Detecting Product Review Spammers using Rating Behaviors

Itay Dressler
• What is Spam?
• Why should you care?
• How to detect Spam?
What is Spam?
What is Spam?

- All forms of malicious manipulation of user generated data so as to influence usage patterns of the data.

- Examples of web spam include search engine spam (SEO), email spam, and Opinions spam (talk-backs).
Search Spam

Keyword stuffing
<table>
<thead>
<tr>
<th>Sender</th>
<th>Subject</th>
<th>Date</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>eBay</td>
<td>andy_kishore_in1, deals on mobiles, digital cameras and more, this November on e...</td>
<td>Wed 11/09</td>
<td>20k</td>
</tr>
<tr>
<td>Winter Desktop</td>
<td>3D Snowy Cottage ScreenSaver- Get into the season</td>
<td>Tue 11/08</td>
<td>6k</td>
</tr>
<tr>
<td>Dept. Store</td>
<td>Claim your $500 Dept. Store Gift Card</td>
<td>Tue 11/08</td>
<td>8k</td>
</tr>
<tr>
<td>Apple® iPod® nano Alert</td>
<td>FREE* Apple® iPod® nano Offer</td>
<td>Tue 11/08</td>
<td>6k</td>
</tr>
<tr>
<td>Tsunoni Health</td>
<td>Oprah and Terrell Owens - New Waves in Healthy Living</td>
<td>Tue 11/08</td>
<td>6k</td>
</tr>
<tr>
<td>Fast Food Survey</td>
<td>Wendy’s or Jack in the Box? Vote and receive $100</td>
<td>Tue 11/08</td>
<td>6k</td>
</tr>
<tr>
<td>Houseware Gift Card</td>
<td>A $500 BedBath&amp;Beyond Gift Card is waiting for you</td>
<td>Tue 11/08</td>
<td>8k</td>
</tr>
<tr>
<td>Christian Debt Advisor</td>
<td>Are you drowning in debt?</td>
<td>Tue 11/08</td>
<td>5k</td>
</tr>
<tr>
<td>Fast Food Survey</td>
<td>Tell us your favorite burger and get $100</td>
<td>Tue 11/08</td>
<td>6k</td>
</tr>
<tr>
<td>Entry Confirmation</td>
<td>Please confirm your iWon entry</td>
<td>Tue 11/08</td>
<td>5k</td>
</tr>
<tr>
<td>Autos.com</td>
<td>You will be shocked these prices are incredibly low</td>
<td>Tue 11/08</td>
<td>7k</td>
</tr>
<tr>
<td>Business Opportunity</td>
<td>Home Workers Needed</td>
<td>Tue 11/08</td>
<td>6k</td>
</tr>
<tr>
<td>Christian Dating</td>
<td>Meet Real Christian Singles</td>
<td>Tue 11/08</td>
<td>7k</td>
</tr>
<tr>
<td>eBay Inc</td>
<td>eBay: please confirm your data</td>
<td>Tue 11/08</td>
<td>20k</td>
</tr>
<tr>
<td>LapRunner Weather</td>
<td>Free Weather Toolbar - One click access to Weather Forecast.</td>
<td>Mon 11/07</td>
<td>8k</td>
</tr>
<tr>
<td>MomExec</td>
<td>This is the perfect job for your family</td>
<td>Mon 11/07</td>
<td>7k</td>
</tr>
<tr>
<td>Convenient Laptop</td>
<td>Hewlett Packard Laptop - F.R.E.E. Offer</td>
<td>Mon 11/07</td>
<td>8k</td>
</tr>
<tr>
<td>Sample Promotions Group</td>
<td>This New RAZR Phone FREE*</td>
<td>Mon 11/07</td>
<td>9k</td>
</tr>
</tbody>
</table>

Mail Spam

(before mail spam detection)
Our Focus

Spam found in online product review sites

Review Spam
Opinion Spam
• Review spam is designed to give unfair view of some products so as to influence the consumer’s perception of the products by directly or indirectly inflating or damaging the product’s reputation.

• Under rating / Over rating.

• Unfair treatment of products.
Review Spam Example

"quoted me a price that was literally 3 times that of 3 other companies. shop around!"

