TrueView: Harnessing the Power of Multiple Review Sites

Amanda J. Minnich
aminnich@cs.unm.edu

Nikan Chavoshi
chavoshi@cs.unm.edu

Abdullah Mueen
mueen@cs.unm.edu

Shuang Luan
sluan@cs.unm.edu

Michalis Faloutsos
michalis@cs.unm.edu

ABSTRACT

Online reviews on products and services can be very useful for customers, but they need to be protected from manipulation. So far, most studies have focused on analyzing online reviews from a single hosting site. How could one leverage information from multiple review hosting sites? This is the key question in our work. In response, we develop a systematic methodology to merge, compare, and evaluate reviews from multiple hosting sites. We focus on hotel reviews and use more than 15 million reviews from more than 3.5 million users spanning three prominent travel sites. Our work consists of three thrusts: (a) we develop novel features capable of identifying cross-site discrepancies effectively, (b) we conduct arguably the first extensive study of cross-site variations using real data, and develop a hotel identity-matching method with 95% accuracy, (c) we introduce the TrueView score, as a proof of concept that cross-site analysis can better inform the end user. Our results show that: (1) we detect 7 times more suspicious hotels by using multiple sites compared to using the three sites in isolation, and (2) we find that 20% of all hotels appearing in all three sites seem to have low trustworthiness score.

Our work is an early effort that explores the advantages and the challenges in using multiple reviewing sites towards more informed decision making.

Categories and Subject Descriptors: H.3.3 [Information Search & Retrieval]: Relevance feedback; Selection process; Search process; Information filtering.

Keywords: Opinion Spam; Multi-Site Features; Review Mining; Anomaly Detection; Hotels; Trust

1. INTRODUCTION

How do you use online reviews before booking a hotel, especially if the hotel appears on multiple review sites? This is the overarching question that motivates this work. On the one hand, online reviews provide an unprecedented mechanism for customers to inform potential customers. On the other hand, reviews can suffer from: (a) relevance, as the needs and expectations of customers varies, and (b) fraudulent behavior, from “biased” users or the product providers themselves. As a result, it is not easy for customers to interpret and use such reviews. One often relies either on the overall rating or laboriously goes through many reviews to identify relevance and trustworthiness.

How can we leverage reviews from multiple review sites to make more informed decisions and detect fraud? The use of multiple review sites is the key differentiating factor for our work. Currently, the existence of multiple review sites can add to the confusion of the user. Each of these sites may give a different view of a hotel, and it can be difficult for the consumer to know whom to trust. We focus on reviews of hotels for reasons we discuss below, and we frame the problem as follows: We are given several review sites that review a set of partially overlapping hotels. The reviews typically include a score, user comments, time of review and possibly a user-id. The required output is a trustworthiness score for the reviews of a hotel that take into consideration the reviews from each site. We use the term suspicious hotel to refer to hotels whose reviews seem to have been manipulated, e.g. padded with fake reviews. Note that we only focus on trustworthiness and not the relevance of reviews for a particular user, e.g. whether a hotel is a pet-friendly, which is an important but distinct problem.

We focus on hotel reviews for a combination of reasons. First, the hospitality industry exhibits relative stability and consistency. Proximity to the beach does not change, unlike product properties which quickly lose their attractiveness, let alone that the products themselves may be replaced altogether by newer models. Second, hotel properties provide some structure and ubiquitousness. Hotels are consistently evaluated on a relatively limited number of factors (cleanliness, service, location, noise, comfort) as opposed to say electronic goods which can vary significantly depending on the interests and intent of the user (e.g. think digital cameras, or TVs). Our work could easily be expanded to reviews of other “well-defined and persistent” services, such as restaurants. Thus, our overall framework and fundamental functions are a great starting point for expanding our work to other commercial sectors, even if some sector-specific adjustments are required. Specifically, we intend to expand our work to restaurants in the near future.

Most previous work so far has focused on analyzing a single review site, and typically, focus on temporal [7], textual [13], behavioral [16], distributional [8] and graphical features.
to detect fraudulent reviews. In section 7, we discuss the existing work in more detail.

As our key contribution, we develop a systematic methodology to analyze, compare, and synthesize reviews from multiple review sites. The key novelty of our work lies in its focus on multiple review sites. Another key contribution is a systematic method to match hotel identities across different sites. First, we introduce and evaluate features that capture cross-site discrepancies effectively. Second, we conduct arguably the first extensive study of cross-site discrepancies using real data. Third, we provide the TrueView score, a non-trivial synthesis of multi-site reviews, that assesses the trustworthiness of a group of reviews. We provide TrueView as a proof of concept that a cross-site analysis can significantly improve the information that the user sees. In our study, we use more than 15M reviews from more than 3.5M users spanning three prominent travel sites, TripAdvisor, Hotels.com, Booking.com spanning five years for each site.

We highlight our contributions and key results below.

a. A systematic approach to cross-site review evaluation using behavioral features. We propose a comprehensive methodology for comparing and synthesizing reviews from different sites. We use 142 features that go beyond simple rating analysis to consider a set of behavioral and contextual features including review-centric, reviewer-centric, and hotel-centric features. Our features capture temporal, spatial, behavioral, and graph-based characteristics, which provides a multi-faceted view of the reviewing process of a hotel. A key feature of the work is that we evaluate the trustworthiness of the overall review in one site using cross-site features leveraging information from the other sites. We find that using cross-site features nearly double the number of suspicious hotels that we find in our experiments.

b. An extensive study of cross-site review differences. We apply our approach to our 15M reviews spanning three sites. As a necessary step, we develop an automated method to match hotel identities on different sites, which is a non-trivial problem. Our study provides several interesting observations:

1. Our identity-matching method matches hotels with 93% accuracy which we validate manually. This method could be of independent interest even outside the scope of this work.

