also able to upload photos what will now appear directly in-line with their text. A genuine user is more likely upload legitimate photos of the product or business to share real experience.

- Number of compliments: To facilitate conversation on Yelp, the site offers a set of compliments. Users give each other compliments directed either at the person or at their reviews. As communication solicits response, a compliment given is often a compliment returned. By structuring compliments, Yelp can improve its data on members and on businesses. Compliments also help users to be more precise even when they would rather not be.
- Number of followers: When a user chooses to follow another user on Yelp, the user will see other users reviews listed first in the default sort order in business pages, and their activities will be listed under the Following tab on own home pages recent activity feed. A genuine user is more likely to have an agreement from fellow users and general public for the user's opinion about a particular place or products. The audience keeps listening to their favored reviewers opinions and will have more chance to be followers.

As we saw for each feature, it provides us with the outline of the user's digital personality. For example, the number of relevant photographs allows us to determine the expressiveness of the user when it comes to the businesses he/she is trying to describe. The number of followers indicates how well written are the reviews and tips, those parameters prove as a standard to evaluate the user's judgment over a particular business. The more the people agreeing with the user's judgment, the more the likelihood of it being true. Correlating these features with our understanding of how a society functions allows us to define proper characteristics that each of these features indicate. The importance of each of these features is evaluated using a bag of decision trees and the results are discussed in a latter section. The results indicate the most important to the least important feature for classifying spam.

User behavior features

We evaluate our feature set with the help of user behavioral features [36] used in other related work to detect online opinion spam. We combine our features with this feature set to further evaluate the importance of respective social interaction features. The user behavior features include the maximum number of reviews written by a user in a day, indicating the user activity on the website. This feature set also considers the sentiment with which review was written, categorizing the sentiment in positive and negative classes. These features are specific to the user's activity on Yelp website, the social interaction features help us derive the characteristics specific to the user while on the other hand user behavior features allow us to explore the manner in which the user interacts with Yelp's platform. Another way for the user to interact with the platform is the rating distribution, which allows us to statistically estimate the ratings given by the user to various reviews along with the time in between a user was active on Yelp and the total number of tips he/she has given related to businesses for other users to refer to and take corresponding action.

Neural networks: an overview

Neural networks are an inherent part of our experiment. Before we go into the details of the results we obtained, let us first have a simple overview of what a neural network is. This section focuses on providing a brief introduction to neural networks and the algorithm we use for training. We believe that discussing about the specifics of the algorithm is important before we move on to the discussion of results.

A neural network comprises of multiple nodes connected with each other, each node representing the activation function, the sigmoid function in our case. Let $w_1 = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \end{bmatrix}^T$ be the weight vector to the input layer, considering the sigmoid function as $g(z)$ the outputs at the first layer would be computed as follows [5, 9, 15]:

$$\begin{aligned} a_{11} &= g\left(w_{11}^{(1)} \cdot x_1\right) \\ a_{12} &= g\left(w_{12}^{(1)} \cdot x_2\right) \\ a_{13} &= g\left(w_{13}^{(1)} \cdot x_3\right) \end{aligned} \tag{1}$$

Now these outputs $a_{1i's}$ will be the input to the hidden layer, the integration function for these outputs at any node is summation function, that is any node in the hidden layer would first compute the sum of all the inputs and then compute the respective output using sigmoid function. The computation is represented using the following equations:

$$\begin{aligned} a_{21} &= g\left(w_{11}^{(2)} \cdot a_{11} + w_{12}^{(2)} \cdot a_{12} + w_{13}^{(2)} \cdot a_{13}\right) \\ a_{22} &= g\left(w_{21}^{(2)} \cdot a_{11} + w_{22}^{(2)} \cdot a_{12} + w_{23}^{(2)} \cdot a_{13}\right) \\ a_{23} &= g\left(w_{31}^{(2)} \cdot a_{11} + w_{32}^{(2)} \cdot a_{12} + w_{33}^{(2)} \cdot a_{13}\right) \end{aligned} \tag{2}$$

For the hidden layer the weight vector is represented by

$$w_2 = \begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} & w_{13}^{(2)} \\ w_{21}^{(2)} & w_{22}^{(2)} & w_{23}^{(2)} \\ w_{31}^{(2)} & w_{32}^{(2)} & w_{33}^{(2)} \end{bmatrix}$$

