Predicting Upvotes and Downvotes in Location-Based Social Networks Using Machine Learning

Jianxi Zhang, Jinyan Zhu, Tianzheng Meng, Chenfan Zhuang, and Yang Chen

School of Computer Science, Fudan University, Shanghai 200433, China
{jianxizhang18, jinyanzhu19, chenyang}@fudan.edu.cn, tianzhengmeng@hotmail.com, zhuangchenfan@gmail.com

Abstract. Nowadays, Online Social Networks (OSNs) have become indispensable spaces for people to express their opinions. In order to evaluate comments, tips, answers or posts, most OSNs design “upvote” or “like” buttons, and some of them provide “downvote” or “dislike” buttons as well. While there are some existing works making predictions related to upvote, downvote prediction has never been systematically explored in OSNs before. However, downvote is just as meaningful and informative as upvote, representing opposite voices. In this paper, we focus on predicting both numbers of downvotes and upvotes together on Foursquare, a leading location-based social network (LBSN). Our work has three main contributions. Firstly, by unprecedentedly viewing downvotes and upvotes together from a holistic prospective, we discern features that are effective to the differentiation of both downvote and upvote prediction. Secondly, by making use of structural hole theory and information theory, we propose a robust model that can be used for both downvote and upvote prediction. To the best of our knowledge, we are the first to predict number of downvotes in OSNs. Finally, we complete a thorough prediction performance and feature importance analysis. Our predictions of downvotes and upvotes using XGBoost model achieve AUC scores of 0.99 and 0.98, separately. In other words, our approach not only fills the gap of downvote prediction, but also increases the prediction performance of upvote prediction in OSNs.

Keywords: Location-based social networks (LBSNs) · Machine learning · Social graph analysis · Structural hole theory · Downvote prediction

1 Introduction

Online Social Networks (OSNs) [14] are widely used for communication and connection between people. It is one of the biggest uses of the Internet nowadays. Different types of OSNs include Blog [1] such as Facebook, Twitter, and Tumblr; discussion forum such as Quora, Reddit and Zhihu; as well as location-based
social network (LBSN) [21] such as Foursquare [6], Yelp [18], and Dianping [11]. One of the key features of these OSNs is “upvote”, which is more commonly called “like”. Upvote has multiple uses in OSNs, including expressing preferences and trending content. While most OSNs provide upvote or like function [8], the “downvote” or “dislike” function [13], which may result in negative emotions, also reflects a user’s opinion from another angle. Blogs like Facebook and Twitter, which focus on sharing content, only have upvote function, for the reason that downvote would reduce people’s willingness to share [8]. Other OSNs, such as Foursquare [6], Quora and Reddit, use downvote to manifest comprehensive viewpoints. While there are many existing works about upvote, such as predicting popularity of Reddit post [10], and predicting answer quality on generic social Q&A sites [17], downvote is often overlooked. Nonetheless, sometimes conclusions extracted from upvote would be extremely misleading if ignoring downvote. For example, a viewpoint with a lot of upvotes does not guarantee correctness, for it may be controversial if it has a large amount of downvotes too. Hence, only combining upvote with downvote could ensure us to get a panorama of the community’s attitudes toward a certain post.

Our work mainly focuses on Foursquare, one of the most popular LBSNs. Foursquare has two key functions: User interaction and point of interest (POI) recommendation [28]. Through user-generated content (UGC) called tips, other users could understand POIs from those who have been there. Based on this, Foursquare allows publishers to upvote or downvote tips, which may affect the credibility of tips significantly. Also, according to [7], Foursquare was considering adding downvote to tips as a feature in tip ranking algorithm, making downvote prediction on Foursquare even more useful. It allows Foursquare to detect negative emotions on a tip in advance, and adjust tips ranking dynamically.

In this paper, we focus on the problem of predicting both upvotes and downvotes in LBSNs. We crawl a Foursquare dataset and then build a supervised machine learning-based model for the prediction. In addition to conventional features related to tip publisher, venue and tip context, we leverage structural hole theory [19] and information theory to extract some unique features to further enhance the prediction performance. We have made the following three key contributions. First, we extract a series of features according to the Foursquare dataset, and conduct a data-driven study to show how they could distinguish between upvote/downvote tips and ordinary tips1. A set of features related to descriptive information, venue, and tip context has been chosen, including the number of tips posted by the publisher, the number of friends or followers of the tip publisher, the number of unique visitors at the venue, length of tip context, etc. Second, we build a machine learning model for the upvote/downvote prediction. Besides conventional features, we adopt both structural hole theory and information theory to enhance the prediction. The entropy of countries and venue categories that the publisher has visited and the effective size have been selected as new features for the model, and they play an important role in making the prediction more accurate.