2. There are big differences in the overall score of a hotel across different sites. We find that 10.4% of common hotels from Booking.com and TripAdvisor.com, 9.3% from Hotels.com and TripAdvisor.com, exhibit significantly different rating characteristics, which is usually a sign of suspicious behavior.

3. Using multiple sites can help us detect 7 times more suspicious hotels than the union of suspicious hotels found for each site in isolation.

c. Developing a cross-site scoring system: TrueView. We develop the TrueView score as a proof of concept that leveraging multiple sites can be very informative and change our assessment of the hotels significantly. TrueView is a sophisticated: (a) temporal, contextual, and behavioral features from each site, and (b) cross-site features across the sites. By applying TrueView, we find that 20% of all hotels appearing in all three sites seem to have low trustworthiness score (TrueView score less than 0.75). Although there may be better ways to combine cross-site reviews, we argue that TrueView already demonstrates the potential of such an approach.

Our work in perspective. Our work is arguably the first effort that focuses on the synthesis of reviews from different sites. Our goal is to raise the awareness of the opportunity and the challenges related to this problem. At the same time, our approach provides a foundation for follow up studies in the following directions: (a) detection of fraudulent behaviors, (b) assessing the trustworthiness of review sites, since some may have policies that enable misbehavior, and (c) creating effective review aggregation solutions. Ultimately, the collective wisdom of the users is valuable and empowering, and we would like to protect this from fraudulent behaviors.

2. LOCATION DISAMBIGUATION

We crawled TripAdvisor.com, Hotels.com, and Booking.com. We collected all the reviews for some hotels in these sites. For each review, we collected its text, reviewer-name, date, time and rating. For each hotel, we collected its address. A summary of the whole dataset is given in Table 1.

The three websites have a lot of hotels in common and thus, provide rich information about those hotels together. We focus on location disambiguation problem across these three websites.

2.1 Challenges

The most important challenge in location disambiguation is that hotel names are not unique. Therefore, a full name and address is needed to uniquely identify a hotel. Unfortunately, addresses are also not standard across websites, and the differences between sites are seemingly random. Differences can be as simple as Street versus St., or as complex as Hotel Atlântico Búzios Convention & Resort - Estrada da Usina, s/n, Humaita, Búzios, CEP 28950-000 and Hotel Atlantico Buzios Convention and Resort, strada da Usina 294, Morro do Humaíta, Búzios, RJ, 28950-000, Brazil. For international addresses, words can be in different orders, names can be formatted differently, country names can be excluded, and numbers can be morphed. Addresses can even use waypoints as reference, such as 1000 Yang Gao Road N Pudong Near Pilot Free Trade Zone Gate 3, which are not standard across websites. Even US addresses, which one might assume follow a standard format, are not immune. For example: Hilton New Orleans Riverside, 2 Poydras Street, New Orleans Central Business District, New Orleans, LA 70130 and Hilton New Orleans Riverside, 2 Poydras St, New Orleans, LA 70130, United States. We can look at these two addresses and tell that they are describing the same hotel, but their differences are non-trivial: the use of 'Two' versus 2 and the inclusion of 'New Orleans Central Business District'. Another domestic example: Mauna Kea Beach Hotel, 62-100 Mauna Kea Beach Drive, Wai'oloa, HI 96743 and Mauna Kea Beach Hotel, 62-100 Mauna Kea Beach Dr, Kamuela, HI, 96743 United States. These two addresses have every field in common but one: one city is listed as Waikoloa and the other as Kamuela. How much weight should be given to each field? Due to the variety of differences possible, this is a difficult problem to automate.
2.2 Disambiguation techniques

We use a combination of hotel name analysis and geodesic distance to disambiguate hotels. Geodesic distance ensures that the addresses are located in the same place, despite differences in formatting, and name comparison makes sure the hotels' names are similar enough to likely refer to the same business.

**Hotel name comparison** To compare two hotel names, we divide a similarity measure comparing the number of letters they have in common. The similarity measure we use is the length of the set intersection of the hotel names divided by the length of the longer name. This measure is faster to compute than edit distance and succeeds in the face of small spelling differences or omitted words. This measure on its own has very high precision but low recall, so when combined with geodesic distance we are able to loosen this matching requirement for good results.

**Geodesic distance** To compare the physical location of the hotels, we employ geocoding. Using the Google Geocoding API, we translate hotel addresses into latitude and longitude coordinates for all of our hotels. This API works well with strangely formatted addresses, both domestic and international. We use a cluster of computers to speed up the coordinate generation process as there is a limit on the number of requests per day. We then calculate the geodesic distance between two sets of latitude and longitudes. To do this we first convert latitude and longitude to spherical coordinates in radians, compute the arc length, then multiply this by the radius of earth in miles to get the distance in miles between the two addresses.

