Predicting Twitter Hashtags Popularity Level

Shing H. Doong
Department of Information Management
ShuTe University, Taiwan
tungsh@stu.edu.tw

Abstract
This paper investigates the problem of predicting Twitter hashtags popularity level. A data set of more than 18 million tweets containing 748 thousand hashtags has been prepared by using Twitter’s Streaming API. Early adoption properties including profile of tweet authors and adoption time series are used to predict a tag’s later popularity level. The followers count and tweets count are two such characteristics related to adopters’ profile. On the other hand, two types of frequency domain analyses are used to augment the simple mean and standard deviation characteristics of the adoption time series. Fourier transform (FT) spectrum and wavelet transform (WT) spectrum are considered in this study. Experimental results show that WT spectrum improves the prediction result of viral hashtags while FT spectrum does not.

1. Introduction

According to Merriam-Webster’s collegiate dictionary, a meme is “an idea, behavior, style or usage that spreads from person to person within a culture.” Studying how memes diffuse is of great interests to sociologists. However, it is difficult to analyze the diffusion of memes in physical world due to the difficulties of data collection. Social media using today’s information and communication technology (ICT) make Internet memes tracking and analysis possible.

Web 2.0 has created a distributed structure of Internet contents that are produced and consumed by today’s prosumers. By meeting the love and belonging level of Maslow’s hierarchy of needs with easy-to-use ICT, online social networks have blossomed in the last decade. Twitter as one of the major microblogging services has more than 500 million registered users who send more than 500 million tweets each day (Wikipedia). A tweet (status) is a short message containing no more than 140 characters. Users can use diversified devices from desktop to mobile to send and receive tweets. Being “the SMS of the Internet”, Twitter has impacted the world in many respects including the Arab Spring in 2011.

Users can embed topical items called hashtags in tweets. A hashtag is a single word without space following the hash sign #. Some hashtags are lexically meaningful while some are not. Each day, hashtags are created by users, diffused with tweets and mutated by new idea. Hashtags compete with themselves for user’s attention; some die immediately after creation, some survive for a longer time and may become viral at some point.

According to Twitter’s own research, tweets with at least one hashtag receive 2 times more engagements than tweets with no hashtags. Here engagements include clicks, retweets, replies and favorites. Users may increase their network influence with popular tweets. Marketers may improve their marketing efforts by adopting popular hashtags. Detecting viral hashtags at their early age has practical applications for marketers.

In this study, we investigate the problem of predicting Twitter hashtags popularity level. Viral hashtags are rare, but they may offer advantages to marketers. If a methodology can predict the popularity level of hashtags by using their early adoption characteristics only, marketers can use this technique to try out competing hashtags and invest marketing resources in those predicted viral ones.

Content of the tag as well as context of tweets containing the tag have been used to predict the tag’s popularity [1][2]. Additionally, network properties of tag adopters have also been used [3], because Twitter’s follow network provides a conduit to propagate memes. Other types of tag characteristics that has been used before include the adoption time series of a hashtag [4], because differences between two successive adoption times prevail the diffusion speed of a tag.

We use Twitter’s Streaming API to collect public tweets. According to Twitter, the Streaming API may sample up to 1% of all available public tweets [5]. Our research goal is to predict the later popularity level of a hashtag by using its early adoption properties only. By early, we mean the earliest few tweets of a hashtag.
Our properties include the number of different authors creating these tweets, various profiles of these authors, the number of mentions and hyperlinks in early tweets, and adoption time series of these tweets.

Previous studies often use simple statistics like the mean and standard deviation to describe the adoption time series [4]. We notice that these simple statistics cannot catch the oscillation properties of a time series, since they are unchanged no matter how we rearrange the time series. In order to get a good understanding of the oscillation, frequency domain analysis is used to analyze the time series. We adopt Fourier transform (FT) to compute the Fourier spectrum [6] of the time series. In addition, we also use wavelet transform (WT) to decompose the adoption time series into time-frequency domain and compute the wavelet spectrum [7].

The research objectives of this study are whether these spectral properties may help predict the popularity level of hashtags, and if so, which mechanism (FT or WT) provides a better solution. We will put more attention on the prediction of viral hashtags which are scarce but of higher stakes in applications.

This paper is organized as follows. Section 2 is devoted to a literature review on hashtag popularity prediction and frequency domain analysis. We describe our methodology and data set in section 3. Section 4 presents the experimental results and discussions. We conclude the paper with remarks in section 5.