1 Here an ordinary tip denotes a tip without any upvote and downvote.
Finally, applying our model to downvote prediction, XGBoost [5] model shows an AUC score of 0.99. When using the same model on upvote prediction, the achieved AUC score is 0.98. Compared with the current baseline [27], which has a low precision of about 0.5 for SVM model, our SVM model in upvote prediction outperforms it with a precision of 0.74. Also, our feature importance analysis indicates that features related with venues are the most critical to the performance in general. More specifically, country entropy is in the dominant place of downvote prediction, while the outcome of upvote prediction is driven by a more diversified feature set.

Fig. 1. Feature differences between downvoted and non-downvoted tips.

2 Data Collection and Conventional Feature Analysis

In this section, we will firstly describe our dataset in Sect. 2.1, and then examine some conventional features to distinguish between upvote/downvote tips and ordinary tips in Sect. 2.2.
2.1 Dataset

We crawled our dataset from January 15th, 2021, using Foursquare API and a distributed Python crawler developed by us. After selecting a popular tip publisher as the starting point, we managed to use BFS algorithm to fetch 1.6 million entries, and each entry contains all the information related to a user, i.e., basic user profile, his/her posted tips (tip content, venue, number of upvotes and downvotes), and a friend list. Because of the adoption of BFS algorithm, the whole dataset represents a subnetwork of Foursquare. Therefore, for any given tip, we can extract the information of its publisher, venue, and the tip publisher’s ego network.

2.2 Analysis of Conventional Features

Normally, it takes a period of time for tips to gain upvotes and downvotes. Therefore, we examine a set of conventional features to distinguish between upvote/downvote tips and ordinary tips.

By observing the dataset, it is natural to generate the idea that whether a tip will be upvote or downvote is mainly affected by three domains of factors, i.e., descriptive information of tip publishers, venue information, and tip content information:

- **Descriptive information**: since descriptive information is a compelling exposure of the publishers’ historical behavior, we extract some common statistics such as “number of tips posted,” “the number of venues where tips were posted.”
- **Venue**: One unique character of LBSN is that every tip is associated with a venue, such as a bar, a restaurant, or a scene. Therefore, the popularity of the venue will have a significant impact on the tip’s vote number. To reflect the influence of venue, we select the following features, i.e., “number of tips posted at the venue,” “number of upvotes received by tips posted at the venue,” “number of unique visitors at the venue,” and “venue category defined by Foursquare.”
- **Tip context**: the tip content is also relevant to the numbers of upvotes and downvotes it received. For example, the information density, the attitude, and the correctness will all give the readers positive or negative feelings, and then further lead to the vote behavior. Here, we mainly use NLP measurements, such as LIWC [23] and NRC Emotion Lexicon [22]. The open-source tool used is named lexica\(^2\).

To catch a glimpse of whether these features are distinct when applying to tips with and without upvote/downvote, we select two representative features from each of the three feature categories, and visualize their distributions as shown in Fig. 1. Specifically, for sentiment analyzing, since the lexica library provides 10 types of sentiments and emotions including negative, positive, anger, anticipation, disgust, fear, joy, sadness, surprise, trust, we use all of them and simply

\(^2\) https://github.com/AbdulSaleh/lexica.
use a binary statistic of 0 or 1 to indicate whether the text includes the words of certain type of sentiment or emotion. In general, it is satisfying that all these selected features are discriminative to our predicting target. Features extracted from downvoted tips, the red line of Fig. 1, are validated to be more “active” in all kinds of criteria. More concretely, downvoted tips, as well as publishers and venues associated with them, are more likely to possess more posted tips, more friends or followers, more unique visitors, longer text, and stronger emotions. It can be assumed that such characteristics could lead to higher attention paid on the tips. Given the fact that only a small fraction of tips on Foursquare have downvote, it is reasonable that high attention of tips within the community lays the foundation of gaining downvote. The same feature analysis is also conducted on upvoting scenario generating similar outputs. Therefore, it is natural to leverage these features to further build a supervised machine learning classifier to uncover upvote/downvote tips.