**Combining measures:** By combining geodesic distance with a distance measure of the hotel names, we are able to have efficient, high-precision matching. To find the ideal distance maximum and similarity minimum, we explored the parameter space for distances from 90 miles to 0 miles and similarities from .1 to .9. By sampling 30 hotels at each parameter combination and manually checking matching fidelity, we found a local maximum in precision at a geodesic distance max of 1 mile and a name similarity minimum of 0.66. Since we want high-quality matching results, we err on the side of caution with our matching constraints.

**Results of disambiguation** Using these constraints, we find 13,100 total matches. 848 hotels were matched across all three sites, 1007 between Booking.com and Hotels.com, 655 between Booking.com and TripAdvisor.com, and 10,590 between Hotels.com and TripAdvisor.com. Booking.com is a much newer site, and we hypothesize that is the reason for its reduced coverage.

3. NOVEL FEATURE SET

Hotels show various inconsistencies within and across hosting sites. In this section, we present several such inconsistencies that are not discussed previously in the literature and derive relevant features for our trustworthiness scoring. We categorize our features in a similar way to Jindal et al. [9]: reviewer-centric features, review-centric features and hotel-centric features. Features are either generated using one site or multiple sites after the hotels are joined based on location. All the single-site features are combined to put them in the context of multiple sites.

A note worth mentioning is that the location information of these hotels provide an unprecedented opportunity to validate goodness of the reviewers and reviews. All of our novel features described below show promising capabilites in identifying suspicious behavior, and most of the time it is possible to spot such behavior for our novel location disambiguation and merging.

In the reminder of the paper, we will use HDC for Hotels.com, TA for TripAdvisor.com and BDC for Booking.com.

3.1 Reviewer-Centric Features

We identify three new scenarios involving inconsistent and unlikely reviews and capture these scenarios through reviewer-centric features. Since it is not possible to disambiguate users across sites, all of the reviewer-centric features are based on individual sites.

3.1.1 Spatial Inconsistency

We focus on identifying reviewing behaviors that are spatially inconsistent. We find a user named “AmishBoy” who reviewed 34 hotels in Amish Country located in Lancaster, PA over 5 years. Lancaster County spans only 984 square miles, meaning that many of the hotels this user reviewed are right next door to each other. He gave 32 out of these 34 hotels 4 or 5 star ratings, which means he was pleased with his stay. Why then would he continually switch to new hotels? Even if he is simply visiting the area for business or pleasure every couple of months, his pattern of continually switching to new hotels is suspicious. Whether innocent or not, the reviews of this user should be discounted on the grounds that he does not represent a ‘typical’ traveler visiting this area.

We create a feature that identifies if a user has such a bias to a specific location and accumulate the contributions of such reviewers to every hotel. This feature is first calculated as the maximum count of the number of reviews a reviewer made in a given zip-code. If this value is greater than a threshold, typically 5, the reviewer is marked as a suspicious user. To propagate the feature up to the hotel level, we then sum the number of reviews each hotel has that came from suspicious users. If a hotel has many reviewers who have such spatial preference, it strongly suggests potential review manipulation by the hotel.

3.1.2 Temporal Inconsistency

Another type of suspicious behavior of a reviewer is writing many reviews on the same day. Typically a traveler may review a set of businesses after he comes back from a trip. However, the frequency of such trips and the spatial distribution of the businesses can provide valuable insights. For example, a user named “tonyk81” reviewed 25 businesses on January 22, 2006, which are located across 15 states. Additionally, 21 of these reviews are “first time reviews” for the respective hotels. The reviews describe visit dates starting from May, 2005 till January, 2006 and all of the reviews have the words good and great in the titles. Furthermore, many sentences in the review titles and text are repeated verbatim, for example “Cleanliness seems to be a high priority at this hotel and the rooms are in like new condition.” and “This hotel lives up to Marriott’s high standards of quality.” The user has only reviewed 3 hotels in the last eight years after this burst, all on the same day in 2007. Putting these pieces of evidence together, it is clear that tonyk81 is likely a spam account.
Table 1: Simple Statistics of the three datasets collected from three prominent travel websites.

<table>
<thead>
<tr>
<th>Number of reviews</th>
<th>Booking.com</th>
<th>Hotels.com</th>
<th>TripAdvisor.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>11,275,833</td>
<td>9,050,133</td>
<td>3,167,035</td>
<td></td>
</tr>
<tr>
<td>Number of hotels</td>
<td>52,696</td>
<td>155,763</td>
<td>51,395</td>
</tr>
<tr>
<td>Number of unique usernames</td>
<td>1,462,460</td>
<td>1,020,054</td>
<td>1,426,252</td>
</tr>
<tr>
<td>Average number of reviews</td>
<td>213.98</td>
<td>74.3</td>
<td>68.71</td>
</tr>
<tr>
<td>Average rating</td>
<td>3.35</td>
<td>3.07</td>
<td>3.99</td>
</tr>
<tr>
<td>Percent empty reviews</td>
<td>24.4%</td>
<td>19.6%</td>
<td>0.0039%</td>
</tr>
<tr>
<td>Date range</td>
<td>09-30-2009 - 05-17-2014</td>
<td>02/01/2006 - 06-01-2014</td>
<td>02/28/2001 - 09/08/2013</td>
</tr>
<tr>
<td>Geographic range</td>
<td>International</td>
<td>International</td>
<td>United States</td>
</tr>
</tbody>
</table>

Figure 1: (top) Locations of the hotels the user AmishBoy reviewed around Lancaster, PA in TA (bottom) A snapshot of the reviews tonyk81 wrote in TA. Note the frequent occurrences of January 22, 2006.