Flatrate Moving™
555 W 25th St, 3rd Floor, New York, NY
(212) 425-5200 · flatrate.com
★★★★★ 794 reviews
piece of furniture · storage facilities · experiences · accommodating · flawless
"this company quoted me a price that was literally 3 times that of 3 other companies. shop around!" - citysearch.com

NYC Moving Manhattan Movers
278 Broadway, New York, NY
(646) 808-3769 · nycmanhattanmovers.webs.com
★★★★★ 145 reviews
licensed and insured · personal care · communication · best price · great price
"NYC Moving Manhattan Movers quoted me a price that was literally 3 times that of 3 other companies. shop around!"

Budget Van Lines Inc
244 5th Avenue, New York, NY
(212) 380-1812 · budgetvanlines.com
★★★★★ 68 reviews
broker · free moving quotes · carrier · fees · committed
"Budget Van Lines quoted me a price that was literally 3 times that of 3 other companies. shop around!"
Review Spam Examples

Ruhel Ahmed  October 23, 2013
★★★★★

Thanks Dear Mr Ahmed, Please post the following comment on the new BlackBerry Messenger Android APP. 'Thank you so much BlackBerry team. I was waiting this app. Its really great user friendly and smooth.' BB Team.

Asif Malick  October 22, 2013
★★★★★
Review
Spamming is a Profession today
Why should you care?

- Customers rely on reviews today more than ever.
- In general, every decision we make is heavily depended on reviews.
What is Amazon?

- Largest Internet-based company in the United States.
- Revenue totaled $61 billion in 2012.
- 244 Million Users.
- Books, Kindle and FirePhone.
What is Amazon?

• Amazon's warehouses have more square footage than 700 Madison Square Gardens and could hold more water than 10,000 Olympic Pools.
Example

• “Mr Unhappy”

• Lack of seriousness (Identical reviews).

• Very different from other reviews (96 in total).
Extremely Hard to Detect

- Spam reviews usually look perfectly normal until one compares them with other reviews.

- Tedious and Non-trivial task.

- Amazon allows users to vote reviews (Spam-able).
Review Spammer Detection

- Detecting Spammers vs Detecting spammed reviews (using Spammer classified behavior).
- The amount of evidence is limited (One review & one Product).
- Scaleable approach - incorporate new spamming behaviors as they emerge.
- Each model assigns a numeric spamming behavior score to each reviewer.
Spamming Behavior Models

- (TP) - Targeting Product.
- (TG) - Targeting Group (Brand).
- (GD) General Rating Deviation.
- (ED) Early Rating Deviation.

- Overall numeric spam score to each user (linear weighted combination).
- Avoiding deep natural text understanding (High computational costs).
Related Work

• Opinion and Sentiment Mining.
  • Extracting and aggregating positive and negative opinions (Text Mining).
  • Do not address spam detection unless being used to derive other more relevant features.

• Item Spam Detection.
  • Singleton reviewers - users who contribute only one review each..

• Helpful Review Prediction (Votes).
  • Not all unhelpful reviews are spam.
Amazon Dataset

- Product.
  - Brand.
  - Attributes (Book product has ‘author, ‘publisher and ‘price).
  - One-to-Many reviews.
- User - can contribute one or multiple reviews.
- Review.
  - Textual comment.
  - Numeric rating (normalized to [0 1].
  - Helpfulness anonymous votes from users.
Preprocessing of Amazon’s Dataset ("MProducts")

- Removal of anonymous users (Each anonymous user id may be used by multiple persons).
- Removing duplicated products (Some products have minor variations from others - e.g. color).
  - One product is chosen randomly, and all reviews are attached to it.
- Removal of inactive users and unpopular products.
  - The threshold is 3 reviews per product, and 3 reviews per user.
- Resolution of brand name synonyms - done manually (only few hundred brand names in DB).
  - HP is the same brand as Hewlett Packard.