The k in $w_{ij}^{(k)}$ represents the layer to which this set of weights is provided and in this case it's layer 2. These outputs now propagate to the last layer of the neural network architecture under consideration and this layer generates the probability vector for the given input.

$$z = g\left(w_{11}^{(3)} \cdot a_{21} + w_{12}^{(3)} \cdot a_{22} + w_{13}^{(3)} \cdot a_{23}\right) \tag{3}$$

For this layer the weight $w_3 = \begin{bmatrix} w_{11}^{(3)} & w_{12}^{(3)} & w_{13}^{(3)} \end{bmatrix}^T$ wherein, the k in w_{ij}^{k} represents the layer to which this weight is provided, in this case it'll be 3. Also in all the above mentioned equations,

$$g(z) = \frac{1}{1 + e^{-z}} \tag{4}$$

The above set of equations represent a procedure called as forward propagation, the name originates from the fact that at every layer the output is sent to the next layer for computation. This computation would occur with some initial set of weights at first to provide us with the predicted outcome on a particular set of inputs. The training of the neural network begins after these equations are computed, wherein the weights are adjusted simultaneously to work with the given set of inputs.

Training the neural network

For computing the correct set of weights for a given set of inputs, backpropagation algorithm [14, 40] is used. Before we jump to the details of this algorithm, let us first discuss about measuring the error.

Let's say that the label or the actual outcome given the set of inputs is y. And the output obtained from the neural network is z. The error would be calculated using the equation,

$$Error(z, y) = \frac{1}{2} \cdot ||y - z||^2 \tag{5}$$

This error would be computed for every single layer in the network and using this very error the weights would be adjusted to the given set of inputs. Let us have a look at the pseudocode for the backpropagation algorithm.

Algorithm 1 Backpropagation Algorithm

procedure TRAIN
  $X \leftarrow$ Training Data Set of size mxn
  $y \leftarrow$ Labels for records in X
  $w \leftarrow$ The weights for respective layers
  $l \leftarrow$ The number of layers in the neural network, 1...L
  $D_{ij}^{(l)} \leftarrow$ The error for all l,i,j
  $t_{ij}^{(l)} \leftarrow 0$. For all l,i,j
  For $i = 1$ to m
    $a^l \leftarrow feedforward(x^{(i)}, w)$
    $d^l \leftarrow a(L) - y(i)$
    $t_{ij}^{(l)} \leftarrow t_{ij}^{(l)} + a_j^{(l)} \cdot t_i^{l+1}$
  if $j \neq 0$ **then**
    $D_{ij}^{(l)} \leftarrow \frac{1}{m} t_{ij}^{(l)} + \lambda w_{ij}^{(l)}$
  else
    $D_{ij}^{(l)} \leftarrow \frac{1}{m} t_{ij}^{(l)}$
    where $\frac{\partial}{\partial w_{ij}^{(l)}} J(w) = D_{ij}^{(l)}$

Thus, the neural network calculates its weights in an iterative manner. It updates the weights in every iteration based on the error calculated for that iteration. This is a very basic algorithm for curve fitting, which has the intuition of improving by error. This algorithm first predicts the values, checks for the error and then tries to optimize the prediction. This optimization is carried out in such a way that the error is minimized. The learning rate is a very important parameter for the algorithm to converge, and we have used a value of "0.01" for the same purpose. The intuition behind this learning rate was to pick a value which is not too big nor too small for the weights to converge to the optimal solution. The important factor in this research was the value of the feature set,

rather than the parameters used to fine tune the neural network algorithm. It is indeed important to have an optimal model, but it was even more important to have a feature set which captures fraud from a user-specific perspective. And in this work, we managed to fine tune the neural network with minimal adjustments and discovered the feature set which can capture fraud through a user's perspective.

Evaluation

For the purpose of our experiment, we trained the neural network on various combinations of feature sets with the data we gathered. This model was then used to determine fraudulent users. The aim of this experiment was to analyze the significance of the social interaction features when it comes to fraud detection. This analysis was carried out on the basis of precision, recall and F1 score obtained by training and testing the neural network model on these feature set combinations. We observe a relatively smaller contribution to fraud detection using these social interaction features when combined with user behavior features. But when the neural network is trained using only social interactive features, we notice performance very similar to the one obtained using only user behavior features. This result shows that user social interaction features standalone prove to be an excellent way to detect opinion fraud, and can be used for analyzing it.

