Cascade Size Prediction in Online Social Networks

Zubair Shafiq† and Alex Liu‡
†Department of Computer Science, The University of Iowa, Iowa City, IA 52242, USA.
‡Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA.
Email: zubair-shafiq@uiowa.edu, alexliu@cse.msu.edu

***Abstract*—Cascades represent an important phenomenon across various disciplines such as sociology, economy, psychology, political science, marketing, and epidemiology. The goal of this paper is to develop a model for cascade size prediction in online social networks. Specifically, *given the first τ_1 edges in a cascade, we want to predict whether the cascade will have a total of at least τ_2 ($\tau_2 > \tau_1$) edges over its lifetime without any a priori information.* In this paper, we propose a Multi-order Markov Model (M^3) for cascade size prediction in online social networks. Our evaluations using a Twitter data set show that M^3 based cascade size prediction scheme outperforms the baseline scheme based on cascade graph features such as edge growth rate, degree distribution, clustering, and diameter. M^3 based cascade size prediction scheme consistently achieves more than 90% prediction accuracy in different experimental scenarios.**

I. Introduction

Background and Motivation. The term *cascade* describes the phenomenon of something propagating along the links in a social network. That something can be information such as a URL, action such as a monetary donation, influence such as buying a product, discussion such as commenting on a blog article, and a resource such as a torrent file. Based on what is being propagated, we can categorize cascades into various classes such as information cascades [7], action cascades [10], influence cascades [23], discussion cascades [16], and resource cascades [36]. Consider a toy example where user A, connected to users B and C in a social network, broadcasts a piece of information (*e.g.* a picture or a news article) to his neighbors. Users B and C, after receiving it from user A, may further rebroadcast it to their neighbors resulting in the formation of a cascade.

Cascade phenomenon has been a fundamental topic in many disciplines such as sociology, economy, psychology, political science, marketing, and epidemiology with research literature tracing back to the 1950s [32]. A key challenge in these studies is the lack of large scale cascade data. As online social networks have recently become a primary way for people to share and disseminate information, the massive amount of data available on these networks provides unprecedent opportunities to study cascades at a large scale. Studying cascades in online social networks will benefit a variety of domains such as social campaigns [39], product marketing and adoption [28], online discussions [16], sentiment flow [29], URL recommendation [31], and meme tracking [17].

Problem Statement. The goal of this paper is to develop a model for cascade size prediction in online social networks. Specifically, *given the first τ_1 edges in a cascade, we want to predict whether the cascade will have a total of at least τ_2 ($\tau_2 > \tau_1$) edges over its lifetime without any a priori information.* This prediction has many real-world applications. For example, media companies can use it to predict social media stories that can potentially go viral [18], [31]. Furthermore, solving this problem enables early detection of epidemic outbreaks and political crisis. Despite its importance, this specific problem has not been adequately addressed in prior literature.

Predicting the sizes of social network cascades is technically challenging from many aspects. For instance, real-world cascades sometimes have large sizes, containing thousands of nodes and edges [12], [25]. Besides, cascade size prediction without any a priori information about the users has to rely solely on the shape and structural information of initial cascade propagation.

Limitations of Prior Art. A lot of prior work has studied the characteristics of cascades in online social networks. For example, Dow *et al.* studied the anatomy of two large photo sharing cascades in Facebook [12]. Kwak *et al.* investigated the audience size, tree height, and temporal characteristics of the cascades in a Twitter data set [25]. These properties of cascades are important; however, they are far from being sufficient for accurate cascade size prediction.

Some prior work has also proposed models to capture various aspects of cascades in online social networks. For example, Galuba *et al.* proposed cascade propagation models to predict which users a likely to mention which URLs [15]. Sadikov *et al.* investigated the estimation of the sizes and depths of information cascades with missing data [35]. Gomez *et al.* developed a generative model based on the maximum likelihood estimation of preferential attachment process to simulate synthetic discussion cascades [16]. However, little work has focused on developing models to predict the sizes of cascades in online social networks.