Singleton bursts have been identified as potential indicator of manipulating hotels in [22]. The above example suggests non-singleton bursts can show evidence of spamming users as well. If a hotel has a large number of such reviewers, it likely has manipulated its reviews. To calculate this feature, we find the maximum number of reviews a reviewer left in a given day. If this is greater than a threshold, typically 3, that user is marked as suspicious. We then propagate up to the hotel level as described above, by summing the number of reviews each hotel has that came from suspicious users.

3.1.3 Graphical Inconsistency

Bipartite cliques in user-store graphs can identify groups of users that boost a set of stores by doing a number of small transactions in the form of positive reviews [6]. Inspired by this, we create a feature to capture information about the cliques a reviewer participates in. We restrict ourselves only to cliques of size two and search for the maximum number of reviews a user has reviewed in common with any other user. We find several cases where such a clique points to an abnormal pattern in reviews. One such case we identify is two users who have reviewed over 95% of the hotels in common on nearly the same dates, usually off by one day. Upon researching the reviewers further we discovered that they are married! While a valid reason for such behavior, this shows that our method for finding anomalous behavior by identifying bipartite cliques is successful. Another example is two users from Germany, who have reviewed 117 of the same restaurants, hotels, or landmarks all over the world in the past 5 years out of 189 and 186, respectively. Furthermore, they each are ‘first to review’ (meaning they were one of the first five reviewers of that location in their native language) for 136 and 144 locations. They also have surprising synchronicity in both the review dates and stay dates for the hotels they have in common. These facts indicate that these may be two accounts may be held by the same person, which is the behavior of a paid spammer.

3.2 Hotel-Centric Features

Hotel-centric features are relatively more successful in identifying outliers (see Experiments). As before, we have three types of inconsistencies for hotels.

3.2.1 Temporal Bursts and Oscillations

Bursts of positive reviews for a hotel specifically aimed to increase the average rating is well identified in the literature. We find several such cases where there are bursts of singleton reviews with an increase in the average ratings. Typically such behavior is prevalent in the early years of the inception of a hotel in the hosting site. Figure 2 shows an example of such a burst. Bursts were calculated by taking the difference between maximum number of reviews in a day and average number of reviews in a day for each hotel.

In addition to bursts, we show cases of oscillation in ratings that are targeted to keep a high rating at the top of the “new-to-old” sorted list. We created two features to characterize oscillation: the number of times a 5 star review was immediately followed by a 1 star review for each hotel, and the number of times a 1 star review was immediately followed by a 5 star review for each hotel. These summed feature values capture the level of oscillation of ratings for a given hotel.
An example of oscillating ratings is shown in Figure 2. We see that eight out of ten 1 star ratings were followed immediately by 5 star ratings.

### 3.2.2 Temporal Correlation

Temporal correlation between number of ratings of a hotel in different sites is a valuable indicator of consistent customer satisfaction. Commonly, a hotel should have very similar behavior across sites in terms of number of ratings. For example, any hotel in Myrtle Beach, SC shows a high number of ratings per day in summer, decreasing in the winter. Since the number of occupants at hotels in Myrtle Beach, SC [2] decreases from 80% in the summer to 30% in the winter, this pattern makes sense; the number of ratings per day follows the average occupants of the hotel and such behavior is consistent across sites. In Figure 4, we show hotels and their number of ratings per day in HDC and TA. We see the top two hotels have summer peaks in both of the sites. However, Bluewater Resort has dense summer ratings in TA but no significant peak/density in HDC, especially in summer of 2012.

Such discrepancy in temporal behavior can exist for several reasons, each of which are worth considering when it comes to evaluating hotels. The hotel might have received more poor ratings in the summer which were omitted by the hosting site, or the hotel could have sponsored some winter reviews. There can be extreme cases such as the hotel was closed for construction, but irrespective of the real reason, a lack of correlation across sites is a significant indicator for spotting misbehavior. We calculate Pearson’s correlation coefficient between the time series of number of ratings per day from two sites. We use only the overlapping time duration for calculation. If the overlap is too small, we naively assume perfect correlation.

### 3.2.3 Rating Distributions

In [8], authors have suggested that the shape of rating distribution has a strong tie to the spamming hotels. Typically, a “J” shaped distribution has a large number of 5-star and 1-star ratings but a small number of 4-, 3-, and 2-star ratings. Such a distribution may occur if users review only when they are extremely happy or disappointed. However, it is suspicious to have nearly equal numbers of people who hated and loved a hotel. This suggests that one of those rating groups has been artificially inflated by spam reviews. Whereas in [8] they had to compare distributions between hotels, we take a multi-site approach to validate the consistency of distributions across sites for individual hotels. To find this value we calculate the Pearson’s correlation coefficient between the rating distributions of a hotel across different sites, represented by two vectors of counts of integer ratings. We also directly compare the distributions of the ratings by using the one-sided p-value from the Mann Whitney U Test as a feature.

Figure 4 shows the correlation coefficients of the distributions between HDC and TA. We take a threshold of -0.9 and locate all the hotels that show less than -0.9 correlation. As shown in Figure 3, we find an interesting co-location pattern for negatively correlated hotels around Atlanta, GA which is more densely populated with such hotels than any major urban area or popular vacation spot, such as Las Vegas, NV or Los Angeles, CA. Around 5% of the hotels in GA show negative correlation, which is much greater than that (1%) in CA and NV.