Dataset description

The data set for this study was collected from Yelp.com using a web crawler based on Jsoup [13]. The crawler starts with the Yelp homepage and it follows all the links present on that page, determining whether it is a user page or a business page and parsing it further along the lines to store the data in MySQL database. For the purpose of our experiments we took a subset of this dataset into consideration which consists of 135,413 reviews, out of which 103,020 were recommended reviews and 32,393 not-recommended reviews. These reviews were written by 66,936 unique users, consisting of 61,083 users are genuine users and 5853 are fake users.

The data contains some missing values for features and it is also possible that a user who wrote a review for a business has discontinued using Yelp or Yelp has terminated the user account for violating the code of conduct. The values are filled in with null and default values for respective features, along with some parsing in the textual data that was collected from the website. There is no guarantee that a user is going to follow all the conventions of ASCII standard, but for considering the same information for our computation we need the encoding to be in a specific format. The neural network algorithm expects the data to have the features specific to the user/reviewer, further more we have created 6 combinations of all the features that we collected.

Yelp developed a filtering algorithm that determines whether a review is to be published or not based on its credibility. The algorithm is used to flag suspicious reviews and to filter those from the main Yelp page. Roughly 16% of restaurant reviews are filtered by Yelp. Recent study by Luca and Zervas has estimated the evidence of review fraud and the conditions under which it is most prevalent. They assembled two complementary datasets from Yelp and provided empirical support for using filtered reviews as a proxy for review frauds [27]. We have grouped collected review datasets into reviews by the recommended review group (non-spam) and reviews by the fake review group (filtered).

