Exploring Time Series Spectral Features in Viral Hashtags Prediction

Shing H. Doong
ShuTe University
tungsh@stu.edu.tw

Daniel Chung
University of California, San Diego
dac007@ucsd.edu

Abstract

Viral hashtags spread across a large population of Internet users very quickly. Previous studies use features mostly in an aggregate sense to predict the popularity of hashtags, for example, the total number of hyperlinks in early tweets adopting a tag. Since each tweet is time stamped, many aggregate features can be decomposed into fine-grained time series such as a series of numbers of hyperlinks in early adopting tweets. This research utilizes frequency domain tools to analyze these time series. In particular, we apply scalogram analysis to study the series of adoption time lapses and the series of mentions and hyperlinks in early adopting tweets. Besides continuous wavelet transforms (CWTs), we also use fast wavelet transforms (FWTs) to analyze the time series. Through experiments with two sets of tweets collected in different seasons, out-of-sample cross validations show that wavelet spectral features can generally improve the prediction performance, and discrete FWT yields results as good as the more complicated CWT-based methods with scalogram analysis.

1. Introduction

The rapid development of information and communication technologies (ICTs) enables every Internet user to be both an information consumer and a producer. Founded on human's nature to connect and share, social networking sites and services have burgeoned in the past few years.

Facebook is the world's largest social networking site, while Twitter is the largest microblogging site. In comparison to Facebook, Twitter allows users to send only short messages (tweets), but the application devices and channels can be more flexible than Facebook. A tweet is a short message containing no more than 140 characters in its context.

Users can embed topical items called hashtags in their tweets. A hashtag is a string of characters without spaces following the hash sign (#). Hashtags can be lexically meaningful or not. Each day, many hashtags are created and diffused with tweets. Once created and diffused, hashtags may be mutated by new ideas or current events and compete with other hashtags for users' attention; some die immediately after creation, and others may survive for a longer time and become viral at some future time.

According to Twitter's own research, tweets with at least one hashtag receive 2 times more engagements than tweets with no hashtags [1]. Here, engagements can be defined as clicks, retweets, replies and favorites. Different studies regarding the boosting of engagement via hashtags may show the result differently [2], yet hashtags are a "hard feature" of Twitter and used with photos, links and videos to promote tweets engagement [3]. Predicting hashtags' virality at their early age allows marketers to design effective marketing practices and may have real applications in business. The research community has paid much attention to tackle this problem and created many interesting results.

Previous research shows that tag content and tweet context are useful features to predict a tag's popularity level [4][5]. The embedded network structures of tweet adopters are also useful because Twitter's follow network provides a convenient conduit to spread hashtags [6]. Other types of tag features may include the adoption time series [7], because different adoption time patterns indicate different diffusion speeds.

Like previous research, our goal is to predict the popularity level of a hashtag by using its early adoption properties. By early, we mean the earliest few tweets adopting a tag. Previous studies often use prediction features in an aggregate sense, e.g., the total number of hyperlinks in early tweets adopting a tag. Since each tweet is time stamped, we may decompose this aggregate feature into a series of numbers of hyperlinks according to timestamps. Thus, fine-grained time series data may be obtained.

Even though fine-grained time series data can be obtained via decomposition, they were mostly analyzed by using simple statistics such as the mean and standard deviation [7]. Notice that these statistics cannot capture wavy properties of a time series because they are invariant no matter how we rearrange the time series. Wavy properties describe energy of time series. If we do not consider them, we may lose

URI: http://hdl.handle.net/10125/41381
ISBN: 978-0-9981331-0-2

opportunities to extract valuable features from the fine-grained time series.

In order to capture energy features, frequency domain tools are often used to analyze time series. Wavelet transforms (WTs) were used in [8] to expand the adoption time series of [7] in time-frequency domain, and wavelet spectrum was extracted and used as prediction variables.

We explore the opportunities of spectral analysis to analyze time series decomposed from aggregate features that have been used in literature. In particular, scalogram analyses are used to analyze spectrum obtained from CWTs. Alternatively, discrete FWTs are also used to study the time series.

This paper is organized as follows. Section 2 is devoted to a literature review on hashtag popularity prediction, frequency domain tools and scalogram analysis. Methodology and experimental data sets are described in section 3 followed by experimental results and discussions in section 4. We conclude the paper with remarks in section 5.

2. Literature review

Twitter is the world's largest microblogging service offering both weblog functions and social networking features. By following a user, tweets created or retweeted by the followee will automatically show up in the follower's home timeline. This follow network is a directed network and has made Twitter very different from other social networking services such as Facebook.