2. Literature review

Twitter, the world’s largest microblogging service offers both weblog function and social network function. Though Kwak et al. questioned the social network characteristics of Twitter [8], Twitter by far is one of the most active sites that have social links built-in. By following a user, tweets created or retweeted by the follower will automatically show up in the follower’s home timeline. This follow network is a directed network and has made Twitter different from many other social networks such as Facebook or LinkedIn. Kwak et al. found that most relationships between users of Twitter are asymmetric and public media companies or celebrities use Twitter as a convenient channel to disseminate news to their followers [8]. A tweet can contain hashtags like #fifa to indicate user-defined topics. Authors can also mention other accounts by using the mention sign such as @worldbank.

2.1. Hashtags popularity prediction

Like other data mining tasks, predicting the popularity of a hashtag needs to choose a suitable set of predictors in the first step. Unlike many data mining problems where records are often retrieved from a relational database system, various features of different forms can be extracted from a hashtag, e.g. content based, network based or time series based features.

The inherent content of a hashtag is considered to be an important factor for its popularity. If a tag can evoke viewers’ emotion, it is more likely to be imitated and spreads into a large population. Tsur and Rappoport used content based features such as the number of words contained in the tag, lexical items and emotional characteristics to study the spread of memes in Twitter [2]. In addition to the hashtag itself, contextual features from the tweets have also been used to predict the popularity of newly emerging hashtags. For example, Ma et al. used fraction of tweets containing URL, fraction of tweets containing mention @, and sentiment of tweets as content based features in their study [1]. Suh et al. found that the numbers of URLs and hashtags in a tweet are strongly correlated to the retweetability of the tweet [9].

Through the follow network, Twitter provides a convenient channel to disseminate information quickly. A basic assumption in the social influence theory is influential nodes are more likely to spread messages successfully, though there are many definitions of the influential capacity. The in-degree count in the follow network is arguably the simplest indicator to measure influential capacity. Twitter users with a large population of followers enjoy the advantage of exposing their ideas to a large audience immediately, though some research indicates that the followers count property may not be a good predictor [10]. Pal and Counts used user’s activity such as the number of statuses to identify influential users [11]. Weng et al. considered the community structure of Twitter sociogram in their prediction task [3][4]. We ignore community structure in this study because of two reasons: the rate limit policy of Twitter API has made building the social graph very difficult, and the Twitter API does not report when a follow relationship is created, i.e. before or after a hashtag is imitated through the social influence.

Research has shown that early popularity of a hashtag is closely related to its later popularity, and viral tags are expected to spread more quickly than others [12][13]. Weng et al. used the early adoption time series to predict the popularity of a hashtag [4]. However, they only considered simple statistics like mean and standard deviation of the differential series derived from the adoption series. It is well known that these simple statistics cannot describe the oscillation properties of a time series. For example, the two series
in Figure 1 have the same mean and standard deviation, but they have quite different oscillation properties. Therefore, we incorporate frequency domain analysis to analyze the oscillation properties of the differential adoption time series.

Our prediction task in this study is to predict the later popularity level of a hashtag by using its early adopting tweets. Contextual properties of tweets, profiles of authors, and adoption time series are extracted from these early tweets as predictors. The later popularity level of a tag is judged by how many tweets have adopted the tag by the end of our data collection period.

![Figure 1: Two time series with the same mean and standard deviation.](image)

2.2. Frequency domain analysis

The most popular frequency domain analysis tool is given by FT. Given a time series \( x_n, n = 0, 1, \ldots, N-1 \), the discrete FT is defined as follows [6]

\[
X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}, \quad k = 0, 1, \ldots, N-1
\]  

(1)

In the above formula, Fourier spectrum is indexed by \( k \) and amplitude \( |X_k| \) denotes the strength of the corresponding frequency.

Being a frequency domain method, FT fails to capture the local oscillation properties of a time series. WT solves this problem by using multiple resolutions of the time variable to decompose a series [7]. WT has been used in many meteorological data analysis problems.