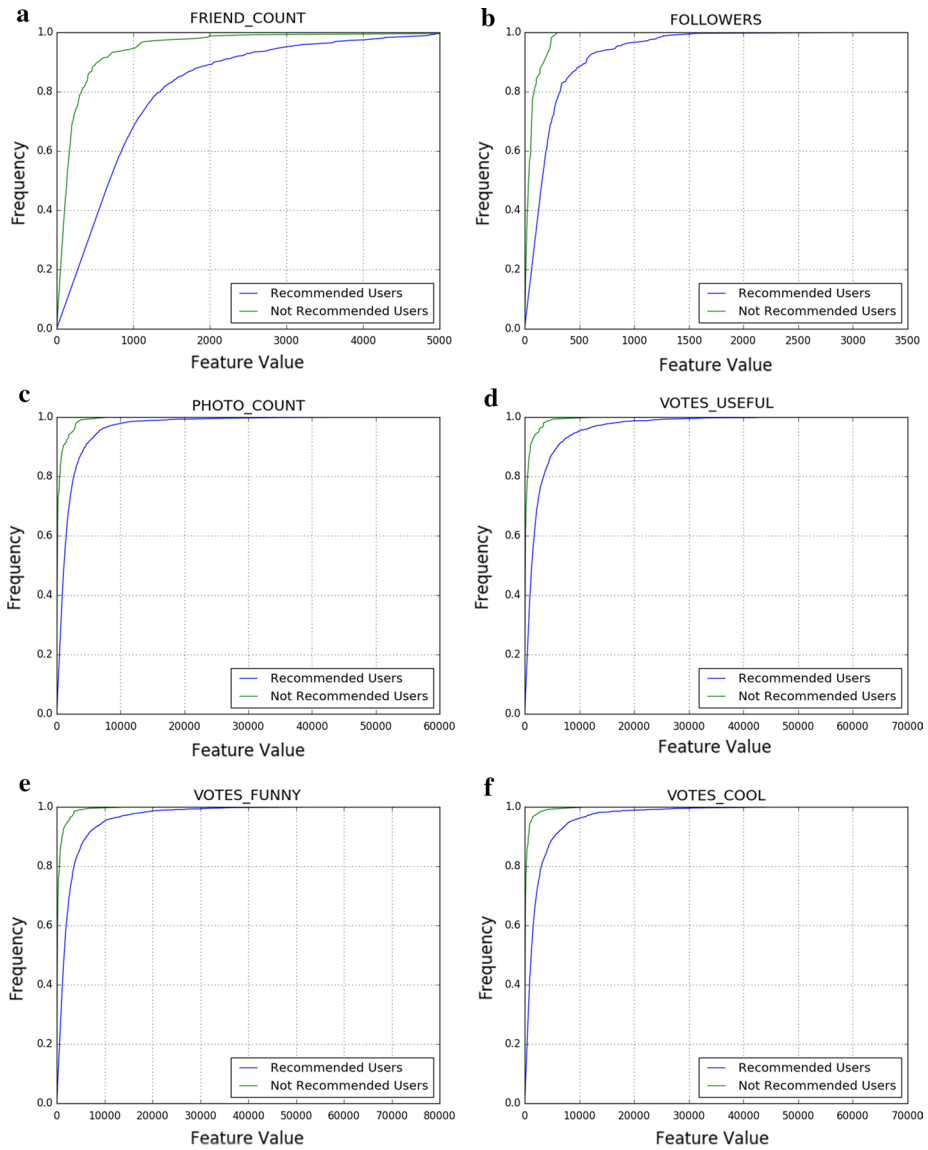

Fig. 2 Cumulative distribution function of social behavioral features **a** number of friends, **b** number of followers, **c** number of photos, **d** number of useful votes, **e** number of funny votes, **f** number of cool votes

In addition to grouping reviews, we also grouped reviewers in the Yelp dataset into the same two groups according to whether the reviews were considered to be recommended or fake. Simply, if Yelp's filter marked a review as not-recommended, that becomes a label for us associated to the particular reviewer as fake and vice-versa .

Figure 2 represents the cumulative distributive function for the user interaction features that we consider for our experiments. Each of the subplots represents the cumulative frequency for the % of users having that exact feature value under the categories that we have considered.

Experimental results

We cover the results obtained by experimenting with the neural network algorithm with our various combinations of the features. We record the accuracy, precision, recall, specificity and F_1 score for each of the experiments we conduct and compare these results to evaluate the particular feature set that we considered as well as determine feature importance.

The experimental results show that there is some improvement in the performance of the neural network algorithm as we add on user interaction features. A neural network behaves much in similarity with how our brain thinks and perceives information, it is an imitation of that behavior, the way in which we might derive better conclusions towards a subject based on better information or more relevant information, the neural network too will. However, we also conducted an experiment by training the neural network only with the social interaction features. And the results obtained from it are equivalent to the ones we obtain using user behavioral features. This could only indicate that the neural network is able to discern a relationship between the social interaction features and opinion fraud. An accuracy of 91.5% is a very good start and it is very much possible that it might improve with further research on it.

The table shows the results we obtained through the neural network on the subset of data which we considered. This table summarizes the results on 6 feature sets, wherein feature set 1 is the user behavioral features shown in Table 1. And we add one by one user interaction features shown in Table 2 to this feature set for obtaining the further feature sets. Incrementally adding such features gives us a chance to look at the individual feature importance. It also gives us an idea of what the impact is on overall performance of the algorithm. The state of the art works in this field can very possibly be identified as FraudEagle [1] and SpEagle [36] but both of them are unsupervised models, a comparison between our approach and these frameworks would mean comparing

Table 1 User behavioral features

Feature	Description
Maximum number of reviews by the user	Maximum number of reviews written in a day
Positive review ratio	Ratio of positive reviews (4–5 star)
Negative review ratio	Ratio of negative reviews (1–2 star)
User burstiness	Captures the burstiness related to the age of a user, defined as the number of days between their first and the last review
Average rating distribution	The rating distribution available on the users profile
Entropy of rating distribution	Entropy of rating distribution of users (products) reviews
Entropy of temporal gaps	Entropy of temporal gaps
Total review count	The total number of reviews a user has written
Tips	The total number of tips a user has given on Yelp

Table 2 User social interaction features

Feature	Description
Friend count	The total number of friends a user has on Yelp
Total photo count	The total number of photos a user posted on Yelp
Compliments	The total number of compliments a user has received
Followers	The total number of followers a user has
Total useful, funny and cool votes	The total useful, funny and cool votes a user has received for all the reviews

unsupervised learning with supervised learning. However, our focus is the social interactive behavior feature set and that is what we have evaluated with the neural network algorithm. Let us go over every measure that we record:

- Precision: Precision provides information of correctly predicted positive samples from the total number or positive predicted samples. Defining properly, it would be the ratio of true positives over the total number of positives that the classifier predicted. The precision increases as we add friend count and total photo count but it decreases by a significant amount when we sum up all the feature sets.
- Recall: Recall indicates the ratio of the true examples found from the predicted positive examples, that is out of n positive examples how many positive examples were found. Unlike precision, the recall does not have that big of a difference when all the feature sets are added, only a difference of 0.0058 is observed.
- Specificity: Specificity gives us an idea of the total number of negative samples being classified as negative by the classifier. Specificity goes to an all time low of 0.06 as compared to the user behavioral features which is 0.09.
- Accuracy: This measure is just the ratio of correctly predicted samples but does not give sufficient insight when the data is not in a good balance. That is, when the number of positive samples and negative samples in the data set do not match. The accuracy follows a similar trend as the precision with the addition of new features.
- F_1 score: The F_1 score can be interpreted as a weighted average of the precision and recall and gives us a better idea of how well the model is, overall. When we increment the basic feature set with friend count, total photo count and compliments the performance improves by 0.06% but on the other hand when the number of followers and the total number of votes are added to the feature set, the performance degrades.