### 3.3 Review-Centric Features

Review-centric features are based on the texts of the reviews and their titles. We do not use any natural language processing techniques to spot a spam review by only text, rather rather focus on structural properties of reviews.

#### 3.3.1 Empty Reviews

Reviewers often write empty reviews with no title and no comment. The incentive for such reviews is that they can be submitted without typing. We find an abundance of empty reviews in HDC and BDC, with 20% and 25% empty reviews, respectively, while TA has only 3% empty reviews. In addition to these reviews being potentially suspicious, we find that hosting sites treat these reviews differently. For example, in HDC, empty reviews were visible during the time when we collected the data; however, HDC now hides the empty reviews, while keeping counts of them in their over-
Figure 3: (left) The time series of number of ratings per day for three hotels in Myrtle Beach, SC. The top two hotels show summer peaks in both TA and HDC. The bottom hotel does not show any summer effect in HDC. (center) Distribution of correlation between ratings of hotels in TA and HDC. (top right) Distribution of ratings for the same hotel in two websites showing negative correlation. (bottom right) Negatively correlated hotels are more abundant in the vicinity of Atlanta, GA and almost non-existent in Las Vegas, NV area where both the cities have more than three hundred hotels.

Figure 4: Dissimilar distribution of ratings, temporal bursts in number of ratings per day and frequent empty reviews for the same hotel in Statesboro, GA.

dissimilar distribution of ratings, temporal bursts in number of ratings per day and frequent empty reviews for the same hotel in Statesboro, GA.

all review tallies. For example, Excalibur Hotel and Casino in Las Vegas, NV has 2500 empty reviews out of 13,144 reviews in HDC, but none of them is visible through the website. Such behavior of hiding consumer reviews has been reported several times in [1] and clearly such an omission is not fair to consumers.

There exist some hotels that have a significantly greater proportion of empty reviews than average. For example, Comfort Suites Newark in New Jersey has 86 empty reviews out of 193 reviews which is more than 44% of the reviews. We find 66 hotels that have only empty reviews in HDC. BDC has a similar-sized body of empty reviews. Sometimes multiple features together describe a suspicious hotel. In Figure 3, we show a hotel having three suspicious behavior captured by three of our features.

3.3.2 Matching Reviews

Inspired by tonyk81’s matching review text described above, we create a feature that captures the proportion of sentences any pair of reviews from the same reviewer share to the total number of reviews that reviewer made. While repeating the occasional phrase is to be expected, having reviews that share the majority of the sentences, or are completely identical, is not. We calculate this score by comparing pairwise all reviews made by a given user. We keep a count of the number of sentences these reviews share, and then divide this value by the total number of reviews a reviewer wrote. We then attach this score to each review a reviewer left. For each hotel we aggregate this score for all of its reviews to characterize the lack of uniqueness of its review set. A hotel with many reviewers who use the same sentences repeatedly, suggests that the hotel is soliciting many spam reviews.
3.4 Cross-Site Feature Preparation

We generate 90 features including the novel features described in the previous section for each hotel in each site. In addition to these single-site features, we use to calculate a TrueView score, we calculate 52 cross-site features for the hotels that exist in multiple sites. These cross-site features are combination of similar single-site features from many sites. Cross-site features can identify discrepancies that a hotel has between a pair of sites, for example, a difference in the percentages of empty reviews.

We combine the single-site features in three different ways. First, we take ratios for scalar values, such as number of reviews. Some of the scalars are only meaningful relatively, such as number of 5-star ratings. We use relative frequencies when taking the ratio of these scalar features. Second, we take correlation coefficients for sequences, such as numbers of high to low ratings. And third, we take the p-value of the Mann-Whitney U test for distributions, such as the distributions of review lengths in words.

The next step is to normalize these features to make them reasonably close to standard normal distribution. The intention is to treat each feature as independent and identically distributed. Some of the features require a log transformation to be converted to standard normal because of their exponential fall out, for example, counts of rating scores, counts of empty reviews, and text similarity scores. After log transformations, we normalize each feature by Z-score. For a complete list of features and their values for all of the hotels please visit the supporting page [3].

4. TRUSTWORTHINESS SCORE

Our work is fundamentally different from existing works. We do not wish to evaluate if a review is fake or untruthful as most existing works do. We believe it is very difficult to achieve high recall in identifying review fraud, mostly because of the dynamics of the fraudsters and the lack of labeled data. Even if we assume hypothetically that we can achieve high recall and prevent fake reviews from being shown to the readers, the hotels and sites that promote fraudsters are not penalized and site users are not notified of which hotels are trying to lie to their customers.

We believe that a better approach to this problem is to evaluate hotels and sites to find which ones promote untruthful reviews, and to produce a trustworthiness score to present to the user. We think such scores are more beneficial to the review readers as they can use these scores in their ultimate judgment.

4.1 Outlier Scores

Once all the features are available, we take three approaches to rank the hotels with outlier scores. We use global density-based score, a local outlier factor, and a hierarchical cluster-based score. We describe each technique in this section.