3 System Design and Implementation

In this section, we propose a robust machine learning classifier with newly added features, which can be applied to both upvote and downvote prediction, as shown in Fig. 2. Section 3.1 provides an overview for the design and basic workflow of our system. Later, Sect. 3.2 and 3.3 introduce the two categories of newly added features, i.e., entropy and effective size, and demonstrate their effectiveness, separately. Finally, Sect. 3.4 describes the details of model construction and the differences between upvote and downvote prediction.

3.1 Overview

As shown in Fig. 2, our model has a feature list with four different types. Except for the conventional features mentioned in Sect. 2.2, we also add some sophisticated and meaningful features, entropy and effective size, to feature list, which will be elaborated in the following subsections. With sampled dataset divided into training set and test set, we could extract feature list of the training set and feed it into upvote/downvote classifiers. The classifiers are different supervised machine learning models, and each of them makes predictions for whether a tip will be upvoted/downvoted. After the classifiers are successfully trained, we evaluate their performance on the test set. Also, it is worth mentioning that the technologies such as location spoofing [30], would not weaken our results. This is because our model does not contain locations of those users who conduct upvote/downvote.

3.2 Entropy

In this subsection, we introduce entropy, a concept derived from information theory, to our feature list. Both country entropy and category entropy are used as the features. For the country entropy of a publisher, it is an indicator of the
Fig. 2. The prediction model of a Foursquare tip’s upvote and downvote

number of different countries and regions the publisher has visited. In detail, the “upvoting” entropy only includes the locations of those upvoted tips, while the “downvoting” entropy collects the places of tips with downvotes. Similarly, the category entropy defines the degree of difference of the tips’ category. We also distinguish the “upvoting” and “downvoting” situation for the category entropy.

We calculate the entropy according to its definition. Using the country entropy as the example, the algorithm needs the probability of each country \( P_{cnty} \) to get the result. In the “upvoting” situation, \( P_{upvoting_{cnty}} \) is defined as the ratio of the number of upvoted tips showing the country as location to the total number of tips, as shown in Eq. (1):

\[
P_{upvoting_{cnty}} = \frac{\text{number(\text{upvoting}_{cnty})}}{\text{number(\text{upvoting}_{tips})}}
\]  

(1)

Then as in Eq. (2), for each publisher \( p \), we can get the entropy using the following equation:

\[
\text{Entropy}(\text{upvoting}_{p}) = \sum_{\text{upvoting}_{cnty}} P_{upvoting_{cnty}} \times \log(P_{upvoting_{cnty}})
\]  

(2)

Then we can get the other three entropy metrics – “downvoting” country entropy, “upvoting” category entropy and “downvoting” category entropy – in the same way.

Figure 3(a) shows the cumulative distribution function (CDF) of “downvoting” country entropy of all the Foursquare tip publishers. We can see that over
Fig. 3. The CDF of country entropy: (a) graph of “downvoting” country entropy; (b) graph of “upvoting” country entropy. The red line represents people with downvotes/upvotes and the blue line is about people without downvotes/upvotes. (Color figure online)

(a) CDF of “downvoting” country entropy  
(b) CDF of “upvoting” country entropy

Fig. 4. The CDF of category entropy: (a) graph of “downvoting” category entropy; (b) graph of “upvoting” category entropy. The red line represents people with downvotes/upvotes and the blue line is about people without downvotes/upvotes. (Color figure online)

(a) CDF of “downvoting” category entropy  
(b) CDF of “upvoting” category entropy

80% of the publishers without being downvoted have an entropy of 0, which means they always label themselves as in the same place. For those whose tips are downvoted, on the contrary, the entropy varies for different publishers. Only about 10% of these publishers have an entropy of 0 and the range of 0 to 0.4 almost includes all the publishers we sampled. This indicates that people tend to visit more than 1 but still limited places. Things are similar for “upvoting” country entropy (Fig.3(b)), except that this time about 70% of the publishers without upvotes receive the entropy of 0 and more than 30% of those with upvotes have the “0” entropy.