As shown in the Table 3, the results we obtained provide improvement on a minute scale with the features that we add to the user behavioral feature set. The feature set, social interaction feature set that we propose can obtain a decent accuracy on detecting opinion spam. Though, these results aren't as good as user behavioral features, they significantly help the neural network distinguish user characteristics of being fake or genuine. And as the results endorse, a combination with friend count and photo count boosts up the over all performance with user behavioral features. Another observation that we would like to point out is, while preparing the results we trained the neural network with various sizes of training sets. Certainly, the data was biased with a smaller number of reviews, say 10,000. The number of fake reviews or "not-recommended" reviews for a business is less than the number of "recommended" or legitimate reviews for a majority of the businesses from the

Table 3 Results of neural network classifier

Feature set	1	2	3	4	5	6	7
Precision	0.915376	0.916781	0.916144	0.916144	0.915687	0.915719	0.915717
Recall	0.999836	0.999689	0.99982	0.99982	0.999591	0.999656	0.999623
Accuracy	0.915501	0.916906	0.916323	0.916323	0.915636	0.915725	0.915696
Specificity	0.035366	0.052964	0.044934	0.044934	0.039467	0.039809	0.039809
F_1 score	0.955744	0.956441	0.956155	0.956155	0.955801	0.955849	0.955832

gathered data. The time taken by the neural network for training on each of these feature sets was approximately the same, as the number of records weren't altered. A reason for this was the focus on evaluating our feature set against the same data, only then we can make a comparison between two feature sets or a set of different feature sets.

Feature importance

For obtaining importance of every single feature that we consider in our experiment, we evaluated the entire feature set with a bag of decision trees and gathered the error in prediction with respect to the over all performance of the classifier. From the results we obtain the following order for feature importance:

> *Maximum number of reviews* > *number of Firsts* > *average rating distribution* > *review count* > *Tips* > *Photo Count* > *Friend Count* > *Cool Votes* > *Followers* > *Useful Votes* > *Burstiness* > *Funny Votes* > *Negative review ratio* > *Positive review ratio* > *ERD* > *ETG* > *Compliments* > *Bookmarks.*

This order suggests a weight for every particular feature, the higher the weight the more important a feature is. Some of the features mentioned in the above comparison are features from the feature set used by Rayana and Akoglu [36]. Not all of the features considered by them are included in our set.

A strange discrepancy which we observed from these results is the decrease in the performance when the total vote count is added to the feature set. It is strange because upon implementing feature importance through a decision tree, the total number of votes do carry a higher importance than many other features that we have. Now, one might assume that because of that importance, it will improve the performance of the neural network upon being added into computation but the experimental results show a decrease in the F_1 score and precision.

Discussion

As we already know Supervised learning has many advantages over unsupervised learning because of the presence of labels, and that is clearly visible from our framework's performance and the performance of frameworks such as FraudEagle [1] and SpEagle [36]. But the ground truth remains the same, it is not possible to find accurate labels for all kinds of data easily from the Internet or easily from nature to conduct experiments on. The performance of neural network algorithm is exceptionally good when it comes to deriving patterns from the data, a reason being its variants used for facial recognition applications. It is quite common that a lot of features are used with neural network models, however, it is not uncommon for the neural network to work with any number of features. The main importance of the neural network lies in determining the pattern from the data, and the number of features just serve a purpose for the non-linearity of the pattern. In this paper, we took up the intuition that humans can spot fake behavior in social interaction quite well and hence an algorithm which imitates the human brain to some extent should be able to do the same.

The proposed feature set is again the unique contribution of our work. The first reason being, almost all the work in fraud detection is focused on determining which data points belong to fraud and which ones don't. We took up a different approach, one to

determine the characteristics that lead towards fraudulent behavior. The reason being, that we want to solve a much larger problem. Not only do we wish to identify fraud from various features but we want to find the various incentives, and hints the reviewer leaves when (s)he is committing fraud. One of the most important reasons that we selected the social interactive behavior feature set is the goal to identify these characteristics. It is also hoped that, once such characteristics are identified, certain measures can be taken to prevent the user from indulging in fraudulent activities.