4.1.1 Global Density-based Score

We calculate how different a hotel is from the centroid of hotels in the feature space. We take the mode of each feature and form a hypothetical hotel which lies in the center of a large “normal” cluster. We use the simple intuition that most hotels are playing fair and only a small fraction of hotels are generating fake and untruthful reviews. If a hotel is largely dissimilar to the mode/centroid hotel, the hotel might disagree in many features with the mode, which makes its reviews less trustworthy.

We use the concept of density connectedness to form a large cluster of points (i.e. hotels) carrying the mode. A point is a core point if it has $k$ or more points within $\epsilon$ distance in the Euclidean space. Two points are density-connected if there is a sequence of core points from one to another where every point is within $\epsilon$-neighborhood of the previous point. Thus, any point that is not in the core cluster is an outlier and the degree of outlying is proportional to the distance from the nearest core point. This method is inspired from the original density based clustering algorithm, DBSCAN. This method has a unique advantage that the core cluster can be of any shape and based on the two parameters (i.e. $\epsilon$ and $k$) we can control the size of the core cluster and thus, the size of the outlying cluster.

4.1.2 Local Outlier Factor (LOF)

The global method assumes that all normal hotels form a core cluster. However, there can be alternative structures with numerous small clusters of normal hotels with varying densities in the feature space. We use local outlier factor to score outliers. Local outlier factor uses the notion of k-distance ($dist_k(x)$) which is defined as the distance to the $k$-th nearest neighbor of a point. Local reachability distance ($lrd_k$) of $x$ is the average of the reachability distances from $x$’s neighbors to $x$. Here $N_k(x)$ is the set of $k$-nearest neighbors of $x$.

$$lrd_k(x) = \frac{\sum_{y \in N_k(x)} lrd_k(y)}{|N_k(x)|}$$

The LOF of a point $x$ is the average of the ratios of the local reachability of $x$ and its $k$-nearest neighbors. LOF can capture several normal clusters of arbitrary densities that makes it robust for any data domain. Formally, LOF is defined as below

$$LOF_k(x) = \frac{\sum_{y \in N_k(x)} lrd_k(y)}{|N_k(x)|}$$

4.1.3 Hierarchical Cluster-based Score

Both of the density-based methods above use $k$-neighborhoods to estimate densities or connectivities. Our third approach differs from that and uses a hierarchical clustering approach. We cluster the data hierarchically using the single linkage method. The single linkage method starts with singleton clusters. The method evaluates pairs of clusters based on the minimum distance between any pair of points that form the two clusters and merges the closest two clusters at every step. This simple bottom-up strategy can produce a dendrogram over the points without any input parameters.

Once the complete hierarchy is built, a set of clusters can be obtained by cutting the hierarchy at a cutoff level. Here again, we assume there is a global normal cluster that contains most hotels and any hotel not in this cluster is an outlier. Under this assumption, we can tune the cutoff level to get a certain percentage of points in the set of outliers.

4.2 TrueView Scores

The three above approaches produce three different outlier scores. A big challenge is that the scores are not at the same scale. Before combining individual scores, we need to regularize and normalize the outlier scores. We take the approach of [11][19] to normalize the outlier scores from each of
the above approaches using logarithmic inversion and gaussian scaling.

Let a hotel be \( x \) and the outlier score of \( x \) is \( S(x) \). We do a log transform \( z = -\log S(x) / S_{\text{max}} \) for each hotel \( x \) and scale it using a gaussian distribution to produce a probability \( P(x) \) of the hotel \( x \) being an outlier. Such a probability score is useful because it is bounded within \([0,1]\) and can easily be converted to a trustworthiness score by a simple linear inversion.

\[
P(x) = \max(0, erf \left( \frac{z - \mu}{\sigma} \right))
\]

\[
TV(x) = 1 - P(x)
\]

Here, \( TV(x) \) is the TrueView score. The TrueView score describes the probability of a hotel’s reviews being truthful and is unitless. As described above, we can now produce TrueView scores from any feature set of these hotels. We define two intermediate scores that can be created as well as the overall TrueView score. First, \( \text{TV}_1 \) is produced only using features from one site. Second, \( \text{TV}_2 \) is produced using an union of features from all sites. Third, the TrueView score is produced using the union of all single-site features and the cross-site features from all three sites. Each outlier detection algorithm produces its own scores, and we average them to get the overall TrueView score. We provide empirical evidence in Experiments section that TrueView identifies successfully identifies outliers.

There are two weaknesses of the above approaches. First, the score of a hotel can change if other hotels connect it to the core cluster. Second, there can be hotels which are unusually good labeled as untrustworthy. These cases are pathological and become rare with more reviews per hotel.

5. EXPERIMENTS

We start with our reproducibility statement, all of the experiments in this section are exactly reproducible with the code and data provided in the supporting page [3]. We also have additional materials such as presentation slides, csv sheets of raw numbers, and more experiments.

5.1 Parameter Sensitivity

The three algorithms we use for outlier detection have a set of parameters to tune. Based on the values of these parameters, we can have different sized sets of outliers. We experiment to test the sensitivities of these parameters and select values for subsequent experiments.