2.1. Hashtags popularity prediction

Like many data mining tasks, the first step in predicting the popularity of a hashtag is choosing a suitable set of predictors. Unlike data mining tasks based on relational databases, there are many ways to extract features from hashtags.

The inherent content of a hashtag is considered to be an important factor for its popularity. Content based features such as the number of words, lexical items and emotional characteristics have been used to study the spread of hashtags in Twitter [5]. Contextual features from tweets have also been used in literature. Ma et al. used fractions of tweets containing URLs, fractions of tweets containing mentions (@), and fractions of polarized sentiments of tweets as contextual features of hashtags in their study [4]. Suh et al. found that the numbers of URLs and hashtags in a tweet are strongly correlated to the retweetability of the tweet [9].

Using the follow network, tweets and hashtags may be disseminated conveniently. Central dogma in social influence theory predicts that influential nodes of a network are more likely to spread messages successfully, albeit there are many ways to define the influential capability. The in-degree (follower) count in the Twitter network is arguably the simplest indicator to measure influential capability. Weng et al. considered the community structure of Twitter sociogram in their study [6]. Sociogram is dynamic in time and difficult, if not impossible, to obtain in Twitter, thus community structures are not considered in this study.

Early popularity of a hashtag is closely related to its later popularity [10]. Weng et al. used the early adoption time series to predict the popularity of a hashtag [7]. However, they considered only the mean and standard deviation statistics. Doong adds spectral features derived from the Fourier transform (FT) and WT of the series [8].

2.2. Frequency domain analysis

The venerable FT has been used in engineering to study waves for a long time. FT converts a time series into its frequency domain data. Using a global convolution with the basic harmonic functions, FT loses time resolution in the transformed domain. Thus, with FT we can hear pitches (frequencies) but cannot tell when they happen. Short time Fourier transform (STFT) with windowing functions has been developed to overcome some shortages of FT.

In order to recover the time resolution in frequency domain analysis, WT uses multiple scales of a mother wavelet to decompose time series [11]. The output of WT is in a time-frequency-amplitude format, whereas FT has frequency-amplitude resolution only. WT can be divided into the categories of CWT and discrete wavelet transform (DWT) depending on whether the scaling factor is continuous or not [12].

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

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

In the above formula, * indicates the complex conjugate and δ is the time difference between two successive events. Being a mother wavelet function, $\psi(n)$ has a finite effective support in the time domain, thus the above transform is a local convolution around the focal point m. In this study, we use the Mexican hat mother 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

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

By varying the scale *s*, we obtain a picture of the time series at different resolutions. Though scale *s* can be varied continuously, it is not necessary to do so because nearby scales produce highly correlated wavelet coefficients. We follow instructions in [11] to choose discrete samples of scales. For the DOG wavelet, the corresponding Fourier wave length is approximately equal to 4*s*, thus the Fourier frequency is about 1/(4*s*).

Unlike CWT, scales in DWT can be varied only discretely. Most of the time, scales in DWT are dyadic scales (2^n). One needs to choose a scaling function $\phi(x)$ and a wavelet function $\psi(x)$ in the application of DWT. The former is used to approximate a given function while the latter is used to detail the difference between two successive levels of approximations. The expansion bases in equations (3) and (4) describe dyadic scaling and integral translation of these two functions. A given function *f*(*x*) is expanded in equation (5) with these bases [12]. Many DWT expansion bases exist in literature, and we adopt the classical Haar bases in the following experiments.

$$\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k),\ j \in Z, k \in Z \quad (3)$$

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k),\ j \in Z, k \in Z \quad (4)$$

$$f(x) = \sum_{k \in Z} c_{j_0,k}\phi_{j_0,k}(x) + \sum_{j \geq j_0, k \in Z} d_{j,k}\psi_{j,k}(x) \quad (5)$$

Using orthogonality, the expansion coefficients $c_{j_0,k}$ and $d_{j,k}$ are computed with a convolution operator like equation (1). Mallat developed an FWT algorithm to compute these coefficients without tedious convolution operation [13]. The output of FWT is a series with the same length as the input series. This series stores detail coefficients from the finest scale to the approximation coefficients of the coarsest scale.