Online review fraud is a crucial problem which can be addressed using many view points from many other applications, such as the one with convolutional neural networks. It may be a good idea to derive some relationship between the business name or specific type of the restaurant from the review text. For example, there can be some relationship between a review posted for a Sushi restaurant and the name which might be something related to Sushi, or all in all once a restaurant is categorized as such there can be a relationship of relevance of the review to the name/type of the restaurant. This relevance can again prove to be a measure in classifying reviews.

Another approach to look at this problem might be from the area of recommendation systems, with developing models that replicate the behavior of each and every user present on the system and the models' predicted values can be compared with the actual values of ratings on every single new review that a user writes, to determine the truthfulness of that particular rating. Models such as these would be able to imitate a user's personality well and it is possible to flag suspicious behavior when a user diverges too much from his/her interests and reviews a restaurant which might not be listed under the restaurants that the user may want to visit.

Various viewpoints would allow us to actually deduce the digital personalities of users on almost any social network, based on the intricate feature set each social network presents us. The same feature set cannot be obtained from a social network such as facebook or Twitter, in the same way the feature set that one might obtain from facebook or Twitter is not similar to the one obtained from Yelp.

Conclusions

Our model for opinion spam detection uses social interaction behavior features and evaluates the importance of each feature with respect to fraudulent activities. The social interaction behavior features include number of friends, number of followers, number of photos, number of compliments, and number of votes received. With the opinion spam it is possible to identify the fraudulent users on the platform, in our case Yelp. We train a supervised model using neural network and backpropagation algorithm for classifying users and reviews as fraud and genuine. We first obtain the results using user behavior features and then incrementally evaluate the performance of the model by extending the feature set with features from social interaction. We also compare the performance by training the model on only user behavioral features and only user social interaction features. The performance of the model improves when we extend the user behavioral feature set. However, if we train the model with only social interaction features, the results thus obtained are significantly closer to the results from other standards such as user behavior features. Basically, the neural network is able to derive a relationship from user

specific characteristics between the user and opinion fraud. These characteristics are captured in the social interaction features.

Our efforts in this work were focused upon the user specific perspective of fraud. The neural network algorithm certainly proved to be a beneficial choice with the learning rate we used. However, much of the efforts were put into carefully devising the feature set. It is what we identify as the novel perspective towards Opinion Spam. The machine learning algorithm too, can be improved from what we trained. There are various methods and parameters which one could apply to do so. However, the focus and scope of our research is the social interaction feature set.

We observe that for a platform such as Yelp it is not possible to gain a similar behavioral insight as one might obtain from Twitter or Facebook user information. The traits which can be determined from the data publicly available are limited because of Yelp's nature as a platform. However, it provides information on the business, review and user scale which interrelates well to provide a certain depth in fraud analysis. Another limitation one might consider is that the number of genuine reviews as compared to the number of fraudulent reviews is much greater resulting into over-fitting or misclassification. Considering a limited number of the recommended reviews might be a way to balance out the ratio. Also, there is a group of users on Yelp containing more friends and having more votes than 50% of the remaining Yelp users combined, these users might become outliers if not handled with proper normalization of the features and affect the overall genuineness of a user.

We also observe that the results suffer to some extent because of the data being biased towards recommended reviews. The number of not-recommended users is always going to be less than the number of recommended users when it comes to opinion spam, regardless of the number of businesses that we consider for our experiment. However, in an ideal scenario where we have equal number of recommended and not recommended users, or a nearly equal number, one might observe a completely different result than what we recorded. For the case study we consider, we discover a unique feature set that captures the user characteristics facilitating fraud detection to a great extent.

Abbreviation

SDN: : software defined networks.

Authors' contributions

This work is completed by KG, YP and CS. KG and YP developed this idea and KG implemented and tested the framework under her supervision. CS gave his guidelines and comments on approaching the problems faced with various experiments. As a corresponding author, YP managed this project during its tenure and advised the direction of this work while reviewing the paper. All authors read and approved the final manuscript.

Acknowledgements

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

We do not wish to disclose the data which we collected. The data is information regarding the various users on Yelp and though it is all public information, it is information specific to users and businesses. It can be categorized as sensitive information when looked upon as a collection, it was collected with the help of the Crawler we developed and was solely for the purpose of this work.

