Fraud Detection in Online Reviews Using FraudEagle with Priors

Rajendra Kumar Raghupatruni
rraghupatrun@cs.stonybrook.edu

Mudit Mehrotra
mmehrotra@cs.stonybrook.edu

ABSTRACT
Online Reviews on products (services) plays an influential role on the prospective customer decision on purchasing (availing) the product (service). This decides the fate of the product (services). Due to this there is a need for an automated system to identify and curb such instances. However, it is a challenge for such machines to identify fake reviews. There are several techniques proposed and developed to identify such reviews and spammer groups. One such technique is the FraudEagle framework that exploits network effects among reviewers and products for fraud detection. In our work, we extended this technique by adding new attributes to the review network such as *helpfulness*, *verified purchase* and *duplicates in reviews* for improved opinion spam detection which are discussed in the below sections.

Keywords
Opinion Spam, Fraud, Review.

1. INTRODUCTION
The objective of fraud detection is to identify fake reviews from a pool of user reviews posted online. This is a challenging problem as spammers post reviews that appear believable but are actually untruthful. Sometimes they even post genuine reviews for camouflage. There are experienced individuals and groups who publish untruthful reviews to make money. Detecting such spam from online reviews is a difficult task. However, there are many prior works which solved this problem using techniques described in Jindal et al. (Jindal, Nitin, & Bing Liu.), exploited the review text duplicates in the reviews and (Geli et al.) used burstiness in the reviews. The method proposed in (Jindal, Nitin, & Bing Liu.) used duplicates in review as a positive training data to classify spam. The duplicates are detected using shingle method (Broder et al. 1997). As this is a supervised model based on the training sets, it cannot be extended to other domains easily. For every other domain we need to collect data based on domain characteristics and train the classifier to identify the spam. Moreover selecting those specific sets is a tedious task, as datasets from different domains possess different characteristics. Exploiting burstiness in reviews specified in (Geli et al.) is another approach to detect spam in reviews. In normal circumstances, the reviews arrive randomly. However, there are certain situations where the number of reviews for the product are concentrated during a specific time period which is identified as review burst. This burstiness in reviews are detected using Kernel Density Estimation. However, on the downside of this approach, it works only when there is a burst in reviews which is not commonly seen in online reviews.

And there are other novel approaches like (Akoglu et al. 2013), identifying the spam based on the positive or negative sentiment in reviews. This algorithm uses review network undirected graph in which the users and products are nodes and edges are the relations between users and products (i.e., a review). This forms a bipartite graph with signed edges. Sign of the edge weight is based on the positive or negative sentiment in the review ratings. This is an unsupervised method which can be adapted to any domain. However, this method uses only the review rating feature to obtain the sentiment and correlations among users and products. There are other well defined influential features such as usefulness count of a review, duplicates in review text, and verified purchase tags etc which will give accurate results, are not considered in the (Akoglu et al. 2013).

The objective of this project is to improve the existing Fraud eagle technique by incorporating new orthogonalities into the review network graph as priors. The new attributes added to the review network (esp., to the User Nodes) are helpfulness, verified purchase and duplicates in reviews. In the following sections, we described the prior work, modified framework, evaluation and conclusion.

2. Prior Work
Past research in the field of fake reviews detection has been multifaceted. Pattern mining and spam indicator heuristics have been exploited by Mukherjee et al. (Mukherjee, Liu, Glance & Jindal, 2011) to study group spamming behaviors. Candidate spamming groups are identified first followed by computation of spam indicator values. These groups are then ranked on the basis of indicators such as content similarity across the group, group size, group support, time window and whether the group posted early review in order to make a big impact. Similarly, Keystroke patterns have been used for deception detection by Banerjee et al. (Banerjee, Feng, Kang & Choi, 2014) by taking cues from editing manoeuvres and duration of pauses. Writing rate, pauses and revision rate have been measured from keystroke logs and used in detecting fake reviews. However, these approaches are largely based on and limited to the review text alone to detect the spamming behavior, which might not be working well when there is no review text available for a review.

Wang et al. (Wang, Xie and Yu, 2011) proposed use of a heterogeneous review graph to understand the interrelationship between trustiness of reviewers, honesty of reviews and the reliability of stores. Their technique uses connectivity structure of reviewer's reviews, all the stores he/she reviewed and reviews from other reviewers. On similar lines, Network effect among reviewers and products have been exploited by Akoglu et al. (Akoglu, Chandy & Faloutsos, 2013) for opinion fraud detection. The algorithm proposed above is unsupervised and scalable and is applicable to large datasets. However, this algorithm have considered only the review ratings to predict the online spam.