Let \( \psi(n) \) denote a mother wavelet. The continuous WT of the series \( x_n \) with a scale \( s \) is given by the formula

\[
W_m(s) = \sum_{n=0}^{N-1} x_n \psi^* \left( \frac{(n-m)s}{s} \right), \quad m = 0, 1, \ldots, N-1
\]  

(2)

In the above formula, \( * \) indicates the complex conjugate and \( s \) is the time difference between two successive events. In our experiments, we use the Mexican hat wavelet, which is also called the derivate of Gaussian (DOG) wavelet because it is the second derivative of the Gaussian function. The Mexican hat wavelet is given by the formula

\[
\psi(n) = \frac{1}{\sqrt{2\pi}} (1-n^2) e^{-n^2/2}
\]  

(3)

By varying the scale \( s \), we obtain a picture of the time series at different resolutions. For the DOG wavelet, the corresponding Fourier wave length is approximately equal to \( 4s \), thus the frequency is about \( 1/(4s) \) [7].

3. Methodology

In this section, we describe our data collection method and basic statistics of the collected data. Feature selection is an important task for any prediction problem. We describe features used in our experiments. Finally, the experimental procedure including prediction algorithms is explained.

3.1. Data collection

Twitter has released two types of APIs (REST and Streaming) allowing authenticated users to collect or manipulate tweet data [5]. The REST API provides programming interfaces to read and write Twitter data, author a new tweet, read author profile and follower data. The Streaming API gives developers low latency access to Twitter’s global stream of public tweets. In order to avoid a heavy burden on the service, Twitter implements a rate limit policy on most REST APIs. For example, the GET friendships/show API allows an authenticated user account to retrieve detailed information about the relationship between two
arbitrary users, and this API allows a maximum of 180 calls in a 15-minute window according to the policy.

In contrast to the REST API, most Streaming APIs do not use a similar policy to restrict the access to public streams. The GET statuses/sample API allows a program to retrieve a small random sample of all public statuses. Tweets returned by the default access level are the same, i.e., two different users connecting to this endpoint will see the same tweets. According to Twitter, this API can return up to a maximum of 1% public tweets that are currently being created [5]. On the other hand, the GET statuses/firehose API returns all public tweets and requires a special permission to access.

We use the GET statuses/sample Streaming API to collect a small sample of public tweets between May 13, 2005 and June 2, 20015. After excluding non-English based tweets, we ended up with more than 18 million tweets. Each tweet is listed in one line starting with the screen name followed by user id, timestamp, status id and the tweet context. A tweet may contain RT (indicating a retweet), mentions, hyperlinks or hashtags. The following is an annotated sample.

@MAUREEN_WHITE_ (screen name), 3225914647 (user id), 1433062773000 (time stamp), 604935234283372546 (status id) - RT @Cynthiapoet: Sunset Leopardess by Mark Dumbleton #WeAreAlive #Animals #Photography http://t.co/mu3q6UBk3t (tweet context)

### 3.2. Feature extraction

After collecting the tweets, we wrote a program to extract user id, time stamp, status id and hashtags from each tweet. In addition, we also counted how many mentions and hyperlinks are contained in a tweet. The processed raw data have 6 fields (user id, timestamp, status id, mention count, hyperlink count, tag) and 6.87 million records. All these data are stored in a MySQL database for further processing.

After grouping records by using the tag field and ignoring case, we found 748237 different hashtags in our collected tweets. In order to conduct the experiments, we delete tags supported by 199 or fewer tweets and tags with one single character. This left us with 3331 tags with a tweet support count ranging from 200 to 391553 (associated with #3queens). Since viral tags are rare, the tweet support count distribution of these remaining tags is seriously skewed towards the lower end. Table 1 lists the count distribution of the tags.

<table>
<thead>
<tr>
<th>Support (cnt)</th>
<th>Floor</th>
<th># of tags</th>
<th>Popularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>200–999</td>
<td>2</td>
<td>2701</td>
<td>Low (1)</td>
</tr>
<tr>
<td>1000–9999</td>
<td>3</td>
<td>581</td>
<td>Medium (2)</td>
</tr>
<tr>
<td>10000–99999</td>
<td>4</td>
<td>48</td>
<td>High (3)</td>
</tr>
<tr>
<td>100000 and up</td>
<td>5</td>
<td>1</td>
<td>High (3)</td>
</tr>
</tbody>
</table>

Floor is the largest integer less than or equal to log10(cnt)

Because the last category contains only one tag (#3queens), we combine this category with the previous one to form a high popularity class 3 of 49 tags. The medium popularity class has 581 tags, and the low popularity class contains 2701 tags. This popularity level (1-3) will be our target variable in the task of popularity prediction.