When we look at the category entropy (Fig. 4(a)), it shows that for publishers without downvotes, those whose entropies are 0 cover more than 40% of the total number while the percentage is about 10% for publishers with downvotes. The distribution of the category entropy is similar in both situations. However, for a certain entropy, the number of publishers with downvotes is less than the number of publishers without them. This indicates people with downvotes are having a higher entropy in general. As shown in Fig.4(b), the percentage of people is always lower for publishers with upvotes than for those without them.
Compared with the “downvoting” situation, the entropy for “upvoting” is more diverse.

The “downvoting” situation and the “upvoting” situation can both be explained with the fact that people prefer those who like to travel around or share different things of life and may be attracted to give their comments. Besides, people who are influential on the social platform are more likely to make controversial remarks, thus receiving more upvotes and downvotes. To understand the difference between “downvoting” and “upvoting”, we notice that compared with giving upvotes, publishers are more cautious when giving downvotes, so the range of people being downvoted are smaller than the one of those with upvotes, which means a greater difference with general situation in entropy distribution. Therefore, as can be seen in Fig. 3 and Fig. 4, the two entropy metrics have a greater effect on “downvoting”.

### 3.3 Effective Size

Another newly-added feature is effective size. By adding effective size originated from the structural hole theory [19], we add another aspect to our feature list: graph structure. As graph structure is relevant to the information inflow and outflow to certain tips, it is likely to be useful for predictions. The effective size is a quantitative measure of the non-redundancy size of an individual’s ego network. Defined by Burt [4], the effective size is used as one of the four measurements to identify structural hole spanners (SHS) in a social network. In social networks, the connections between people form small groups in the large community. According to the structural hole theory, the unconnected parts between the groups form structural holes, and individuals who occupy the holes are defined as SHS. Being an SHS, an individual has advantages of both information and control. Information advantage means that one can access information from different sources, and because of those distinct information, one can be more influential between groups. Thus, it is meaningful to examine the upvotes and downvotes of SHS and non-SHS publishers.
There are four measurements given by Burt to identify SHS: effective size, constraint, efficiency, and hierarchy. As in [16, 29], we choose effective size as the structural hole feature in our model. The effective size can be calculated by subtracting the total redundancy from an individual’s ego-network. The redundancy of node $i$ related to $j$ is defined by

$$ R = p_{iq} - m_{jq}. $$  

(3)

Here, $p_{iq}$ is defined by the proportion of energy that $i$ has spend in relationship with $q$, and $m_{jq}$ is given by $j$’s interaction with $q$ over $j$’s strongest relationship among all the users.

Then, we can calculate the total redundancy of $i$’s ego-network by summing up all of $q$’s redundancy in the network.

Thus, the definition of effective size is as follows:

$$ \sum_j \left[ 1 - \sum_q p_{iq}m_{jq} \right], q \neq i, j \quad (4) $$

There is also a simpler approach given by Borgatti [2], in which the redundancy is given by

$$ R = \frac{2t}{n}, $$

(5)

where $t$ is the number of the total non-ego ties in the ego-network, and $n$ is the number of total non-ego nodes. Thus, we have the equation for effective size as follows:

$$ n \frac{2t}{n} $$

(6)

We build the ego-network graphs of 1.6 million users from Foursquare, and calculate the effective size of each of them using EasyGraph3. Then, as it is shown in Fig. 5, we calculate the cumulative distribution function (CDF) of the effective size of publishers with and without upvotes, and with and without downvotes from 10,000 random samples.

Figure 5(a) shows the CDF of “downvoting” effective size of the Foursquare publishers, and Fig. 5(b) shows the CDF of “upvoting” effective size. In Fig. 5(a), under the same cumulative probability, the “downvoting” effective size is larger than the “upvoting” effective size. For example, 80% of publishers without downvotes have the effective size around 90, and those with downvotes have the effective size around 120. To explain the result, we may refer to the definition of effective size. As we mentioned above, effective size measures the redundancy of an ego network, and higher effective size represents lower redundancy. Publishers with high effective size are more likely to receive and release more non-redundant information, and be more influential. Thus, they are more likely to get more attention, and receive downvotes. Similarly in Fig. 5(b), the effective size of publishers with upvotes is higher than the effective size of publishers without

3 https://easy-graph.github.io/.
upvotes. We also notice that the difference is larger in Fig. 5(b) than in Fig. 5(a). 80% of publishers with and without upvotes have the effective size of 200 and 90, compared to the effective size of 120 and 90 in “downvoting”. This can be explained by the fact that publishers with more non-redundant information are more likely to post more valuable and recognized content than controversial content. Thus, the impact of effective size in upvoting is more significant than downvoting.