2.3. Scalogram analysis

When amplitudes for CWT of a time series are plotted against time and frequency, we get a scalogram for the time series. The horizontal axis is the time domain, the vertical axis is the frequency (scale) domain, and the color of a pixel indicates the strength (amplitude) of the wavelet coefficient at that specific time and frequency position. Using data set I (to be described later), we pick three hashtags with different popularity levels (low, medium and high) and plot their scalograms in Figure 1. The amplitude ranges from very weak in red to very strong in violet. We can see that for a substantial time and frequency domain, the amplitude of a low popularity tag (left panel) is very weak. More activities are detected for the medium popularity tag (middle panel) and the high popularity tag (right panel). However, it is difficult to tell differences between the last two scalograms with the naked eye.

Scalograms are considered textures and 2D DWT have been used to analyze these textures in many applications [14][15]. We follow [16] to decompose a scalogram into 2D DWT image. Figure 2 shows the two-level decomposition of a scalogram. The left panel decomposes the scalogram into four quadrants by using separable discrete wavelet bases. The lower right quadrant (HH1) contains detail coefficients in both directions, the upper right quadrant (HL1) contains detail coefficients in *x* (time in Figure 1) and approximation coefficients in *y* (frequency in Figure 1), the lower left quadrant (LH1) contains approximation coefficients in *x* and detail coefficients in *y*, and the upper left quadrant (LL1) contains approximation coefficients in both directions. The LL1 region can be further decomposed into the second level coefficients with a coarser scale (right panel).

In the following experiments, we use three-level 2D DWT decompositions to split scalograms into ten regions (HH1, HL1, LH1, HH2, HL2, LH2, HH3, HL3, LH3 and LL3). For each region, we compute the mean magnitude of coefficients in that region. Thus, each scalogram is represented by 10 features.

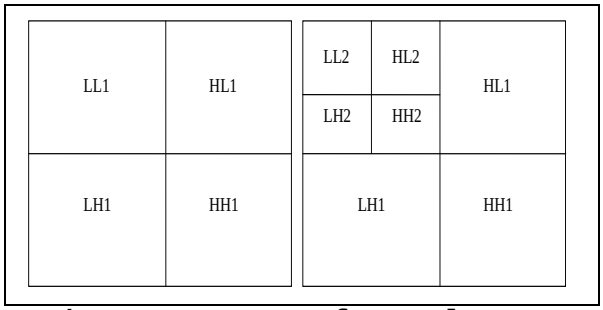

Figure 2: 2D DWT of a scalogram.

3. Methodology

We describe our data collection method and basic statistics of the collected data, followed by feature preparation. The experimental procedure is explained next.

3.1. Data collection

Twitter has released two types of APIs (REST and Streaming) allowing authenticated users to collect or manipulate tweet data [17]. The REST API provides programming interfaces to read and write Twitter data, author a new tweet, read author profiles and follower data. The Streaming API gives developers low latency access to Twitter's global stream of public tweets. According to Twitter, the streaming API can return up to a maximum of 1% public tweets that are currently being created [17].

We use the GET statuses/sample Streaming API to collect two sets of public tweets. The first set (data set I) was collected between May 13, 2015 and June 2, 2015. After excluding non-English based tweets, we ended up with more than 18 million tweets. Each tweet is processed to preserve the screen name of the author, followed by user id, timestamp, status id and the tweet context. The tweet context may contain RT (indicating a retweet), mentions, hyperlinks or hashtags.

From each tweet, we extracted user id, timestamp, status id, number of mentions, number of hyperlinks and hashtags after discarding the screen name and the remaining tweet context. By ignoring case, we found 748224 different tags in data set I. In order to conduct the experiments, we deleted tags supported by less than 300 tweets and tags with one character only. This left us with 2287 tags. Table 1 lists the support distribution for hashtags in data set I. The popularity field is the dependent variable for the current study.

Table 1. Distribution for data set I.

Support (cnt)	Floor	# of tags	Popularity (level)
300~999	2	1657	Low (1)
1000~9999	3	581	Medium (2)
10000~99999	4	48	High (3)
100000 and up	5	1	High (3)

Floor is the integer part of $\log_{10}$(cnt).

The second set (data set II) of tweets was collected between October 7, 2015 and November 28, 2015. This corpus has about 46 million tweets and spans over a period of 7 weeks. After processing tweets as above, we ended up with 1463802 different tags. By deleting tags with support less than 500 and tags with one character, there are 3774 remaining tags. The support distribution of data set II is listed in table 2. For both data sets, we combine the very popular tags with support greater than 100000 and popular tags of the previous level to form the class of hashtags with high popularity.

Table 2. Distribution for data set II.