Funding

This work was supported by College Of Engineering at San Jose State University and partially supported by SJSU Research Foundation.

Publisher's Note

Received: 28 November 2016 Accepted: 4 May 2017
Published online: 15 May 2017

References

1. Akoglu L, Chandy R, Faloutsos C. Opinion fraud detection in online reviews by network effects. In: International AAAI conference on web and social media; 2013. http://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/view/5981.
2. Anderson E, Simester D. Deceptive reviews: the influential tail. Tech Rep. 2013;2:1. **(Citeseer)**.
3. Anderson ET, Simester DI. Reviews without a purchase: low ratings, loyal customers, and deception. J Mark Res. 2014;51(3):249–69. doi:10.1509/jmr.13.0209.
4. Barabsi AL, Albert R, Jeong H. Scale-free characteristics of random networks: the topology of the world-wide web. Physica A Stat Mech Appl. 2000;281:69–77. doi:10.1016/S0378-4371(00)00018-2.
5. Bebis G, Georgiopoulos M. Feed-forward neural networks. IEEE Potentials. 1994;13(4):27–31. doi:10.1109/45.329294.
6. Benczur AA, Csalogany K, Sarlos T, Uher M. Spamrank-fully automatic link spam detection, work in progress. In: AIRWeb'05. First international workshop on adversarial information retrieval on the web. Chiba, 2005. p. 1–14, http://eprints.sztaki.hu/4029/.
7. Chevalier JA, Mayzlin D. The effect of word of mouth on sales: online book reviews. J Mark Res. 2006;43(3):345–54. doi:10.1509/jmkr.43.3.345.
8. Christodoulou C, Georgiopoulos M. Applications of neural networks in electromagnetics. 1st ed. Norwood: Artech House Inc; 2000.
9. Cross S, Harrison R, Kennedy R. Introduction to neural networks. Lancet. 1995;346(8982):1075–9. doi:10.1016/S0140-6736(95)91746-2.
10. Feng S, Banerjee R, Choi Y. Syntactic stylometry for deception detection. In: Proceedings of the 50th annual meeting of the association for computational linguistics: short papers, Vol. 2. Association for computational linguistics. ACL '12: Stroudsburg; 2012. p. 171–5. http://dl.acm.org/citation.cfm?id=2390665.2390708.
11. Floyd K, Freling R, Alhoqail S, Cho HY, Freling T. How online product reviews affect retail sales: a meta-analysis. J Retail. 2014;90(2):217–32. doi:10.1016/j.jretai.2014.04.004.
12. Gardner M, Dorling S. Artificial neural networks (the multilayer perceptron)a review of applications in the atmospheric sciences. Atmos Environ. 1998;32(1415):2627–36. doi:10.1016/S1352-2310(97)00447-0.
13. Hedley J. jsoup: Java html parser. 2009–2016. https://jsoup.org/.
14. Hirose Y, Yamashita K, Hijiya S. Back-propagation algorithm which varies the number of hidden units. Neural Netw. 1991;4(1):61–6. doi:10.1016/0893-6080(91)90032-Z.
15. Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Netw. 1989;2(5):359–66. doi:10.1016/0893-6080(89)90020-8.
16. Jiang M, Cui P, Beutel A, Faloutsos C, Yang S. Catchsync: catching synchronized behavior in large directed graphs. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '14. New York: ACM; 2014. p. 941–50.
17. Jindal N, Liu B. Review spam detection. In: Proceedings of the 16th international conference on world wide web, WWW '07. New York: ACM; 2007. p. 1189–90.
18. Jindal N, Liu B. Opinion spam and analysis. In: Proceedings of the 2008 international conference on web search and data mining. New York: ACM; 2008. p. 219–30.
19. Jindal N, Liu B, Lim EP. Finding unusual review patterns using unexpected rules. In: Proceedings of the 19th ACM international conference on information and knowledge management. New York: ACM; 2010. p. 1549–52.
20. Kalogirou SA. Applications of artificial neural-networks for energy systems. Appl Energy. 2000;67(12):17–35. doi:10.1016/S0306-2619(00)00005-2.
21. Kumar R, Novak J, Tomkins A. Structure and evolution of online social networks. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '06. New York: ACM; 2006. p. 611–7.
22. Lawrence S, Giles CL, Tsoi AC, Back AD. Face recognition: a convolutional neural-network approach. IEEE Trans Neural Netw. 1997;8(1):98–113. doi:10.1109/72.554195.
23. Li F, Huang M, Yang Y, Zhu X. Learning to identify review spam. In: Proceedings of the twenty-second international joint conference on artificial intelligence, Vol 3. AAAI Press, IJCAI'11; 2011. p. 2488–93.
24. Li H, Chen Z, Liu B, Wei X, Shao J. Spotting fake reviews via collective positive-unlabeled learning. In: Proceedings of the 2014 IEEE international conference on data mining, IEEE computer society. Washington: ICDM '14; 2014. p. 899–904.
25. Lim EP, Nguyen VA, Jindal N, Liu B, Lauw HW. Detecting product review spammers using rating behaviors. In: Proceedings of the 19th ACM international conference on information and knowledge management. New York: ACM, CIKM '10; 2010. p. 939–48.
26. Luca M. Reviews, reputation, and revenue: the case of yelp. com. Com (September 16, 2011) Harvard Business School NOM Unit Working Paper (12-016). 2011.
27. Luca M, Zervas G. Fake it till you make it: reputation, competition, and yelp review fraud. Manag Sci DOI. 2016. doi:10.2139/ssrn.2293164.
28. Mayzlin D, Dover Y, Chevalier J. Promotional reviews: an empirical investigation of online review manipulation. Am Eco Rev. 2014;104(8):2421–55.