3. Our Contribution

In this work, we extended the Fraud Eagle Framework by Akoglu et al., which is a network based technique to detect spam in online reviews. We used the below characteristics to compute the prior probabilities of user (product) to be honest (good) or fraud (bad) in addition to the existing features.

- Duplicates in reviews
- Usefulness Count of a review
- Verified Purchase Tag

These priors are used to compute the beliefs of each node (users/products) using Loopy Belief Propagation (LBP). LBP is an iterative message passing algorithm which works well for wide variety of applications. And to compute the probabilities we used pairwise Markov Random Field (pMRF) (Kindermann and Snell 1980) which works very well for an undirected bipartite graph.

3.1 Duplicates Computation

In many online reviews, we often tend to see the same review appearing multiple times. This is one technique used by spammers to popularize/defame the product. So we exploited this feature to compute the duplicate review count using cosine similarity. We considered the reviews are duplicated when the corresponding texts have a similarity score of at least 0.9. The correlation that can be drawn from this feature is, the more duplicate count the more likely the review to be spam and the reviewer to be fraud.

3.2 Usefulness Count

The reviews from Amazon and Yelp has provides a field, usefulness which indicates that a particular review has been found useful by other potential customers who intends to purchase the product. This indicates that higher the usefulness count, it is more likely for the review to be genuine. Whenever there is no count available, we initialized the default influence scores as 0.

3.3 Verified Purchase

Amazon endorses the reviews written by the customer who purchased the product from their website by attaching a tag *verified purchase* to the reviews. This is a very good indicator of the user being genuine in writing the review and the corresponding review is highly likely to be benign.

4. Modified Framework

We used the Fraud Eagle framework as a baseline for this project, the idea is to improve the results by adding more attributes (orthogonalities) like helpfulness of the review, duplicates in the review text and verified purchased tags. Helpfulness in the review (available in the Yelp dataset) is described by three fields, Useful, Funny and Cool. Each field is associated with a count value indicating that some user(s) who wish to avail the service or wants to purchase the product found this review useful or funny or cool. Out of these we decided to utilize only the useful count for this work. As Funny and Cool fields may not actually predict the review fairness and does not give a valid intuition about the review. Duplicate reviews are detected using the cosine similarity, a shingle method. The duplicates are detected using a similarity score threshold > 0.9. Based on this, each user who wrote reviews for a product are given a score to indicate the untruthfulness of the reviewer. Verified Purchase tag is an endorsement given by Amazon (in amazon dataset) to indicate that the user has purchased the product from amazon. If the purchase is verified then the review is more likely to be genuine. With these attributes and assumptions we improved the existing framework by updating the priors based on the new features.

Outline of Modified SIA

Step 1: Scoring
signedInferenceAlgorithm()
Input: Bipartite network of users, products, review ratings, helpfulness, verified purchased, review text.
Output: Score for every user (fraud), product (bad), review (fake)
Step 2: Grouping
findGroups()
Input: ranked list of users by score from Step 1, no. of top users k
Output: bot-users and products under attack

In this framework, each node can be classified as {honest, fraud} for a user, {good, bad} for a product. It utilizes pairwise Markov Random Fields (pMRF) for this classification. pMRF is a set of random variables that satisfy the pairwise Markov property described by an undirected graph i.e. any two non adjacent variables are conditionally independent, given all other variables.

Labels:
L^U = {Honest, Dishonest} represents the domain of users
L^P = {Good, Bad} represents the domain of products.

G(V, E), a signed review network graph in which users and products form the nodes. They are connected with signed edges {+,-}.
Let $\Psi_i^U \in \psi$ be a prior mapping $\Psi_i^U : L^U \rightarrow R_{\geq 0}$, $\Psi_j^P : L^P \rightarrow R_{\geq 0}$ for each unobserved user Y_i and unobserved product Y_j
For each edge, let $\Psi_{ij}^s \in \psi$ be a compatibility mapping.
The probability of users/products belonging to each class (as per pMRF) is given by

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{Y_i \in \mathcal{Y}^V} \psi_i(y_i) \prod_{e(Y_i^U, Y_j^P, s) \in \mathcal{E}} \psi_{ij}^s(y_i, y_j)$$