Support (cnt)	Floor	# of tags	Popularity (level)
500~999	2	1792	Low (1)
1000~9999	3	1871	Medium (2)
10000~99999	4	103	High (3)
100000 and up	5	8	High (3)

Floor is the integer part of $\log_{10}$(cnt).

The reason to use different support thresholds (300 vs. 500) to extract tags of interest in the two data sets is we would like to prepare experimental data of comparable size in the class of low popularity. Had we chosen a threshold of 300 (400) to extract tags in data set II, there were 4015 (2608) tags in the class of low popularity.

3.2. Feature extraction

We use early adoption properties of a hashtag to prepare the predictor variables. By using timestamps of tweets, we can extract the earliest n (=16, 32, 64, 128 and 256 in experiments) tweets that contain the hashtag. In order to use 2D DWT analysis for scalograms, we consider series lengths of powers of 2 only.

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. The number of early adopters has been used in [7] and represents one of our predictors. An early adopter may retweet a tweet containing the tag or simply starts a new tweet containing the tag.

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 [4].

Since our data contain timestamps of all tweets, we can decompose cm and ch temporally into two series. The first series contains the number of mentions in early tweets and the second series contains the number of hyperlinks of these tweets.

The next set of variables comes from the adoption timestamps of early tweets. Let $t_1, t_2, .., t_n$ represent the adoption time of these n tweets. A differential series is derived from this series by taking differences between successive adoptions: $\delta_1 = 0, \quad \delta_i = t_i - t_{i-1}, \; i = 2,3,..., n$. Elements of differential series are nonnegative since we have ordered tweets according to their timestamps. These series may reveal unique diffusion patterns for tags with different popularity levels. 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 this tag may be diminishing. For most tags, the differential series is not entirely increasing or decreasing, and diffusion speed of the tag may speed up or slow down from time to time. Due to this reason, we should consider wavy properties of the differential series. Weng et al. uses the mean (*mu*) and standard deviation (*sd*) to describe this differential series [7]. Like [8], we add spectral features to capture wavy properties of differential series.

Together, there are three series for each tag. The first two are the decomposed *cm* and *ch* series, and the third one is the differential time series. In order to explore the applicability of frequency domain analysis, we consider 3 types of spectral features.

The first type of spectral feature is similar to the one in [8]. Each time series is analyzed with CWT with proper sampled scales [11]. We assume the time gap is one in all our series, thus the equivalent Fourier frequency is between 0 and .5. This domain is equally divided into 10 regions (0~.05, .05~.1, …, .45~.5). In each region, the marginal spectrum is summed up to represent a spectral feature. In total, there are ten spectral features.

The second type of spectral feature is also based on CWT of the time series. Instead of ten equal frequency regions, we divide the frequency domain into 64 equal regions. For each time point, amplitudes within the same frequency region are summed up to form a scalogram of size n x 64, where n is the length of the series. In theory, this scalogram contains more information than the ten spectral features presented above. In order to extract features from scalograms, we use three-level 2D DWT to decompose scalograms. The total number of spectral features is also ten in this case.

The third type of spectral feature is based on discrete FWTs of the time series. The original series is processed with FWT to obtain 4 segments: H1 (n/2 elements), H2 (n/4 elements), H3 (n/8 elements) and L3 (n/8 elements), where H1 contains detail coefficients of the finest scale and L3 contains approximation coefficients of the coarsest scale. Then, magnitudes in each segment are averaged to obtain a representative feature. In total, four spectral features are obtained for each time series. Without resorting to CWT and scalogram analysis, we would like to see how this naive FWT compares to other spectral features in predictions.

The various variables used in the following experiments are summarized in Table 3. The process of extracting spectral features (s_1~ s_k) is applied to each of the 3 time series explained above.

Table 3. Variables used in the study.

Variable	Role	Meaning
na	Input	Number of early adopters
cm	Input	Total count of mentions
ch	Input	Total count of hyperlinks
mu	Input	Mean of differential series
sd	Input	Standard deviation of differential series
s_1~ s_k	Input	CWT marginal spectrum, Scalogram features or FWT features
cla	Output	Popularity level from Table 1 and 2

3.3. Experimental procedure

After extracting features from the collected tweets, we have a table of 2287 and 3774 records for data sets I and II respectively. The *cla* variable is the output variable, while the other variables are the predictor variables. The random forest (RF) algorithm [18] is used as the classification 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 [18]. 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.