We use early adoption properties of a tag to prepare the predictor variables. By using timestamps of tweets, we can extract the earliest \( n = \{25, 50, 100, 150 \text{ and } 200 \} \) in our later experiments) tweets that contain the hashtag. Let \( A \) denote the author set of these \( n \) tweets. Since a user may use the same tag in different tweets, the cardinality of \( A \) (denoted as \( na \)) is less than or equal to \( n \). If \( na \) is relatively small compared to \( n \), then those early tweets are authored by a small population. On the other hand, if \( na \) is close to \( n \), then the diversity of early adopters may promote the spread of a hashtag into a wider neighborhood. The number of early adopters has been used in [4] and is one of our predictors.

The next two predictors are \( cm \) and \( ch \) which respectively represents the total number of mentions and hyperlinks (http or https) in all early tweets containing the tag. These two variables represent the contextual properties from tweets. Previous research has indicated that tweets containing mentions and/or links may increase the attention of readers, and thus enhance the exposure rate of the tag [1][11].

Using the GET users/lookup REST API, we obtain detailed information about early adopters, including their followers count, statuses count and listed count. The followers count has been used in [10], while the statuses count has been used in [1][11]. The listed count is the number that a user has been added to his/her followers’ lists. Twitter lists allow followers to organize their followees into groups of different interests. It can be argued that the listed count has the same or more influential effect than the followers count since followers need to take time to manage their lists. The average followers count (cf), average statuses count (cs) and average listed count (cl) of the early adopters of a tag are used as predictors in the prediction task.

The next set of variables comes from the timestamps of early tweets. Let \( t_1, t_2, ..., t_n \) represent...
the adoption time of these $n$ tweets. A differential series may be derived from this time series by considering differences between two successive adoptions: $\delta_i = t_i - t_{i-1}, i = 2, 3, ..., n$. The differential series is nonnegative since we have ordered tweets according to their timestamps. However, different oscillation patterns of the series may indicate different diffusion patterns of the tag. For example, an increasing differential series indicates that it takes more and more time to spread a tag into the next tweet. Thus, the popularity of the tag may be diminishing. One the other hand, a decreasing differential series indicates the tag is receiving tweet support faster and faster and the popularity of the tag may be rising. In normal cases, the series is not totally increasing or decreasing, thus diffusion speed of the tag may speed up or slow down from time to time. Due to this reason, we should consider the oscillation properties of the differential series.

Previous research using the differential series considers the mean ($mu$) and standard deviation ($sd$) statistics only [4]. We use frequency domain analysis to capture the oscillation properties of differential series.

Using equation (1), we can calculate the Fourier spectrum $|X_k|$, which is symmetric about the central index. To avoid duplicate spectrum, we use the first half spectra only. To make the prediction task easier, we split the frequency domain into 10 regions and compute the total amplitude in each region to extract 10 spectrum properties $s_1, s_2, ..., s_{10}$. For example, when 25 early tweets are used in the experiments, $s_1$ denotes $|X_1|$, ..., $s_8$ for $|X_8|$, $s_9$ for $|X_9| - |X_{10}|$, and $s_{10}$ for $|X_{11}| + |X_{12}|$.

Using equation (2), we compute the WT of differential series with the DOG wavelet of equation (3). For each scale, we sum up amplitudes over the entire time domain to get a marginal spectrum. After choosing proper scales, we manage to obtain instantaneous frequencies between 0 and .5. Similar to the FFT approach, we divide this frequency domain into 10 even regions (0~.05, .05~.1, ..., .45~.5) and accumulate marginal spectrum over those 10 regions. Let $s_1, s_2, ..., s_{10}$ denote the WT spectrum properties. Then, $s_1$ represents the total marginal spectrum for frequencies between 0 and .05, and the other $s_i$ variables are interpreted similarly. The various variables used in the prediction are summarized in Table 2.

<p>| Table 2. Variables used in the study. |</p>
<table>
<thead>
<tr>
<th>Variable</th>
<th>Role</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>na</td>
<td>Input</td>
<td>Number of early adopters</td>
</tr>
<tr>
<td>cm</td>
<td>Input</td>
<td>Total count of mentions</td>
</tr>
<tr>
<td>ch</td>
<td>Input</td>
<td>Total count of hyperlinks</td>
</tr>
<tr>
<td>cf</td>
<td>Input</td>
<td>Average followers count</td>
</tr>
<tr>
<td>cs</td>
<td>Input</td>
<td>Average statuses count</td>
</tr>
<tr>
<td>cl</td>
<td>Input</td>
<td>Average listed count</td>
</tr>
<tr>
<td>mu</td>
<td>Input</td>
<td>Mean of differential series</td>
</tr>
<tr>
<td>sd</td>
<td>Input</td>
<td>Standard deviation of differential series</td>
</tr>
<tr>
<td>$s_{1-10}$</td>
<td>Input</td>
<td>FT or WT spectrum</td>
</tr>
<tr>
<td>cla</td>
<td>Output</td>
<td>Popularity level from Table 1</td>
</tr>
</tbody>
</table>