The probability assigned to the unobserved variables might not be their best assignment. The best assignment is determined by **Loopy Belief Propagation** (LBP) extended to work on a signed network. LBP is a message passing algorithm for inferring classification in a graphical model. In this, each user Y_i and product Y_j sends a message to its neighbour indicating how it thinks the neighbour should be classified. Message passing continues until all the messages stabilize (no change). The objective of the problem is to maximize P(y/x), defined in the above equation.

$$m_{i\to j}(y_j) = \alpha_1 \sum_{y_i \in \mathcal{L}^{\mathcal{U}}} \psi^{s}_{ij}(y_i, y_j)\, \psi^{\mathcal{U}}_{i}(y_i) \prod_{Y_k \in \mathcal{N}_i \cap \mathcal{Y}^{\mathcal{P}} \setminus Y_j} m_{k\to i}(y_i), \quad \forall y_j \in \mathcal{L}^{\mathcal{P}}$$

$$b_i(y_i) = \alpha_2\, \psi^{\mathcal{U}}_{i}(y_i) \prod_{Y_j \in \mathcal{N}_i \cap \mathcal{Y}^{\mathcal{P}}} m_{j\to i}(y_i), \forall y_i \in \mathcal{L}^{\mathcal{U}}$$

Where $m_{i\to j}$ is a message sent by user Y_i to product Y_j, α's are the normalization constants. L^U denotes the label domain for users and L^P denotes the label domain for products. b_i (yi) denotes the belief of assigning Y_i with label y_i

4.1 Priors Computation

Priors are the initial beliefs of the objects in the graph. These priors are used to instantiate the clique potential functions of the framework. Priors are computed using the three new attributes, Useful Count (UC), Duplicate Count (DC) and Verified Purchase Tag (VPT) as follows:

While computing the priors Verified Purchase Tag is considered the most significant attribute to validate the benignity of the review. Therefore, when a review has VPT, we assigned highest probability for it to be genuine. With DC, we are not very sure about the authenticity. However, we have assigned a threshold T_{DC} based on the dataset analysis to give the reviews whose DC > T_{DC} to have higher probability to be fraud. Similarly, with UC the probabilities are based on the threshold T_{UC}. The below table gives a Prior probabilities for each of the attribute when considered individually or in conjunction with other attributes based on their availability.

Feature	Probabilities	
	Good/Honest/Real	Bad/Fraud/Fake
VPT	0.99	0.01
DC	0.1	0.9
UC	0.6	0.4
VPT & DC or VPT & UC	0.99	0.01
DC & UC	0.5	0.5
ALL	0.99	0.01
Others	0.5	0.5

Table 1: Prior Probabilities based on VPT, DC and UC

5. Dataset

The dataset of both recommended and non-recommended reviews is obtained from Yelp.com. We used an automatic crawler to get the details (like Restaurant Name, Author, Review, Rating, Votes for Useful/Cool/Funny) of the reviews about the restaurants. We have also collected the academic dataset available from the website: https://www.yelp.com/academic_dataset

We also manually crawled the data from Amazon to obtain the reviews which have "Verified Purchased" tags.