The prediction performance for each class can be measured 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 (6). It is between 0 and 1 with a higher score indicating a better prediction result. Since high popularity tags have higher stakes in practical applications, we focus on the F_1 score of this class.

$$F_1 = 2pr/(p+r) \tag{6}$$

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

4. Experimental results and discussions

We compare prediction results from four models based on different predictors. Model 1 (Basic) uses basic variables (*na*, *cm*, *ch*, *mu*, *sd*) only, Model 2 (Cwt) uses basic variables plus 10 marginal spectral features from CWT for each time series, Model 3 (Scalog) uses basic variables plus 10 scalogram features for each time series, and Model 4 (FWT) uses basic variables and 4 FWT features for each time series. Three time series are considered: the decomposed *cm* series, the decomposed *ch* series and the differential time series.

If the performance of a classifier is measured on the training set, the result is usually over-optimistic. Thus traditional doctrine in machine learning sets aside a test data set not seen in the training stage to assess the performance of a classifier. When the experimental data set is of moderate sizes, cross validations (CVs) are commonly used to assess the classifier in an out-of-sample setting. A *k*-fold CV starts with a random partition of the data set into *k* disjoint subsets of approximately equal size. Each time, one subset is chosen as the test set and the remaining *k*-1 subsets form the training set. After each subset has taken the role of a test set, the *k* out-of-sample prediction rates are averaged to get a final prediction rate. Though CVs attempt to provide a fairer judgment of the classifier, their results still depend on initial partitions of the data set. Thus, several runs of CVs are needed to get a more robust judgment of the classifier.

In the following, each prediction task is conducted in a 10-fold CV setting. In order to minimize the effect of partition randomization in CV, 10 runs of 10-fold CV were performed for each experimental scenario (5 early tweet sizes x 4 models of predictors). Regarding the classifier, RF is used in a setting with 300 trees and $\log_2$(number-of-predictors)+1 random features for node decision.

4.1. Prediction results

Accuracies (%) from 10 runs of 10-fold CVs were averaged and reported in Table 4 for data set I and Table 5 for data set II. The results show that accuracy increases with the number of early tweets. This can also be seen in Figures 3 and 4. For a fixed size of early tweets, differences between any two models are moderate. The biggest improvement (1.02%) over the Basic model comes from FWT model in data set II when 16 early tweets are used. Experimental scenarios where spectral features add prediction power to the Basic model are highlighted in yellow. For data set I, FWT model beats the Basic model 4 times; for data set II, both Scalog and FWT models beat the Basic model 4 times. In general, FWT model performs better than the more complicated Cwt model.

Because accuracy is the weighted recall rate, class 1 has a dominant effect. Class 3 has the lowest impact on accuracy, yet it contains tags of high stakes in most applications. We turn our attention to the result of this class next. The F_1 score for class 3 prediction is reported in Tables 6 and 7 for data sets I and II respectively. Again, FWT model has a better performance than the other two models in terms of beating the Basic model. The biggest improvement (2.8%) over the Basic model happens when 128 early tweets are used with FWT in data set II.

Table 4. Accuracy(%) for data set I.

Length	Basic	Cwt	Scalog	FWT
16	81.49	80.90	81.29	81.95
32	82.19	81.62	81.33	82.26
64	84.01	83.60	83.66	83.99
128	85.33	85.18	85.08	85.51
256	86.28	86.56	86.32	86.45

Table 5. Accuracy(%) for data set II.

Length	Basic	Cwt	Scalog	FWT
16	70.96	71.95	71.77	71.98
32	73.16	73.75	74.05	73.84
64	75.23	75.20	75.46	75.98
128	77.08	77.56	77.66	77.59
256	80.05	79.40	79.46	79.75

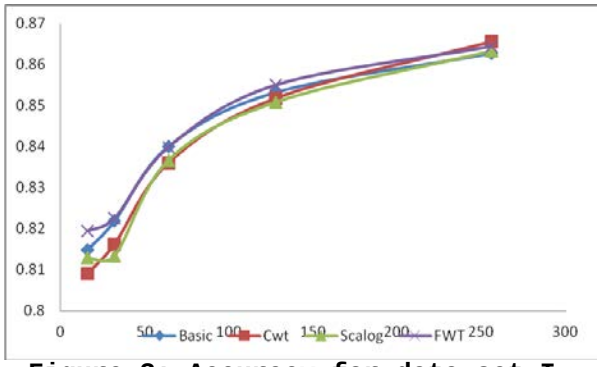

Figure 3: Accuracy for data set I.