3.3. Experimental procedure

After extracting features from the collected tweets, we have a table of 3331 records with 19 fields in each record. The $cla$ variable is the output variable, while the other 18 variables are the predictor variables. The random forest (RF) algorithm [14] is used as the classifier algorithm.

RF is an ensemble classification algorithm that has been used in many data mining problems. Being a bagging algorithm, RF creates multiple decision trees in the training stage and aggregates decisions from these trees to make a final prediction in the operational stage [14]. Each decision tree is trained with cases sampled with replacements from the original training set. At a decision node, RF chooses a random subset of predictors and picks the best one from this subset for the node. By using multiple trees in the operational stage, the problem of over-fitted trees can be avoided.

For each popularity level, we measure the performance of prediction in three perspectives: precision ($p$), recall ($r$) and $F_1$ score. In classification problems, precision is the percentage of predicted samples that are actually relevant, while recall is the percentage of relevant samples that are predicted by the classification algorithm. The $F_1$ score combines both precision and recall in a simple formula in equation (4), and it is between 0 and 1 with a higher score indicating a better prediction result.

$$F_1 = 2 pr / (p + r)$$

Accuracy of a prediction model is the ratio of correctly predicted cases to total test cases. Accuracy can also be defined as the weighted sum of recall rates from all classes.

4. Experimental results and discussions

A naive model is given by predictions using the probability distribution of each class. According to
Table 1, class 1, 2 and 3 appears with a probability of 0.811, 0.174 and 0.015 respectively. Thus, when a case is presented, it is predicted to be with the class of 1, 2 and 3 with the corresponding probability. Table 3 shows results for this naive model with inferior outcomes for class 3, which is often of high stakes in applications. In the following, we will compare results based on three sets of predictors. Model 1 uses basic variables (na, cm, ch, cf, cs, cl, mu, sd) only, Model 2 uses basic variables plus 10 FT spectrum, and Model 3 uses basic variables plus 10 WT spectrum.

### Table 3. Naive prediction.

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>Class</th>
<th>Weight</th>
<th>p</th>
<th>r</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>.702</td>
<td>1</td>
<td>.811</td>
<td>.814</td>
<td>.830</td>
<td>.822</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>.174</td>
<td>.185</td>
<td>.167</td>
<td>.176</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>.015</td>
<td>.018</td>
<td>.020</td>
<td>.019</td>
</tr>
<tr>
<td></td>
<td>W Avg</td>
<td></td>
<td>.693</td>
<td>.702</td>
<td>.698</td>
</tr>
</tbody>
</table>

W Avg: weighted average

We set $n=25, 50, 100, 150$ and $200$ to extract $n$ earliest tweets containing a hashtag. Each prediction task is conducted in a 10-fold cross validation style. In order to minimize the effect of partition randomization in cross validation, 10 runs of 10-fold cross validation were performed for each experimental scenario (5 early tweet sizes x 3 models of predictors). Regarding the classifier, RF is used in a setting of 300 decision trees and 5 random features.

### 4.1. Prediction results

The first assessment is based on the accuracy indicator. Since 10 runs of 10-fold cross validation were conducted, averaged accuracy rates are computed and reported in Table 4. For each model of predictors, as the number of tweets increases, so does the accuracy rate. For a fixed number of tweets, the difference between any two models is minimal (< 1%). Because accuracy is the weighted recall rate, class 1 (low popularity) has the dominant effect on the final accuracy. On the other hand, class 3 (high popularity) has the lowest weight to impact the overall accuracy, but it usually has high stakes in practical applications. We turn our attention to the result of this class.