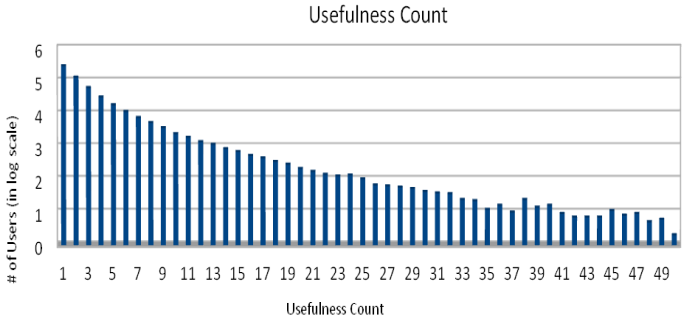

Figure 1

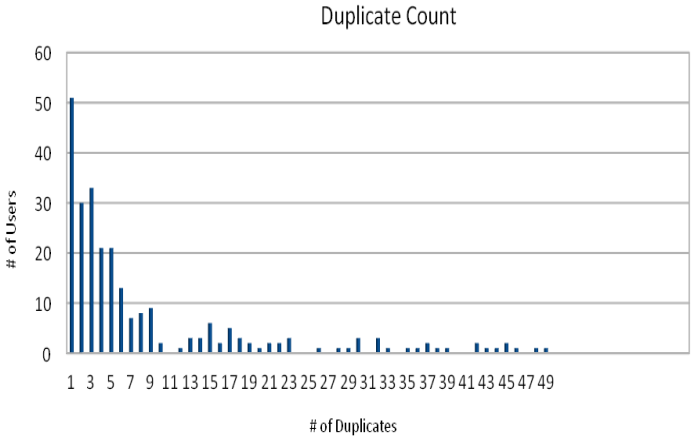

Figure 2

6. Results

We have tested the modified SIA with a synthetic dataset as well as with the real data obtained from Yelp and Amazon.

Version 1: From figure 1, we observed that the dataset had several reviews that were marked useful by the users. We postulated that if a review has been marked useful by a large number of users, the reviewer is less likely to be a fake user, though this is not a very strong indicator of a user not being fraud. We accommodate this by considering that there is 60% chance (slightly over the base case – 50%) that a user who posts very useful reviews is honest.

Inference: Our modified sIA outperformed both sIA and HITS by not reporting users who posted useful reviews amongst top fraudsters. Figure 3 confirms that users who posted very useful reviews – Corey, Amanda, David and Nikole are not reported as fraudsters by modified sIA.

Version 2: From figure 2, we observed that the dataset had several **duplicate reviews**. Current sIA and version 1 of modified sIA did not consider duplicate reviews to be an indicator of user being fraud and did not classify such users as fraudsters. We claimed that a duplicate review is a strong indicator of a user being a fraudster. Hence, if a user posts a duplicate review, there is 90% chance that the user is fraud.

Inference: Figure 4 confirmed that our modified sIA outperformed both sIA and HITS by reporting users who posted duplicate reviews amongst top 10 fraudsters. These users are **Heather, Mark, Tanya, Walker and Raphael.**

Version 3: From Amazon dataset, we identified several reviews to be marked verified purchase. Every though a user whose purchase has been verified may have posted a negative review for a good product, we claim that there is very less chance of such a user being fraudster. Both sIA, ver.1 and 2 of modified sIA may incorrectly identify such user as a fraudster. We postulated that if a review is marked ***verified purchase,*** there is 99% chance that the user is honest.

Inference: Figure 5 confirmed that our modified sIA outperformed both sIA and HITS by reporting users whose purchase was verified as honest. These users include **Charlie.**