![Figure 5: (left) Percentage of outliers detected in the density-based method as we vary \( \epsilon \). (right) The same as we vary neighborhood size \( k \) in LOF method.](image)

In the global density-based method, we have \( \epsilon \) and \( k \) as parameters. We fix \( k = 10 \) and vary \( \epsilon \) and record the percentage of the dataset labeled as outlier. In the local outlier factor method, we have the neighborhood size \( k \) as the parameter. In the hierarchical method we use the cutoff value as the parameter and record the percentage of the dataset labeled as outlier. Figure 5 shows the behavior of mode-distance and LOF methods. We select the parameters such that we can rank order 30% of the hotels using all three methods. This number is an arbitrary choice and can be different in other systems.

5.2 Feature Importance

We evaluate feature importance in the calculation of the TrueView score. We use two different and independent approaches to evaluate the features.

5.2.1 Spectral Feature Selection

We use the method in [23] to evaluate the importance of our 142 features. The method identifies redundancy in the feature values and properly demotes dependent features. The result is shown in Figure 6, where more than 100 features show very uniform importance scores, which supports the validity of our feature set. In addition, we categorize the features into cross-site and single-site classes. We see that the most important features are cross-site features showing the importance of multiple-site data for evaluating hotels. Note that, the feature selection algorithm does not take into account the ultimate usage of these features in the algorithm, which is outlier detection in this work.

![Figure 6: Feature importance percent score for 142 features. Cross-site features are more important than single-site features.](image)

5.2.2 Distance from Mode

In this approach, we consider the outliers produced by the global density-based approach. We pick the most different feature of an outlier with respect to the mode of that feature as the representative for that outlier. We see a massive shift in importance of our cross-site features, especially the star rating based cross-site features that become important for more than half of the outliers. In Figure 7, we show the results for other feature categories as described in section 3.

5.3 Validation

Since the algorithm for computing TrueView score is unsupervised, and we do not have labeled data that identifies fraud hotels, we cannot directly validate with ground truth information. Therefore we take two alternate approaches to validate the soundness of our method.
5.3.1 Sanity Check on Synthetic Frauds

First we generate a synthetic set of outlying hotels to validate that our global density-based approach correctly identifies outliers. To create these outliers, we copy the center point representing the mode of all features and mutate random feature values, randomly setting them to either the ninety-fifth or the fifth percentile. We then calculate their TrueView score based on the global density-based method described above. This experiment is repeated 100 times. We find the distributions of TrueView scores are heavily skewed towards zero, showing that on the whole, they were given very low TrueView scores. Thus our algorithm passes this sanity check, classifying 100% of our synthetic data as outliers and giving them correspondingly low TrueView scores.

5.3.2 Evaluation of Extreme Features

Second, we validate that the hotels with low TrueView scores show a significant difference in the number of extreme feature values from the hotels with high TrueView scores. To calculate this we find the number of features that are below the fifth percentile or above the ninety-fifth percentile for each hotel. We use the Wilcoxon rank-sum test to determine whether the 40 most trustworthy hotels significantly differ in the number of extreme features from the 40 least trustworthy accounts. A p-value of < 0.05 rejects the null hypothesis, and asserts that the two populations significantly differ in the number of extreme features from the hotels with high TrueView scores. Table 2 lists the p-values for each outlier algorithm and each feature subset. The cross-site and combined single-site feature sets show statistically significant results for every algorithm, meaning that these feature sets are effective in differentiating outliers. It also means that hotels given high TrueView scores are indeed trustworthy, as most of them have only a few extreme features.

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>LOF</th>
<th>Mode Density</th>
<th>Linkage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-site</td>
<td>3.93E-07</td>
<td>1.63E-08</td>
<td>4.60E-11</td>
</tr>
<tr>
<td>Single-site</td>
<td>1.63E-06</td>
<td>8.04E-15</td>
<td>1.08E-12</td>
</tr>
<tr>
<td>Booking.com</td>
<td>1.49E-05</td>
<td>0.419</td>
<td>0.019</td>
</tr>
<tr>
<td>Hotels.com</td>
<td>0.308</td>
<td>4.41E-04</td>
<td>0.038</td>
</tr>
<tr>
<td>TripAdvisor.com</td>
<td>0.379</td>
<td>0.052</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Table 2: p-values of Wilcoxon rank-sum test. Bold faced values mean that there is a significant difference between the top and bottom 40.

6. CASE STUDIES

We want to provide some initial ideas on how the TrueView score could be used in practice. We identify two possibilities: (a) enable the site owner to detect misbehaving hotels, and (b) by the end user.

- **A. TrueView for the site administrator.** The usage here is fairly straightforward as a way to identify misbehaving hotels. The administrator can apply TrueView on the data from his own site alone, in which case TrueView resolves to TV1. Assuming cross-site cooperation, the administrator could use data from other sites.

- **B. TrueView for the end user.** The TrueView score could help the user select hotels that are more likely to have reliable TrueView score are not even shown to the user.

- **b. Weighted rating using TrueView.** There are many ways that to modify the rating of a hotel based on the trustworthiness score. The full input is the rating from each site, the number of reviews per site, the trustworthiness of each site (TV1), and the TrueView across all sites. One approach would be to take the average rating and multiply it by the trustworthiness score, but this method’s simplicity is not a proof of effectiveness without extensive study.