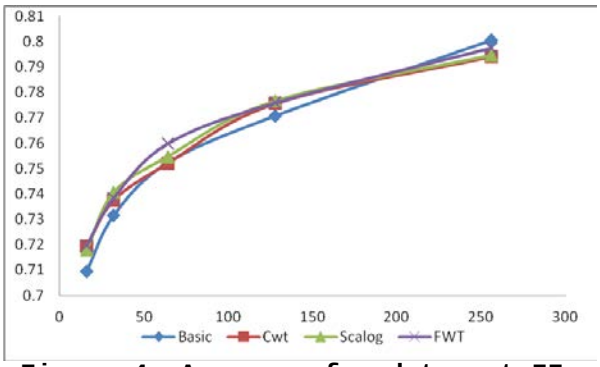

Figure 4: Accuracy for data set II.

Table 6. F_1 score for data set I.

Length	Basic	Cwt	Scalog	FWT
16	.397	.370	.357	.419
32	.338	.345	.315	.348
64	.440	.368	.400	.417
128	.431	.393	.392	.446
256	.489	.390	.416	.441

Table 7. F_1 score for data set II.

Length	Basic	Cwt	Scalog	FWT
16	.547	.495	.500	.525
32	.545	.518	.545	.536
64	.556	.575	.545	.573
128	.590	.613	.614	.618
256	.666	.672	.690	.657

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 (2.1% in data set I and 2.9% in data set II) in the collected data sets. Though RF has many advantages to overcome over-fitting issues, it still suffers from the curse of imbalanced data in sampling training records. The minority class may not be sampled enough for tree learning.

We adopt a balanced random forest (BRF) solution to overcome the data imbalance issue, that is, when sampling training data, the majority class is down-sampled while the minority class is up-sampled [19]. In the following, we use BRF to tackle the high popularity class prediction problem after combining low and medium popularity classes into a single class. The tertiary prediction problem is effectively converted to a binary prediction problem.

The same experimental settings are adopted for BRF: 300 decision trees in a forest, $\log_2$(number-of-predictors)+1 random features in node decision, and 10 runs of 10-fold CVs.

Tables 8 and 9 show the averaged prediction accuracy for data sets I and II respectively. It is obvious that accuracy has improved significantly from the original tertiary prediction problem. Tables 10 and 11 report F_1 score of the viral class for data sets I and II respectively. The F_1 score improves as well, though not as significant as improvements in accuracy. Again, FWT model out-performs the other two models in beating the Basic model.

Table 8. Accuracy(%) for data set I.

Length	Basic	Cwt	Scalog	FWT
16	97.19	97.25	97.55	97.39
32	97.02	97.51	97.57	97.49
64	97.61	97.44	97.58	97.58
128	97.58	97.50	97.51	97.64
256	97.94	97.80	97.78	97.82

Table 9. Accuracy (%) for data set II.

Length	Basic	Cwt	Scalog	FWT
16	97.09	97.32	97.49	97.29
32	97.32	97.46	97.48	97.49
64	97.59	97.60	97.58	97.69
128	97.89	97.94	98.03	98.01
256	98.05	98.13	98.09	98.22

Table 10. F_1 score for data set I.

Length	Basic	Cwt	Scalog	FWT
16	.429	.392	.451	.444
32	.381	.414	.401	.400
64	.435	.427	.427	.425
128	.416	.394	.362	.432
256	.501	.469	.447	.446

Table 11. F_1 score for data set II.

Length	Basic	Cwt	Scalog	FWT
16	.504	.488	.483	.500
32	.521	.506	.509	.519
64	.551	.528	.539	.555
128	.594	.592	.605	.597
256	.621	.623	.603	.632

4.3. Discussions

In general, adding spectral features to the Basic model improves prediction performance in terms of accuracy and F_1 score. Since CWT renders more wavy properties than FWT, we had expected Cwt and Scalog models to out-perform FWT model. Due to the fine-grained features of scalograms, we also expected Scalog model to out-perform Cwt model.

Experiments show that FWT model out-performs the other two models in beating the Basic model and Scalog model is not much better than Cwt model. One possible cause for the unexpected under-performance of Scalog model could be coming from the scalogram analysis, which averages out wavelet magnitudes in different regions. Though we know that scalograms capture more information than spectrum in Cwt model or FWT model, averaging magnitudes may just level out advantages of these fine-grained features. As sharp turns in time series normally indicate special events in social data, we may develop more effective features to spot sharp turns and analyze scalograms.