3.4 Model Construction

Our downvote and upvote prediction have similar feature lists and models. However, there are some subtle differences between them, so we firstly discuss the process of downvote prediction. As discussed in Sect. 2, we extracted and applied 52 features. In our study, we randomly select 10,000 tips as our training set and 2,000 tips as test set. We restrict that one publisher can only have at most one tip in the training or test set. Also, tips with 0 downvote are recognized as negative instances and tips with 1 or more downvotes are recognized as positive instances. Thus, we convert this predicting process into a binary classification problem.

Then, we predict upvotes using the same model constructing methods, except for some adjustments of features and classifying method: (1) we remove the total number of “likes” received by previous tips of the publisher, assuming that previous upvoting number would have a strong correlation with the incoming one, and thus should not be involved in this model; (2) we reset the boundary of positive and negative outputs to 5, which means only tips with 5 or more upvotes are recognized as positive instances, and others are regarded as negative.

Both downvote and upvote prediction leverages a supervised machine learning-based classifier. To begin with, we feed selected data to prevailing machine learning algorithms, including Decision Tree (DT) [24], Support Vector Machine (SVM) [12], Naive Bayes (NB) [9], Random Forest (RF) [3], and XGBoost [5] algorithms. To achieve the best performance, we apply a grid search method to every algorithm to find the best fitting parameters, as shown in Table 1.

4 Evaluation of Prediction Performance

4.1 Performance Evaluation

In this subsection, we use four classical metrics to evaluate the classifier, i.e., precision, recall, F1-score, and AUC. To define these four metrics, we use 4 parameters TP, FP, TN, and FN, representing true positive, false positive, true negative, and false negative respectively. Besides AUC, whose meaning is the area under the ROC curve, the definitions are as follows:

\[
Precision = \frac{TP}{TP + FP}
\]
Recall = \frac{TP}{TP + FN} \quad (8)

F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (9)

Since our sampled dataset is well balanced, we use macro averaged precision, recall, and F1 score to evaluate the models. Thus, all the precision, recall, and F1 score mentioned in this paper are the macro averaged values of the two classes. As shown in Table 1, XGBoost achieves the best performance in downvote prediction, and it has an F1-score of 0.96 and an AUC of 0.99. Also, we notice that all three tree-based machine learning models perform well, whereas other models like SVM and NB show unsatisfactory performances. In Foursquare, the vast majority of tips do not have downvote at all, and other tips often have only 1 downvote, leading us to generate the previous idea that downvote number is nearly unpredictable at Foursquare because limited downvoting quantity means a higher stake of randomness. Nevertheless, our predicting result is encouraging that we actually can still predict the downvoting number with high credibility.

As shown in Table 2, XGBoost keeps performing the best out of 6 machine learning models in upvote prediction, with an AUC score of 0.98. Compared to downvote prediction, tree-based models (XGBoost, RF, and DT) keep showing their robustness, while other models (SVMr, SVMp, and NB) improve significantly both in precision and recall. This finding is likely due to the fact that upvoted tips are much more commonplace than downvoted tips, so the linkage between inputs and outputs is more straightforward for these rudimentary models to capture and then classify. The SVM model, with novel features added, has a precision of 0.74, a recall of 0.73, and an AUC value of 0.81, which significantly outperforms the current prediction method with a precision of about 0.5 [27].