### Table 4. Overall accuracy.

<table>
<thead>
<tr>
<th></th>
<th>25</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>.878</td>
<td>.886</td>
<td>.893</td>
<td>.898</td>
<td>.900</td>
</tr>
<tr>
<td>Model 2</td>
<td>.880</td>
<td>.886</td>
<td>.893</td>
<td>.895</td>
<td>.897</td>
</tr>
<tr>
<td>Model 3</td>
<td>.878</td>
<td>.886</td>
<td>.896</td>
<td>.897</td>
<td>.900</td>
</tr>
</tbody>
</table>

Model 1: basic, Model 2: basic + FT, Model 3: basic + WT

Tables 5 and 6 show the averaged precision and recall rate of class 3 cases respectively. The precision rate is significantly improved when FT or WT spectrum is added to the basic variables. The biggest improvement (11.6%) appears when 25 tweets are used and WT spectrum is added to the predictor set. Generally speaking, as more early tweets are used, the more precise we can expect for class 3 prediction. The performance difference between model 1 and the other two models diminishes as more tweets are used, indicating decreasing return to oscillation properties (spectrum).

### Table 5. Precision for class 3.

<table>
<thead>
<tr>
<th></th>
<th>25</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>.516</td>
<td>.572</td>
<td>.599</td>
<td>.617</td>
<td>.665</td>
</tr>
<tr>
<td>Model 2</td>
<td>.630</td>
<td>.574</td>
<td>.648</td>
<td>.660</td>
<td>.701</td>
</tr>
<tr>
<td>Model 3</td>
<td>.632</td>
<td>.675</td>
<td>.676</td>
<td>.685</td>
<td>.707</td>
</tr>
</tbody>
</table>

Model 1: basic, Model 2: basic + FT, Model 3: basic + WT

In contrary to precision, the recall rate of class 3 prediction is not as rosy as expected. Model 1 shows a higher recall rate as more early tweets are used. The other two models present a non-monotonic pattern for the recall rate. At best, we can say that when 25, 50 or 100 tweets are used, WT spectrum provide advantages over the basic model in recall rate. On the other hand, FT spectrum does not help too much in recall rate.

### Table 6. Recall for class 3.

<table>
<thead>
<tr>
<th></th>
<th>25</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>.273</td>
<td>.278</td>
<td>.310</td>
<td>.310</td>
<td>.319</td>
</tr>
<tr>
<td>Model 2</td>
<td>.288</td>
<td>.278</td>
<td>.292</td>
<td>.265</td>
<td>.290</td>
</tr>
<tr>
<td>Model 3</td>
<td>.347</td>
<td>.321</td>
<td>.312</td>
<td>.278</td>
<td>.298</td>
</tr>
</tbody>
</table>

Model 1: basic, Model 2: basic + FT, Model 3: basic + WT

Combining precision and recall, we present the F1 score for class 3 prediction in Figure 2. As indicated by the precision and recall rates, Model 3 with WT spectrum outperforms the other two models when 25, 50 or 100 early tweets are used. Model 2 with FT spectrum outperforms the basic model by 3.8% when 25 tweets are adopted. The benefits of spectrum predictors seem to have a cap in terms of the number of early tweets. Overall, the best F1 score appears when Model 3 is used with 25 early tweets, resulting in a 9.1% improvement over the basic model.
4.2. Handling imbalanced data

It can be argued that class 3 is the most interesting class in practical applications. Unfortunately, this class has the lowest percentage (1.5%) in the collected data. Though RF uses many trees in aggregate to minimize prediction variance, it suffers from the curse of imbalanced data in the construction of prediction trees. The minority class may not have enough samples in the bootstrapped training data for tree learning.

One solution to handle the imbalanced data problem is to use different error weights for different classes in the training algorithm. For example, if a case of the minority class is incorrectly trained, the penalty will be higher than that of an incorrectly learned case from the majority class.

The other solution to handle the imbalanced data problem is to down-sample the majority class, up-sample the minority class or conduct both actions when bootstrapping training data for RF learning. For example, we can draw a sample from the minority class and randomly draw the same number of cases from the majority class [15]. Effectively, this method is to down-sample the majority class. Chen et al. [15] refer the weighted method as the Weighted RF (WRF) and the down-sampling method as the Balanced RF (BRF). In the following, we use BRF to consider the high popularity class prediction problem.

Since high popularity class is the main concern in the prediction problem, we combine the low and medium popularity classes into a single class and convert the multiple class problem to a binary class problem. The new data set has 3282 (98.5%) regular tags and 49 (1.5%) viral tags.

BRF is applied to the binary class problem with similar settings as above: 300 trees in a forest, 5 randomly selected variables to consider the split of a node, and 10 runs of 10-fold cross validation.