In terms of F_1 score for the viral class, FWT model beats the Basic model with large tweet size in data set II. We had expected the major improvement would go to cases with small tweet size as in data set I. When short early tweet history, e.g. 16 or 32, is used, the distinguishing power of *mu* between popular and unpopular tags is often not significant statistically. On the other hand, when long early tweet history is used, *mu* often provides significant differences between popular and unpopular tags. The failure of FWT to improve F_1 score for short early tweet history in data set II may come from the inherent distribution of data in the set.

5. Conclusions

Online social networks provide communication channels to spread an idea, behavior, style or usage throughout the Internet. Twitter, the largest microblogging service site of the world, provides both social network and microblogging functions. Hashtags with proper topics may spread through the Twitter network like a virus. Viral hashtags are rare, but may have useful applications for marketing companies. Detecting hashtags' popularity at their early age of life is interesting and practical. In this study, we use early adoption properties of a hashtag to predict its future popularity level.

Previous studies have used many types of features to predict the virality of hashtags. These features may be categorized into 3 areas: content/context, network, and time series. The first two types of features have been much more investigated in literature than the third type of features. By using timestamps of tweets, many aggregate features in literature may be decomposed into fine-grained time series which can be investigated with frequency domain tools.

We used CWT to produce a detailed scalogram for each time series, and sought methods to extract meaningful features from scalograms. Literature shows that texture analysis with 2D DWT may help classify textural scalograms. Thus, two prediction models based on CWT have been proposed: Cwt and Scalog. The Cwt model adds marginal spectrum to the Basic model, and the Scalog model uses 2D DWT scalogram analysis to augment the Basic model. By resorting to DWT, we also considered a third prediction model based on FWT.

By using Twitter Streaming API, we collected two sets of sampled public tweets in different seasons. The first set covers a 3-week period in the middle of 2015, and the second set covers a period of more than 7 weeks near the end of 2015. Extensive experiments show that the simple FWT model out-performs the more complicated CWT based models in beating the Basic model. One possible cause for the under-performance of Scalog model has been discussed above. Effective features need to be designed in order to extract CWT spectrum from scalograms.

Many DWT expansion bases exist in literature. We only experimented with the Haar bases in this study. Future work may investigate whether the choice of DWT bases affects the performance of FWT model.

Another direction for future work is to design a decision rule to tell when to use the decomposed time series. For example, if aggregate features already provide distinguishing power to separate viral classes from nonviral classes, is decomposition and spectral analysis still needed? Experiments with other aggregate features may also be conducted, e.g., a decomposed series containing numbers of emoticons in early adoption tweets.

In addition to the above feature-based consideration, classification algorithms may affect the prediction rate as well. In this study, due to the experimental time constraints, we have used a forest of 300 decision trees in RF. In practice, between 500 and 1000 trees are commonly used. Future work may consider using a bigger forest in RF.

One may argue that using an odd number of trees helps the performance of RF since RF uses a voting mechanism to decide the final label of a prediction. This claim is probably true for a binary prediction problem, e.g., viral vs. nonviral hashtags, because tied votes will never happen with odd numbered trees. Regarding the original tertiary prediction problem, the help of odd numbered trees is unclear.

RF is an ensemble classification algorithm using bagging techniques on a forest of trees. Other ensemble algorithms on trees may use boosting techniques as well. XGBoost (eXtreme Gradient Boosting) is a gradient tree boosting system that is scalable and has been successfully applied in many studies and competitions [20]. RF algorithm in this study may be replaced with XGBoost in a future study.

Acknowledgments. This work was supported in part by a grant from the ministry of science and technology (Taiwan) under the contract number MOST-105-2632-E-366-001. The authors appreciate constructive comments from the anonymous reviewers.

6. References

[1] M. Luckie, “Best Practices for Journalists”, The Official Twitter Blog, https://blog.twitter.com/2012/best-practices-for-journalists, accessed August 27, 2016.

[2] I. Mullane, “Are Hashtags Overrated and Overused? The Surprising Effect of Social Media #’s”, Locowise Blog, http://locowise.com/blog/are-hashtags-overrated-and-overused-the-surprising-effect-of-social-media-hashtags, accessed August 27, 2016.

[3] S. Rogers, “What Fuels a Tweet’s Engagement”, The Official Twitter Blog, https://blog.twitter.com/2014/what-fuels-a-tweets-engagement, accessed August 27, 2016.