Table 1. Prediction of Downvotes

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>XGBoost</td>
<td>n_estimators = 50, learning_rate = 0.3, max_depth = 8, subsample = 1</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.99</td>
</tr>
<tr>
<td>RF</td>
<td>n_estimators = 50, Max depth = 11</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.97</td>
</tr>
<tr>
<td>DT</td>
<td>Min samples leaf = 1, Max depth = 5</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
<td>0.97</td>
</tr>
<tr>
<td>SVMr</td>
<td>degree = 1, gamma = 0.0001, C = 100</td>
<td>0.70</td>
<td>0.70</td>
<td>0.70</td>
<td>0.77</td>
</tr>
<tr>
<td>NB</td>
<td>-</td>
<td>0.70</td>
<td>0.61</td>
<td>0.57</td>
<td>0.74</td>
</tr>
</tbody>
</table>

4.2 Feature Importance Analysis

To tell how different kinds of features contribute to the final prediction, we manage to remove features related to upvotes (not necessary in upvote prediction), user, venue, and content, separately, and then run the machine learning model again. Here, we use DT model to evaluate the importance of different feature categories, as shown in Table 3 and Table 4.
In our selected features, there are 8 features that is about upvotes, i.e., (1) total number, median, average and standard deviation of upvotes received by previous tips of the publisher; (2) total number, median, average and standard deviation of upvotes received by tips posted at the venue. These features are worth studying because they possibly reveal the correlation between upvote and downvote. Therefore, Table 3 manifests the predicting results after ignoring these features, as well as removing features of user, venue, and content, separately. The AUC score using complete dataset and DT is 0.97, and it decreases the most to 0.77 after ignoring features of venue, suggesting that venue-related features are the most important to downvote prediction. If removing upvoting features, usage data features, and tip context features separately, the AUC would be 0.87, 0.95, and 0.90, indicating that these features are useful but not critical. In Table 4, with AUC score dropping from 0.95 to 0.77 after removing venue-related features, venue continues to demonstrate its position as the most influential fac-

Table 2. Prediction of Upvotes

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>XGBoost</td>
<td>n_estimators = 100, learning_rate = 0.3, max_depth = 10, subsample = 1</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.98</td>
</tr>
<tr>
<td>RF</td>
<td>n_estimators = 100, Max depth = 11</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
<td>0.97</td>
</tr>
<tr>
<td>DT</td>
<td>Min samples leaf = 1, Max depth = 8</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>0.95</td>
</tr>
<tr>
<td>SVMr</td>
<td>degree = 1, gamma = 0.0001, C = 50</td>
<td>0.74</td>
<td>0.73</td>
<td>0.73</td>
<td>0.81</td>
</tr>
<tr>
<td>NB</td>
<td>–</td>
<td>0.73</td>
<td>0.69</td>
<td>0.67</td>
<td>0.79</td>
</tr>
</tbody>
</table>

Table 3. Prediction of Downvotes after Removing Certain Features

<table>
<thead>
<tr>
<th>Features</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
<td>0.97</td>
</tr>
<tr>
<td>w/o like</td>
<td>0.86</td>
<td>0.84</td>
<td>0.84</td>
<td>0.87</td>
</tr>
<tr>
<td>w/o user</td>
<td>0.89</td>
<td>0.88</td>
<td>0.88</td>
<td>0.95</td>
</tr>
<tr>
<td>w/o venue</td>
<td>0.71</td>
<td>0.70</td>
<td>0.70</td>
<td>0.77</td>
</tr>
<tr>
<td>w/o content</td>
<td>0.86</td>
<td>0.84</td>
<td>0.84</td>
<td>0.90</td>
</tr>
</tbody>
</table>

Table 4. Prediction of Upvotes after Removing Certain Features

<table>
<thead>
<tr>
<th>Features</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>0.95</td>
</tr>
<tr>
<td>w/o user</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>0.95</td>
</tr>
<tr>
<td>w/o venue</td>
<td>0.70</td>
<td>0.70</td>
<td>0.70</td>
<td>0.77</td>
</tr>
<tr>
<td>w/o content</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>0.95</td>
</tr>
</tbody>
</table>
Fig. 6. The features with highest SHAP values: (a) features with top 5 highest SHAP values in upvote prediction; (b) features with top 10 highest SHAP values in downvote prediction.

(a) SHAP value of “downvoting” DT model

(b) SHAP value of “upvoting” DT model

Fig. 6. The features with highest SHAP values: (a) features with top 5 highest SHAP values in upvote prediction; (b) features with top 10 highest SHAP values in downvote prediction.
responding our previous results that venue-related features play a critical role in
downvote prediction. In Fig. 6(b), compared with downvote prediction, upvote
prediction is more of an output generated from mixed features and contributed
by them more evenly. Specifically, top 10 influential features include country
entropy, number of “likes” received by tips posted at the venue, and some fea-
tures about network structure. It is worth mentioning that the new features we
add, i.e., entropy and effective size, are validated to contain wealth of informa-
tion, and play significant roles in our model according to SHAP value. Also, we
notice that there are 125 features used in the baseline approach [27]. In contrast,
we only use 52 features and achieve a better performance. This also confirms
that our newly-added features are more effective.