29. Mukherjee A, Liu B, Wang J, Glance N, Jindal N. Detecting group review spam. In: Proceedings of the 20th international conference companion on world wide web. New York: ACM, WWW '11; 2011. p. 93–4.
30. Mukherjee A, Liu B, Glance N. Spotting fake reviewer groups in consumer reviews. In: Proceedings of the 21st international conference on world wide web. New York: ACM, WWW '12; 2012. p. 191–200.
31. Ott M, Choi Y, Cardie C, Hancock JT. Finding deceptive opinion spam by any stretch of the imagination. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, Vol. 1. Association for computational linguistics. , HLT '11: stroudsburg; 2011. p. 309–19. http://dl.acm.org/citation.cfm?id=2002472.2002512.
32. Ott M, Cardie C, Hancock J. Estimating the prevalence of deception in online review communities. In: Proceedings of the 21st international conference on world wide web. New York: ACM, WWW '12; 2012. p. 201–10.
33. Pao YH. Adaptive pattern recognition and neural networks. Boston: Addison-Wesley Longman Publishing Co.; 1989.
34. Pomerleau DA. Efficient training of artificial neural networks for autonomous navigation. Neural Comput. 1991;3(1):88–97. doi:10.1162/neco.1991.3.1.88.
35. Pomerleau DA. Knowledge-based training of artificial neural networks for autonomous robot driving. In: Robot learning. Springer: New York; 1993. p. 19–43.
36. Rayana S, Akoglu L. Collective opinion spam detection: Bridging review networks and metadata. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. New York: ACM, KDD '15; 2015. p. 985–94.
37. Rowley HA, Baluja S, Kanade T. Neural network-based face detection. IEEE Trans Pattern Anal Mach Intell. 1998;20(1):23–38. doi:10.1109/34.655647.
38. Streitfeld D. The best book reviews money can buy. New York Times 25. 2012. http://www.nytimes.com/2012/08/26/business/book-reviewers-for-hire-meet-a-demand-for-online-raves.html. Accessed 11 Sept 2016.
39. Wang G, Xie S, Liu B, Yu PS. Review graph based online store review spammer detection. In: Proceedings of the 2011 IEEE 11th international conference on data mining, IEEE computer society. ICDM '11: Washington; 2011. p. 1242–7.
40. Widrow B, Lehr MA. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. Proc IEEE. 1990;78(9):1415–42. doi:10.1109/5.58323.
41. Xu C, Zhang J. Towards collusive fraud detection in online reviews. In: 2015 IEEE international conference on data mining; 2015. p. 1051–6.
42. Ye J, Akoglu L. Discovering opinion spammer groups by network footprints. Cham: Springer International Publishing; 2015.
43. You Y, Vadakkepatt GG, Joshi AM. A meta-analysis of electronic word-of-mouth elasticity. J Market. 2015;79(2):19–39. doi:10.1509/jm.14.0169.
44. Zhang J, Yan Y, Lades M. Face recognition: eigenface, elastic matching, and neural nets. IEEE Proc. 1997;85(9):1423–35. doi:10.1109/5.628712.