[4] Z. Ma, A. Sun, and G. Cong, “On Predicting the Popularity of Newly Emerging Hashtags in Twitter”, Journal of the American Society for Information Science and Technology, 2013, 64(7), pp. 1399-1410.

[5] O. Tsur, and A. Rappoport, “What’s in a Hashtag? Content Based Prediction of the Spread of Ideas in Microblogging Communities”. In: Proceedings of the 5th ACM International Conference on Web Search and Data Mining, 2012, pp. 643-652.

[6] L. Weng, F. Menczer, and Y.Y. Ahn, “Virality Prediction and Community Structure in Social Networks”, Scientific Reports, 2013, 3 (2522).

[7] L. Weng, F. Menczer, and Y.Y. Ahn, “Predicting Successful Memes using Network and Community Structure”, In: Proceedings of the 8th International AAAI Conference on Weblog and Social Media, 2014, pp. 535-544.

[8] S. Doong, “Predicting Twitter Hashtags Popularity Level”, In: Proceedings of the 49th Hawaii International Conference on System Sciences, 2016.

[9] B. Suh, L. Hong, P. Pirolli, and E.H. Chi, “Want to be Retweeted? Large Scale Analytics on Factors Impacting Retweet in Twitter Network”, In Proceedings of IEEE International Conference on Social Computing, 2010, pp. 177-184.

[10] G. Szabo, and B.A. Huberman, “Predicting the Popularity of Online Content”, Communication of the ACM, 2010, 53(8), pp. 80-88.

[11] C. Torrence, and G.P. Compo, “A Practical Guide to Wavelet Analysis”, Bulletin of the American Meteorological Society, 1998, 79(1), pp. 61-78.

[12] R. C. Gonzalez, and R. E. Woods, “Digital Image Processing”, 3rd edition, Pearson Education, 2008.

[13] S. Mallat, “Multifrequency Channel Decomposition of Images and Wavelet Models”, IEEE Transaction on Acoustics, Speech and Signal Processing, 1989, 37, pp. 2091-2110.

[14] Y. M. G. Costa, L. S. Oliveira, A. L. Koerich, and F. Gouyon, “Music Genre Recognition Using Spectrograms”, In: Proceedings of the 18th International Conference on Systems, Signals and Image Processing, 2011, pp. 151-154.

[15] J. Dennis, H. D. Tran, and H. Li, “Spectrogram Image Feature for Sound Event Classification in Mismatched Conditions”, IEEE Signal Processing Letters, 2011, 18(2), pp. 130-133.

[16] S. Arivazhagan, and L. Ganesan, “Texture Classification Using Wavelet Transform”, Pattern Recognition Letters, 2003, 24, pp. 1513-1521.

[17] Twitter, API Overview, https://dev.twitter.com/overview/api, accessed November 28, 2015.

[18] L. Breiman, “Random Forests”, Machine Leaning, 2001, 45(1), pp. 5-32.

[19] C. Chen, A. Liaw, A., and L. Breiman, “Using Random Forest to Learn Imbalanced Data”, University of California, Berkeley, 2004.

[20] T. Chen, and C. Guestrin, “XGBoost: A Scalable Tree Boosting System”, Preprint arXiv:1603.02754 (2016).

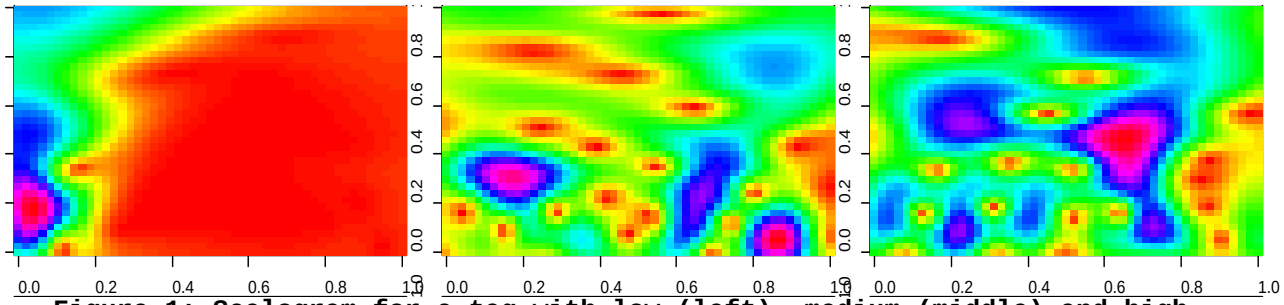

Figure 1: Scalogram for a tag with low (left), medium (middle) and high (right) popularity level.