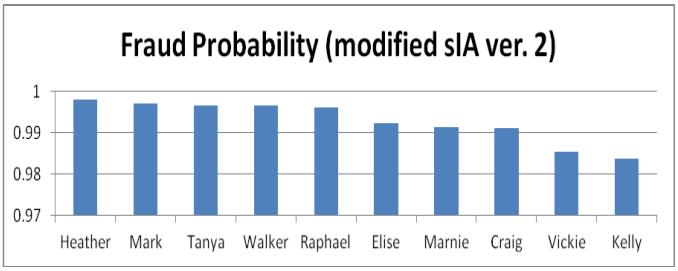

Figure 4: modified sIA ver. 2

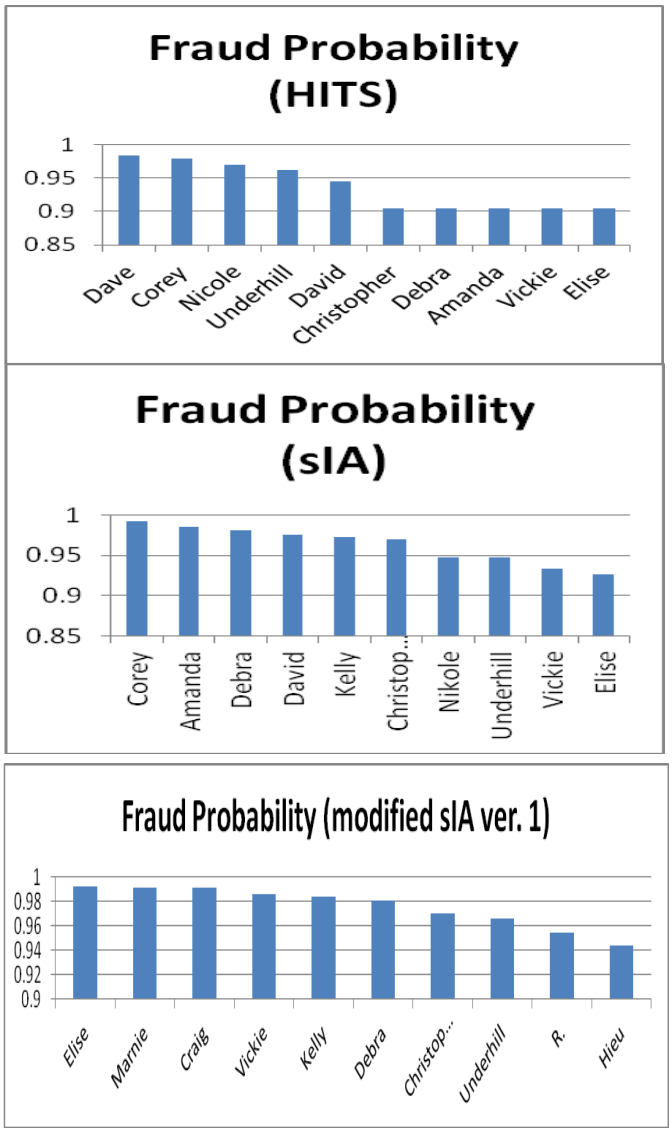

Figure 3: modified sIA ver 1

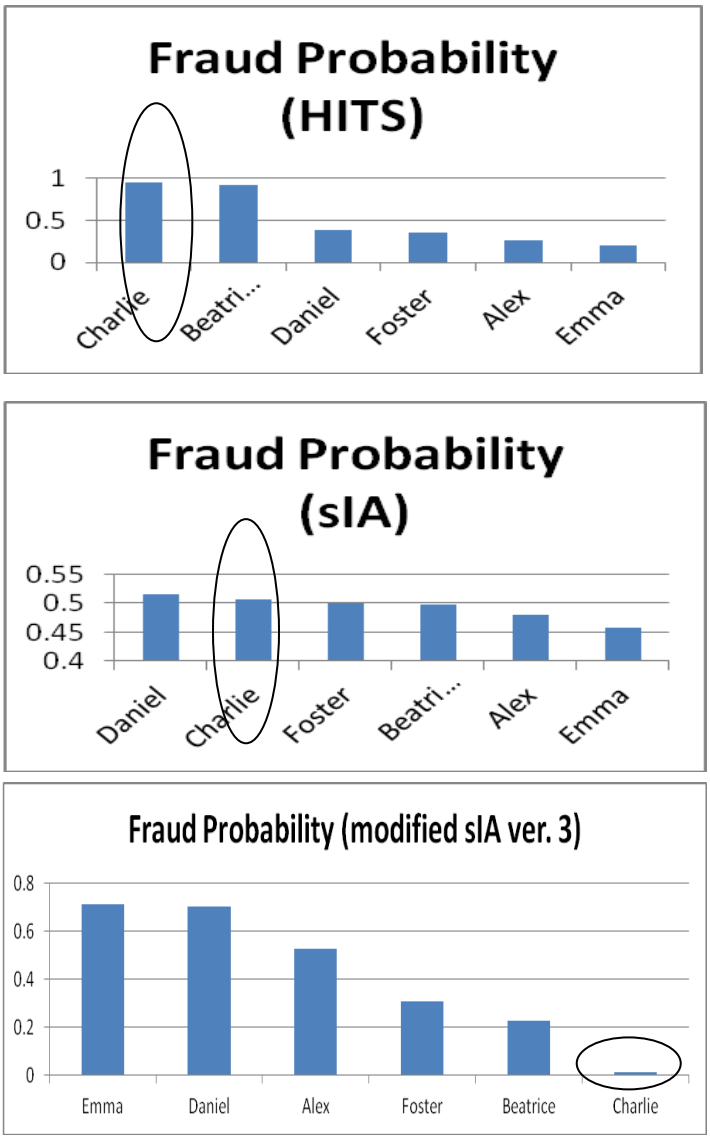

Figure5:Modified sIA version 3

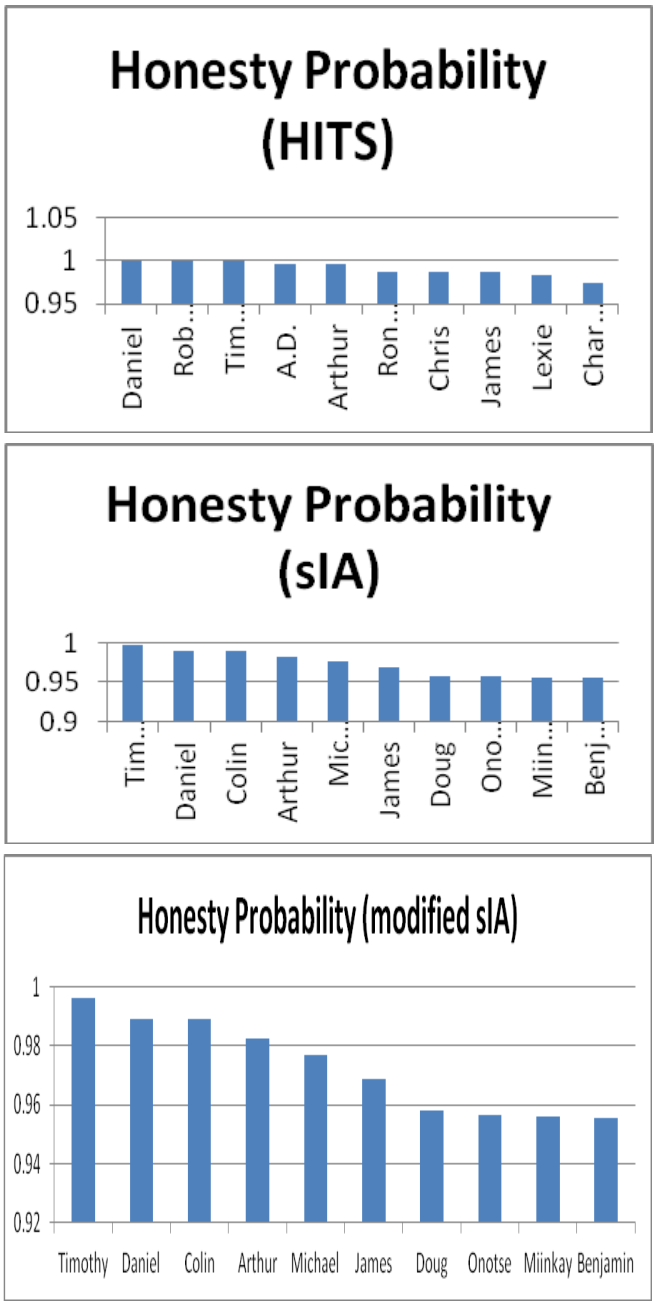

Figure 6: Honesty probability

Accuracy = (18/25) (classification)

7. Conclusion

In this project, we extended Signed Inference Algorithm (SIA) from (Akoglu et al. 2013), a graph based online spam detection algorithm by introducing more orthogonalities such as helpfulness of the review, verified purchase tag etc. These orthogonalities are plugged in as priors to obtain the fake probability of users (products) being fraud (bad). We used a variant of HITS and native SIA to validate the results of this improved algorithm. Our experimental results using Yelp and Amazon datasets are promising. We have also validated our results using a Support Vector Machine (SVM) classifier. We used the Yelp Recommended and Non-Recommended reviews list to train the classifier with non-recommended reviews as positive training data (fake reviews). We tested this trained model with the top 25 users/reviews obtained from our modified algorithm. The results obtained are in agreement with the supervised model. This modified algorithm has produced improved results over its predecessor and also over other similar techniques (HITS).

8. REFERENCES

Mukherjee, A., Liu, B., Wang, J., Glance, N. and Jindal, N. 2011. Detecting group review spam. In *WWW*.

Akoglu, L., Chandy, R. and Faloutsos, Christos. 2013. Opinion fraud detection in online reviews by network effects. In *ICWSM*

Wang, G., Xie, S., Liu, B. and Yu, P. S. 2011. Review graph based online store review spammer detection. In *ICDM, 1242–1247.*

Geli Fei, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Exploiting Burstiness in Reviews for Review Spammer Detection. In *ICWSM. 2013*.

Jindal, Nitin, and Bing Liu. "Review spam detection." Proceedings of the 16th international conference on World Wide Web. *ACM, 2007*.

Broder, A. Z. On the resemblance and containment of documents. In Proceedings of Compression and Complexity of Sequences 1997, IEEE Computer Society, 1997.

Columns on Last Page Should Be Made As Close As Possible to Equal Length