![Figure 8: (left) Distribution of the number of extreme features (95th percentile) in the bottom 100 hotels in TrueView ordering (right) Distribution of the same in the top 100 hotels in TrueView ordering. Distributions are significantly different.](image)

![Figure 9: Empirical cumulative distribution function of TrueView scores.](image)
### 7. RELATED WORK

Existing works focus on identifying fraud reviews and reviewers, while we focus on businesses such as hotels that promote fraudulent reviews. Existing work can be categorized based on the methodologies they adopt to detect frauds. Fraud detection using Graphical/Network structure is studied in [5][6][20] where authors exploit network effects and clique structures among reviewers and products to identify fraud. Text-based detection of fraud is studied to spot a situation of review manipulation. In [7], there are significant variations in reviews, and we find evidence of review manipulation.

<table>
<thead>
<tr>
<th>Hotel</th>
<th>HDC Stars</th>
<th>T1-HDC</th>
<th>TA Stars</th>
<th>TV1-TA</th>
<th>TV2</th>
<th>TrueView</th>
<th>Trustworthy?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Super 8 Terre Haut</td>
<td>3.30</td>
<td>1.0</td>
<td>3.00</td>
<td>0.235</td>
<td>0.734</td>
<td>1.0</td>
<td>✓</td>
</tr>
<tr>
<td>Excalibur Hotel and Casino</td>
<td>3.60</td>
<td>0.846</td>
<td>3.50</td>
<td>0.850</td>
<td>0.951</td>
<td>0.761</td>
<td>✓</td>
</tr>
<tr>
<td>Comfort Suites Newark</td>
<td>3.80</td>
<td>0.688</td>
<td>3.00</td>
<td>0.705</td>
<td>0.773</td>
<td>1.0</td>
<td>✓</td>
</tr>
<tr>
<td>Paradise Resort, Myrtle Beach</td>
<td>4.20</td>
<td>1.0</td>
<td>4.00</td>
<td>0.573</td>
<td>0.699</td>
<td>1.0</td>
<td>✓</td>
</tr>
<tr>
<td>Bluewater Resort, Myrtle Beach</td>
<td>2.90</td>
<td>1.0</td>
<td>3.00</td>
<td>0.735</td>
<td>0.790</td>
<td>0.462</td>
<td>X</td>
</tr>
<tr>
<td>Comfort Inn &amp; Suites, Statesboro</td>
<td>3.60</td>
<td>1.0</td>
<td>3.60</td>
<td>0.783</td>
<td>0.669</td>
<td>0.264</td>
<td>X</td>
</tr>
</tbody>
</table>

**Table 3: TrueView scores for suspicious hotels. (Stars is the average overall rating)**

b. Filtering using TrueView. In this method, we need only to find the cut-off threshold of trustworthiness TV-thres which forms the cut-off point for hotels with a score less than the threshold. The filtering threshold needs to balance the ability to remove suspicious hotels with its false positive rate, which would reduce the overall choice for the end user. This is the fundamental trade-off that needs to be considered. We present some initial thoughts on how we could develop a threshold like this.

Disclaimer: The discussion below is not intended to be indicative of and not a conclusive and thorough determination of a filtering method. For that, a more extensive analysis is needed to derive a conclusive statement. In addition, the user-facing service and even the user herself can have significant impact on how “strict” the filtering process should be.

Calibration: known suspicious hotels. First, we show the distribution of hotels having a TrueView score less than a threshold in Figure 9. Obviously, we cannot just mark the bottom 25% as being untrustworthy. However, we can evaluate how well our scoring process works for a known set of misbehaving hotels. If we filter out the bottom 25%, we need a threshold of 0.7. We present results for the 6 suspicious hotels that were identified in Section 3 in Table 3. We see a threshold of 0.7 would give us 33% precision. In other words, filtering out hotels with a score of less than 0.7 would ensure that 2/6 known malicious hotels are filtered, while reducing the overall choice of hotels by 25%.

For the sake of this case study, one can argue that 0.7 could be a reasonable threshold. In a real deployment, this needs to be based on a much large labeled dataset.

**Case study: Finding hotels in Amish Country.** We choose the top 20 hotels in Amish Country, Lancaster County, PA from Hotels.com and TripAdvisor. We choose hotels in Amish Country because during our initial investigation many potentially suspicious hotels were present. Amish Country is a small area with many competing hotels trying to access a similar market, so the competition is fierce. We calculate the TrueView scores for those that are part of our matched dataset. We find that two hotels have scores below our threshold of 0.7. We manually assess these two hotels, and we find that they do show suspicious behavior.

### 8. CONCLUSION

The goal of our work is to show the significant benefits we can have by combining reviews from multiple review sites. As our key contribution, we develop a systematic methodology to cross-compare, and synthesize reviews from multiple review sites. The novelty of our approach relies on the introduction and assessment of 142 features that capture single-site and cross-site discrepancies effectively. Our approach culminates with the introduction of the TrueView score, in three different variants, as a proof of concept that the synthesis of multi-site reviews, can provide important and usable information to the end user. We conduct arguably the first extensive study of cross-site discrepancies using real data from 15M reviews from more than 3.5M users spanning three prominent travel sites. We find that there are significant variations in reviews, and we find evidence of review manipulation.
9. FUNDING STATEMENT

Amanda Minnich is supported by the NSF Graduate Research Fellowship under Grant No. DGE-0237002 and the Program in Interdisciplinary Biomedical and Biological Sciences through the University of New Mexico award number T32EB009414 from the National Institute of Biomedical Imaging and Bioengineering. This work was also supported by NSF SaTC Grant No. 1314935.

10. REFERENCES


www.cs.unm.edu/~aminnich/trueview.