Table 7 shows the averaged accuracy for the new binary prediction problem with BRF. The accuracy rate is higher than the corresponding rate of the multiple class problem (Table 4). Though Model 3 has the highest accuracy in all tested tweet lengths, the difference between any two models is small (<1%).

<table>
<thead>
<tr>
<th>Model</th>
<th>25</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>.978</td>
<td>.981</td>
<td>.983</td>
<td>.984</td>
<td>.984</td>
</tr>
<tr>
<td>Model 2</td>
<td>.980</td>
<td>.982</td>
<td>.983</td>
<td>.984</td>
<td>.984</td>
</tr>
<tr>
<td>Model 3</td>
<td>.982</td>
<td>.984</td>
<td>.983</td>
<td>.984</td>
<td>.985</td>
</tr>
</tbody>
</table>

Model 1: basic, Model 2: basic + FT, Model 3: basic + WT

The averaged precision rate for the viral class is significantly improved (11.1%) from Model 1 to Model 3 with 25 tweets (Table 8). Except the case with 150 tweets, Model 3 beats the other two models in terms of viral class precision rate. This shows the merits of WT spectrum predictors.

Model 3 also provides the best recall rate for the viral class prediction in most tested tweet lengths (Table 9). The WT spectrum can improve the recall rate as high as 7.2% (100 tweets). By examining results from the multiple class problem (Tables 5 and 6), we observe that precision is generally lower but recall is generally higher in the binary class problem. This can be expected as BRF presented more minority cases to the learning algorithm, patterns for the viral class can be more well learned and more test cases are predicted to be viral.

<table>
<thead>
<tr>
<th>Model</th>
<th>25</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>.281</td>
<td>.354</td>
<td>.397</td>
<td>.451</td>
<td>.450</td>
</tr>
<tr>
<td>Model 2</td>
<td>.277</td>
<td>.363</td>
<td>.410</td>
<td>.465</td>
<td>.453</td>
</tr>
<tr>
<td>Model 3</td>
<td>.392</td>
<td>.461</td>
<td>.414</td>
<td>.448</td>
<td>.504</td>
</tr>
</tbody>
</table>

Model 1: basic, Model 2: basic + FT, Model 3: basic + WT
Figure 3 shows $F_1$ score for the viral class prediction. Unlike the multiple class problem, Model 3 beats the other two models in all scenarios. The lead over the basic model is most significant (~10%) when fewer tweets are used. With more and more early tweets, the advantages diminish. The overall best $F_1$ score appears when 200 tweets are used in Model 3. The performance of Model 2 is mostly inferior to the basic model.

4.3. Discussions

In the last section, we couple RF with different feature sets to predict viral hashtags. The experimental design uses two dimensions (5 tweet lengths and 3 models of predictors) to obtain 15 scenarios. These two dimensions allow us to examine the effect of feature sets from two perspectives: increasing the number of early tweets or adding FT/WT spectrum to the basic variables.

The assessment indicators include overall prediction accuracy and precision, recall and $F_1$ score of the high popularity class. Since 10 runs of 10-fold cross validation were conducted, we took a simple average from the outcome of these 10 runs to compare performances of different feature sets.

Table 4 shows that when more tweets are used to compute features, the overall accuracy increases in all 3 models, though the improvement is small in percentage. Adding oscillation properties does not seem to improve this assessment indicator effectively. Similar results can be observed for the binary class problem in Table 7.

One way to examine these small differences more carefully is to use statistical tests. For example, while adding FT spectrum to the basic variables, accuracy increases only 0.2%, yet independent samples t-test indicates that the difference is significant ($p<0.05$).

Another way to examine the impact of different tweet lengths on accuracy is to use a redesigned indicator. The original accuracy is weighted on all classes, thus the majority class has a much higher influence than the minority class. Chen et al. [15] suggested a simple average of recall rates to compute weighted accuracy. With this new definition, improvement in recall rate of the minority class will have the same influence as the majority class on the accuracy indicator.

We did not pursue the above two means to analyze the overall accuracy further because we think the analysis of high popularity class is more important. To analyze performance on this class, we use precision, recall and $F_1$ score.

From Tables 5 and 8, we observe that increasing the number of tweets is most likely to improve the precision rate of the multiple class problem or the binary class problem. The improvement can be more than 10% from 25 tweets to 200 tweets. In terms of the oscillation properties dimension, FT or WT spectrum generally improves the precision rate, and the improvement can be more than 10%.