Proposed Approach. In this paper, we use a Multi-order Markov Model (M^3) for cascade size prediction in online social networks. The key insight behind our proposed approach is that large and small cascades have different initial propagation characteristics such as shape and structure. Our proposed model aims to capture these differences by automatically extracting distinguishing graph signatures that can be used to discriminate between large and small cascades.

M^3 has three key components: a cascade encoding algo-

ISBN 978-3-901882-94-4

rithm, cascade modeling method, and cascade classification algorithm. The cascade encoding algorithm uniquely encodes the shape and structure of a cascade for quantitative representation. It encodes a cascade by first performing a traversal on the cascade graph and then compressing the traversal results using run-length encoding. The cascade modeling method models the run-length encoded sequence of a cascade as a discrete random process. This random process is further modeled as a Markov chain, which is then generalized into a multi-order Markov chain model. Finally, the states of the multi-order Markov chain model are used as features to train a supervised classification algorithm for cascade size prediction.

Experimental Evaluation. We evaluate the effectiveness of our proposed cascade size prediction scheme on a real-world data set collected from Twitter containing more than 8 million tweets, involving more than 200 thousand unique users. The results show that our Markov model based cascade size prediction scheme consistently achieves more than 90% prediction accuracy in different experimental scenarios. We also compare M^3 based prediction scheme with a baseline prediction scheme based on several cascade graph features such as edge growth rate, degree distribution, clustering, and diameter. The results show that M^3 allows us to achieve significantly better prediction accuracy than the baseline scheme.

Key Contributions. In this paper, we propose the first cascade size prediction scheme based on a multi-order Markov model. In summary, we make the following key contributions in this paper.

1) We propose M^3 to quantitatively characterize and model cascades with arbitrary structures, shapes, and sizes.
2) We use M^3 for cascade size prediction in online social networks. Our evaluation using a real-world Twitter data set shows that our proposed scheme consistently achieves more than 90% prediction accuracy and outperforms baseline prediction scheme which is based on cascade graph features.

II. Related Work

Cascades in online social networks have attracted much attention and investigation. Below we summarize the prior work related to characterization and modeling cascades in online social networks.

A. Characterization

Zhou *et al.* studied Twitter posts (*i.e.*, tweets) about the Iranian election [39]. In particular, they studied the frequency of pre-defined shapes in cascades. Their experimental results showed that cascades tend to have more width than depth. The largest cascade observed in their data has a depth of seven hops. Leskovec *et al.* studied patterns in the shapes and sizes of cascades in blog and recommendation networks [26], [27]. Their work is also limited to studying the frequency of fixed shapes in cascades.

Kwak *et al.* investigated the audience size, tree height, and temporal characteristics of the cascades in a Twitter data set [25]. Their experimental results showed that the audience size of a cascade is independent of the number of neighbors of the source of that cascade. They found that about 96% of the cascades in their data set have a height of 1 hop and the height of the biggest cascade is 11 hops. They also found that about 10% of cascades continue to expand even after one month since their start. Romero *et al.* specifically studied Twitter cascades with respect to hashtags in terms of degree distribution, clustering, and tie strengths [33]. The results of their experiments showed that cascades from diverse topics (identified using hashtags), such as sports, music, technology, and politics, have different characteristics. Similarly, Rodrigues *et al.* studied structure-related properties of Twitter cascades containing URLs [31]. They studied cascade properties like height, width, and the number of users for cascades containing URLs from different web domains.

Dow *et al.* studied the anatomy of photo sharing cascades in Facebook [3]. They found that most cascades have broadcast structure, *i.e.*, most reshares are at a depth of 1 hop from the source. They also showed that large cascades, with comparable sizes, can have different temporal evolution, repeated exposure, branching factors, and user demographics. Recently, Cheng *et al.* studied the problem of prediction cascades using a bucket list of content, structural, and temporal features [8]. We evaluate and and compare to their structural features for baseline comparison. Note that we could not compare to their Facebook platform specific content-based features (e.g., fraction of positive emotion words in the caption) features.