### Table 1: Dataset Statistics

<table>
<thead>
<tr>
<th>Set Type</th>
<th>Number (before preprocessing)</th>
</tr>
</thead>
<tbody>
<tr>
<td>U: set of users</td>
<td>11,038 (313,120)</td>
</tr>
<tr>
<td>O: set of objects</td>
<td>5,693 (32,075)</td>
</tr>
<tr>
<td>V: set of reviews</td>
<td>48,894 (404,637)</td>
</tr>
<tr>
<td>E: set of ratings</td>
<td>as above</td>
</tr>
</tbody>
</table>
Notations -

- **U = \{u_i\}**: set of users
- **O = \{o_j\}**: set of objects\(^2\). Let **B = \{b_k\}** denote the set of brands and \(b(o_j) \in \textbf{B}\) denote the brand of object \(o_j\).
- **V = \{v_k\}**: set of reviews \(v_k\)'s. The review \(v_k\) comes with a rating \(e_k\).
- **E = \{e_k\}**: set of ratings \(e_k\)'s (\(e_k \in [0, 1]\)).
- \(u(v_k) = u(e_k)\): user of review \(v_k\) and rating \(e_k\).
- \(o(v_k) = o(e_k)\): object of review \(v_k\) and rating \(e_k\).
Notations -

- \( r(v_k) = r(e_k) \): order position (\( \in \{1, \cdots \} \)) of review \( v_k \) and rating \( e_k \) with respect to the reviewed object.
- \( V_{ij} = \{v_k | u(v_k) = u_i \land o(v_k) = o_j \} \): set of reviews from user \( u_i \) to object \( o_j \).
- \( E_{ij} = \{e_k | u(e_k) = u_i \land o(e_k) = o_j \} \): set of ratings from user \( u_i \) to object \( o_j \).
- \( E_{i*} = \bigcup_j E_{ij} \): set of all ratings by user \( u_i \)
- \( E_{*j} = \bigcup_i E_{ij} \): set of all ratings on object \( o_j \)
Target Based Spamming

- A spammer will direct most of his efforts to promote or victimize a few products or product lines.
- Targeted products.
- Targeted product groups.
Targeting Products

- Easily observed by the number of reviews (also ratings) on the product (As seen in the previous table).

- In MProducts, 2874 reviewer-product pairs involve multiple reviews/ratings (small number comparing to #Reviews $\sim 50k$).

- Most of these pairs, 1220 of them, involve only ratings of 5 compared with 624 of them involving only ratings of 1 or 2.
Targeting Products

- Rating Spamming - Reviewers who are involved in reviewer-product pairs with a larger number of ratings are likely to be spammers (especially when the ratings are similar).

- UserRatingSpam Score = 

  \[ c_{p,e}(u_i) = \frac{s_i}{\text{Max}_{u_i' \in U} s_{i'}} \]

- Based on the spam score function, reviewers with large proportions of ratings involved as multiple similar ratings on products are to be assigned high spam scores.

\[ s_i = \sum_{e_{ij} \in E_{ij}, |E_{ij}| > 1} |E_{ij}| \cdot \text{sim}(E_{ij}) \]

\[ \text{sim}(E_{ij}) = 1 - \frac{\text{Avg}}{e_k, e_{k'} \in E_{ij}, k < k'} |e_k - e_{k'}| \]
Review Text Spamming - similar to rating spamming.

Such review texts are likely to be identical or look similar so as to conserve spamming efforts, but we need to distinguish them from genuine text reviews.

Similarity of text reviews -

$\text{cosine}(v_k, v_{k'})$ is the cosine similarity of the bi-gram TFIDF vectors of $v_k$ and $v_{k'}$ (Frequency Inverse Document Frequency).

Spam Score -

$$c_{p,v}(u_i) = \frac{s'_i}{\max_{u'_i \in U} s'_{i'}}$$

$$s'_{i} = \sum_{v_{ij} \in V_{ij}, |V_{ij}| > 1} |V_{ij}| \cdot \text{sim}(V_{ij})$$
Targeting Products

- Combined Spam Score -

\[ c_p(u_i) = \frac{1}{2} (c_{p,e}(u_i) + c_{p,v}(u_i)) \]
Targeting Product Groups

• The pattern of spammers manipulating ratings of a set of products sharing some common attribute(s) within a short span of time. (saves the spammer from re-login).