The performance of recall rate is more complicated. In Model 1, increasing the number of tweets increases the recall rate, but the same thing cannot be said when oscillation properties are used (Tables 6 and 9). From the experimental results, we can only say that WT spectrum provides a better performance than FT spectrum, and WT spectrum offers some advantages to improve recall rates when the tweet length is small.

The combined performance is given by the $F_1$ score. Figure 2 shows that for the multiple class problem, WT spectrum improves $F_1$ score when $n=25, 50$ or 100, and the improvement can be as high as 9%. For the binary class problem, WT spectrum always gives an edge with the improvement as high as 8%. In contrary, FT spectrum does not provide significant help to improve the $F_1$ score.

The highest $F_1$ score (0.462) appears when BRF is used in the binary class problem with 200 early tweets in Model 3. In case that time does not allow us to obtain 200 tweets, Model 3 may be used with 50 tweets to reach an $F_1$ score of 0.439. Using the original multiple class problem with RF prediction, Model 3 with 25 tweets provides an $F_1$ score of 0.448.

Overall, we conclude that WT spectrum provides a better help than FT spectrum in improving prediction performance. According to the theory of frequency domain analysis, WT decomposes time series into a finer resolution than FT, thus we expect to obtain better oscillation properties from WT than from FT.

5. Conclusions

Online social networks provide communication channels to spread an idea, behavior, style or usage throughout the world village. Twitter is a special online
service that provides both social network and microblog functions. Posting tweets through devices from desktop to mobile is the main activity of the microblog function, while following and retweeting offer the social network function. Users post tweets by encoding topics in the form of hashtags, which are summarized by Twitter to make a list of current trending tags.

Like other Internet memes, plain looking hashtags can go viral unexpectedly, and high hopes tags can go south without a reason. Detecting the future popularity of hashtags at their early stage has practical applications for marketers because they can use viral tags to run a successful campaign of their products or services. In this study, we use early adoption properties of a hashtag to predict its later popularity level.

Even though hashtags are just a single word made by the tweet authors, many features can be extracted from a hashtag in Twitter. For example, tweets provide a context for the hashtag. Are there any hyperlinks or mentions in the tweet? Social influence resulting from the follow network in Twitter may impact the spread coverage of a hashtag as well. How many users have adopted the early tweets? What are the social influences of these early adopters? The influences can be judged from the number of followers, the number of statuses or the number of lists to which early adopters have been added. The time series associated with the early tweets adoption offers another feature for the prediction task. Differential series derived from the adoption series reveals whether the adoption of a hashtag is speeding up or slowing down.

A quick reasoning shows that the mean and standard deviation statistics cannot catch the oscillation properties of a differential series. Thus, frequency domain analysis must be used to accommodate the requirement. The research objectives are to investigate whether or not the popular frequency domain analysis tools FT and WT may help predict viral hashtags, and if they do, which one provides a better solution.

By using the Twitter Streaming API, we have collected more than 18 million tweets containing 748 thousand hashtags. Extensive experiments show that WT spectrum improves the prediction of viral tags over basic predictor variables while FT spectrum does not. Computing WT spectrum is fast with today’s machines and algorithms, thus one should try to augment the simple statistics of mean and standard deviation with these oscillation properties.

We have ignored detailed follow network of Twitter in this study because of difficulties in collecting relationship data with current Twitter’s REST API. However, summarized network data such as followers count and listed count have been used as predictors. In the future, if sociogram of Twitter can be easily obtained, the impact of network structures such as communities should also be considered.

Another direction for future studies is to differentiate the role of profile variables in different stages of early tweets adoption. Some profile variables such as followers count may play a more important role at the beginning of a tag’s adoption history, while some may become more important at a later stage. A two-stage or multi-stage prediction architecture may be designed to further improve the prediction task.

A possible bias problem of sampled tweets obtained through the Streaming API should be noted. Morstatter et al. have found some statistical differences between sampled tweets and all tweets from the Firehose [16]. Popular hashtags mined through sampled tweets may be different from those observed in the full Firehose. Thus, users should interpret their results cautiously when they use free sampled tweets to make a prediction.

Acknowledgments. This work is supported in part by grants from the ministry of science and technology (Taiwan) under contract numbers MOST-104-2632-E-366-001 and MOST-103-2410-H-366-001. The author appreciates constructive comments from the anonymous reviewers.

6. References