5 Related Work

Previous works on upvotes and tip popularity prediction adopt different meth-
ods, including machine learning methods and methods combined with NLP
features. Vasconcelos et al. [27] studied the upvotes of Foursquare from three
aspects. They crawled data of 13 million users. First, they predicted the popu-
larity of the tip at the time it was posted, using all previous data. Second, they
predicted the popularity evolution of tips in a certain period after it was posted.
Third, they built and evaluated models on different specializations including city-
based and category-based models. On predicting upvotes of a tip at the time it
was posted, they set the boundary of non-upvote to lower than 5 upvotes, which
is the same as our setting. Under such setups, they reached a satisfying recall of
about 0.8, but the precision of around 0.5 remains to be improved.

Kasper et al. [15] conducted research on key features of review helpfulness
on Metacritic, a video game reviews website. By defining helpfulness of a review
as its upvotes divided by the sum of upvotes and downvotes, they found out
that helpfulness is correlated with the score given by the review. Then, they
built prediction models for helpfulness, and achieved up to the F1 score of 0.64,
using only text-related features. Scellato and Mascolo [25] analyzed the pattern
of user activities of an LBSN. They showed that the number of check-ins and
the number of places per publisher followed a log-normal distribution. Also,
they showed that due to the difference between the system’s restrictions on
checking-in new locations and adding new friends, the distributions of friends
and check-ins/places varied.

There are studies on predicting upvotes on LBSNs studying different plat-
forms. Li et al. [17] did a research on predicting answer quality on ResearchGate
(RG). They used two groups of characteristics: those could be obtained directly
from the web content, and those were generated by post-processing. They sam-
ped 1128 posts in 107 question threads, used Naive Bayes, SVM and multiple
regression models for prediction, and reached a highest accuracy of 0.62. Segall
and Zamoshchin [26] conducted a study on predicting Reddit post popularity.
Their dataset contains a random sampling of 2 million posts from Reddit. They
used Naive Bayes, SVM, and linear regression to predict, and used an NLP approach called stemming to decrease the feature set. Overall, ResearchGate and Reddit both provide a downvote option, yet none of the previous work makes predictions about it.

6 Conclusion

In this paper, we undertake predictions of Foursquare tips’ upvotes (outperforms previous results of the baseline [27]) and downvotes (the first prediction on OSN’s downvotes to our acknowledgement) and on our selected dataset of all crawled 1.6 million Foursquare users’ data. Firstly, we establish a comprehensive feature list and conduct analysis about key features. In addition, to build a supervised machine learning-based prediction, we further introduce novel features such as geographical and categorical entropy (information theory), effective size (structural hole theory), and NLP features of tips. Furthermore, we conduct a data-driven analysis on our model and we can see satisfying outcomes. Our XGBoost model shows an AUC value of 0.99 in downvote prediction and 0.98 in upvote prediction. Given the fact that our prediction is quite accurate, social networking service providers can leverage it to improve user satisfaction by adjusting recommending weight dynamically and promptly. Lastly, we dive deeper to calculate the contribution of different features by removing some of them and using SHAP value. It turns out that the venue-related features of the tip and the publisher are the most influential to upvotes and downvotes, whereas the tip content itself and the publisher’s social network are not as crucial as we thought. This is largely because Foursquare is an LBSN, and publishers’ information flow is mostly decided by venues. Our findings are helpful to both LBSN service providers and users. Besides, Foursquare is a place for people to comment on city venues, and therefore we believe that our work can be extended to a wider range of comment-based OSNs, such as hotel evaluation like Booking.com, answer evaluation like Quora, and movie evaluation like Rotten Tomatoes.

Acknowledgments. This work is sponsored by National Natural Science Foundation of China (No. 62072115, No. 71731004, No. 61602122). Jinyan Zhu and Tianzheng Meng have equal contribution.

References