• Ratings given to these target group of products are either very high or low - so we will device them to 2 different scores:
  
  • Single Product Group Multiple High Ratings.

  • Single Product Group Multiple Low Ratings.
Targeting Product Groups

Single Product Group Multiple High Ratings.

- We divide the whole time period into small disjoint time windows of fixed-size and derive clusters of very high ratings.

- The high rating cluster by user $u_i$ to a product group $b_k$ is:

  $$E_{ik}^H(w) = \{e_{ij} \in E_{i*} | o_j \in b_k \land t(e_{ij}) \in w \land e_{ij} \in H\text{RatingSet}\}$$

  $H\text{RatingSet} = \{1\}$

- Only large groups are assumed to be spam (larger than min=3), and were saved in $C_i^H$.

- “$w$” was empirically chosen to be one day interval.

- The product attribute used for MProducts datasets is ‘brand.’
Targeting Product Groups

Single Product Group Multiple High Ratings.

- The spam score \( u \) based on single product group multiple high ratings behavior is thus defined by:

\[
c_{g,H}(u_i) = \frac{C_i^H}{\max_{u_i' \in u} C_i'^H}
\]

\[
C_i^H = \bigcup_{k,w} \{ E_{ik}(w) \mid |E_{ik}(w)| \geq \text{minsize}^H \}
\]
Targeting Product Groups

Single Product Group Multiple Low Ratings

• The motive here is to create a negative perception of the affected products so as to reduce their sales.

\[ E^C_{ik}(w) = \{ e_{ij} \in E_i \mid o_j \in b_k \land (e_{ij}) \in w \land e_{ij} \in LRatingSet \} \]

\[ LRatingSet = \{0, 0.25\} \]

\[ C^C_{i} = \bigcup_{k,w} \{ E^C_{ik}(w) \mid |E^C_{ik}(w)| \geq \text{minsize}^C \} \]

• Mini-size here is 2, due the lower number of or ratings the database.

• Spam score -

\[ c_{g,C}(u_i) = \frac{C^C_{i}}{\text{Max}_{u'_i \in u} C^C_{i'}} \]
Targeting Product Groups

- Combined Spam Score =

\[ c_g(u_i) = \frac{1}{2} (c_g, H(u_i) + c_g, L(u_i)) \]
DEVIAITION-BASED SPAMMING

- General Deviation
- Early Deviation.
A reasonable rater is expected to give ratings similar to other raters of the same product. As spammers attempt to promote or demote products, their ratings could be quite different from other raters.

Deviation of a rating $e_{ij} :=$ difference from the average rating on the same product

$$d_{ij} = e_{ij} - \text{Avg}_{e \in E_{*j}} e$$
DEVIATION-BASED SPAMMING

Early Deviation

- Early deviation captures the behavior of a spammer contributing a review spam soon after product is made available for review.

- Other reviewers are highly influenced from these early reviews - which affects the products highly.

- The early deviation model thus relies on two pieces of information:
  - General Deviation.
  - Weight of each rating indicating how early the rating was given (alpha is a premature greater than one to accelerate decay).

- The final Deviation spam score is -

\[
w_{ij} = \frac{1}{(r_{ij})^\alpha}
\]

\[
c_e(u_i) = \frac{\sum_{e_{ij} \in E_{i*}} (|d_{ij}| \times w_{ij})}{\sum_{e_{ij} \in E_{i*}} w_{ij}}
\]
User Evaluation
The objective is to evaluate the performance of different solution methods, which are based on the declared Spammer Scores:

- Single product multiple reviews behavior (TP).
- Single product group multiple reviews behavior (TG).
- General deviation (GD) behavior.
- Early deviation (ED) behavior with $\alpha=1.5$.
- Newly introduced empirical combined method - (ALL)
  \[ c(u_i) = \frac{1}{2} c_p(u_i) + \frac{1}{4} c_g(u_i) + \frac{1}{8} c_d(u_i) + \frac{1}{8} c_e(u_i) \]
- Newly introduced Baseline method - ranks the reviewers by their unhelpfulness score.
User Evaluation

• The methods will be compared to the results of real human testers, but there are several challenges in conducting the user evaluation experiments:

  • Too many reviewers. Each reviewer can have up to 349 reviews (in MProducts).

  • The existing amazon website is not designed for review spammer detection (designed for real users).

  • The need to train human evaluators.

• These issues were handled by using a smaller subset of the database (reviewers who were highly suspected as spammers by the previous methods, and random reviewers), developing a special software for human testers (review spammer evaluation software).
User Evaluation

Review spammer evaluation software

• Ensures that the human evaluators can easily browse the reviewer profiles and their reviews (both selected and non-selected).

• The software makes the human evaluators go through all of the reviews of the reviewer before determining their judgement about him (10 reviews max per reviewers in this experiment).

• Reduces the amount of evaluation efforts and time.

• Features:
  • Easy visualization of reviews with exact and near-duplicates.
  • Review ratings among recent ratings on the same products.
  • Multiple reviews on same products.
  • Multiple reviews on the same product groups.
User Evaluation

Review spammer evaluation software
User Evaluation

Experiment Setup

- For each spammer detection method, we select 10 top ranked reviewers and 10 bottom ranked reviewers.

- Merge all reviewers into a pool.

- Sort reviewers by their combined spamming behavior scores.

- Select 25 top ranked reviewers and 25 bottom ranked reviewers.

- Randomize the order of the selected reviewers, and hand them over for user evaluation.
User Evaluation

Experiment Setup

• For each reviewer select 10 of his reviews to be highlighted for the human tester, according to:

  • Reviews that are exact (or near exacts) duplicates of other reviews, or reviews that have exact (or near exact) ratings with other reviews from the same user on the same product (TP).

  • Reviews of some products in product groups with multiple identical high or low ratings from the evaluated reviewer within the same day (TG).

  • Reviews by the evaluated reviewer having ratings deviated from the average ratings of the reviewed products (GD).

  • Reviews by the evaluated reviewer having ratings deviated from the average ratings of the reviewed products and are the early reviews of the reviewed products (ED).

• For reviewers that have less than 10 reviews, the shortage is covered by selecting random reviews from other reviewers.
User Evaluation

Experiment Setup

• Three college students were chosen as human testers. The students are familiar with Amazon’s website, and with reading product reviews.

• For each evaluator the software records -
  
• Judgment decision - “Spammer” or “Not-Spammer” for each review.

• The reviews that were actually viewed by the evaluator.

• The evaluators are not informed about the number of spammers to be detected.
User Evaluation

Experiment Setup

• Based on the human results, calculate NDCG (Normalized Discounted Cumulative Gain).

• NDCG is simply DCG normalized by the DCG of the ideal rank order of the items that has spammers agreed by all 3 evaluators ranked before those spammers agreed by 2 evaluators who are in turn ranked before the remaining reviewers.

\[
DCG = \sum_{p=1}^{50} \frac{2^{f(i_p)} - 1}{\log_2(1 + p)}
\]

\[
(f(i_p) \in [0, 3]).
\]

\[
NDCG = \frac{DCG}{DCG \text{ of ideal ranked list}}
\]
User Evaluation

Results

• Inter Evaluator Consistency.

• All the three evaluators agreed on 16 spammers and 18 non-spammers, constituting 78% of 50 evaluated reviewers.

• Given the results of the three evaluators, we assign a final label to each reviewer using majority voting, which ended up labeling 24 reviewers as spammers, and 26 as non-spammers.

Table 2: Evaluation Results

<table>
<thead>
<tr>
<th>Evaluator 1</th>
<th>Evaluator 2</th>
<th>Evaluator 3</th>
</tr>
</thead>
<tbody>
<tr>
<td># Spammers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluator 1</td>
<td>26</td>
<td>18</td>
</tr>
<tr>
<td>Evaluator 2</td>
<td>-</td>
<td>23</td>
</tr>
<tr>
<td>Evaluator 3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td># Non-Spammers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluator 1</td>
<td>24</td>
<td>19</td>
</tr>
<tr>
<td>Evaluator 2</td>
<td>-</td>
<td>27</td>
</tr>
<tr>
<td>Evaluator 3</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
User Evaluation

Results

• Those human evaluated results are being compared with the top ten and bottom 10 results of the previous methods.

• The results show that the ALL and TP methods correctly rank the spammers and non-spammers at the top and bottom ranks respectively.

Table 3: Results of Top 10 and Bottom 10 Ranked Reviewers

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>TP</th>
<th>TG</th>
<th>GD</th>
<th>ED</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td># spammers in top 10</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>6</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td># non-spammers</td>
<td>7</td>
<td>10</td>
<td>9</td>
<td>6</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>
User Evaluation

Results

• Next, we examine the NDCG for different top k positions (k= 1 to 50) in the rank list produced by each method.

• The experimental results show that the method is very effective comparing to the human evaluated results, and by comparing the NDGC results of the methods comparing to Baseline.

• The Baseline method (user helpfulness votes) is finally discovered as not such a good indicator of spam.
SUPERVISED SPAMMER DETECTION AND ANALYSIS OF SPAMMED OBJECTS
SUPERVISED SPAMMER DETECTION AND ANALYSIS OF SPAMMED OBJECTS

Regression Model for Spammers

• With the labeled spammers, we now train a linear regression model to predict the number of spam votes of a given reviewer’s spamming behaviors.

• Used to ensure that the trained regression model will have as little prediction errors as possible at the highly ranked reviewers.

• The regression model learnt by minimizing mean square error has these $w$ values: $w_0 = 0.37$, $w_1 = -0.42$, $w_2 = 1.23$, $w_3 = 2.86$, $w_4 = 4.2$.

$$\text{Number of spam votes}(u_i) = w_0 + w_1 \cdot c_d(u_i) + w_2 \cdot c_e(u_i) + w_3 \cdot c_p(u_i) + w_4 \cdot c_g(u_i)$$

• The only surprise is that general deviation GD is now given a negative weight, This suggests that having larger general deviation does not make a reviewer looks more like a spammer, although early deviation does.
SUPERVISED SPAMMER DETECTION AND ANALYSIS OF SPAMMED OBJECTS

Regression Model for Spammers

• We then apply the regression model on all 11,038 reviewers assigning each of them a new spam score (normalized by the maximum score value and denote it by $s(u)$).

• Most reviewers have relatively small spam scores.

• There are 513 (4.65%) reviewers having spam scores greater than or equal to the upper-whisker of the distribution (0.23).
SUPERVISED SPAMMER DETECTION AND ANALYSIS OF SPAMMED OBJECTS

Analysis of Spammed Products and Product Groups

- To show how the products and product groups are affected by spammers, we define a spam Index for a product $o_i$ and a product group $g_i$ as:

$$s(o_j) = \text{Avg}_{u_i \in s(o_j)} s(u_i)$$
$$s(g_l) = \text{Avg}_{u_i \in s(g_l)} s(u_i)$$

- Unhelpfulness score: Average of unhelpfulness scores of reviewers of the product or product groups.

- Random Index: Average of random score from 0 to 1 assigned to reviewers of the product or product groups.
One way to study the impact of spammers is to compare the average ratings of a product or a product group when spammers are included versus when they are excluded.

The figure shows how the average rating of a product changes after removing the top 4.65% users with highest spam scores.

Figure 6: Products after removing high spam score reviewers.
• Same evaluation for brands.

• To see why the changes in ratings are more significant at higher percentiles, we plot the average proportion of reviewers removed from the products (Figure 6b) and product brands (Figure 7b) as a result of removing the top spammers. Both figures show that most of the reviewers removed by spam scores and unhelpful ratio index belong to the highly ranked products and brands. This explains more rating changes for these products and brands.

Figure 7: Brands after removing high spam score reviewers
Conclusions

- The proposed algorithm of this paper is based on specific spam behavior scores.

- The method was tested on the Amazon database, and compared to human tester results.

- They found that their proposed method generally outperform the baseline method based on helpfulness votes.
Conclusions

• They learned a regression model, and applied it on the full database to score reviewers.

• It is shown that by removing reviewers with high spam scores, the higher ranked spammed products will experience more significant changes in rating comparing to removal of users by unhelpfulness votes, or by random users.