These and similar structural properties of cascades are important; however as we show later in our experimental evaluation, they are far from being sufficient for accurate cascade size prediction.

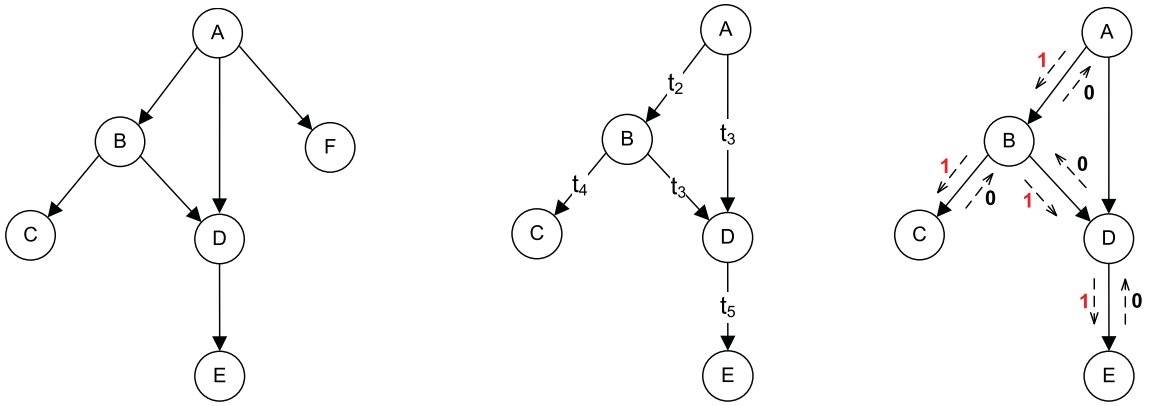
B. Modeling

Sadikov *et al.* investigated the estimation of the sizes and depths of information cascades with missing data [35]. Their estimation model uses multiple features including the number of nodes, the number of edges, the number of isolated nodes, the number of weakly connected components, node degree, and non-leaf node out-degree. Their empirical evaluation using a Twitter data set showed that their model accurately estimates cascade properties for varying fractions of missing data. However, it is not clear how this model can be effectively used for cascade size prediction.

Gomez *et al.* studied the structure of discussion cascades in Wikipedia, Slashdot, Barrapunto, and Meneame using features solely based on the depth and degree distribution of cascades [16]. They also developed a generative model based on the maximum likelihood estimation of preferential attachment process to simulate synthetic discussion cascades. However, their model is limited to the generation of synthetic discussion cascades.

III. Proposed Approach

In this section, we present M^3, a multi-order Markov chain based model for cascade size prediction in online social networks. It consists of three major components. The first component encodes a given cascade graph for quantitative



(a) Follower Graph (b) Cascade (c) Depth First Tree

Fig. 1. Toy example of cascade construction and encoding.

representation such that its structural information is retained. The second component models the encoded sequence using a multi-order Markov chain. Finally, the states of the multi-order Markov chain are used as features to train a supervised classification algorithm for cascade size prediction. Before we describe these components, we first present the details of the cascade graph construction process.

A. Cascade Graph Construction

A social network can be represented using two graphs, a relationship graph and a cascade graph. Both graphs share the same set of nodes (or vertices) V, which represents the set of all users in a social network. A *relationship graph* represents the relationships among users in a social network. In this graph, nodes represent users and edges represent the relationships among users. If the edges are directed, where a directed edge from user u to user v denotes that v is a follower of u, then this graph is called a *follower graph*, denoted as $(V, \overrightarrow{E_f})$, where V is the set of users and $\overrightarrow{E_f}$ is the set of directed edges. If the edges are undirected, where an undirected edge between user u and user v denotes that u and v are friends, then this graph is called a *friendship graph*, denoted as (V, E_f), where V is the set of users and E_f is the set of undirected edges. By the nature of our study, we focus on the follower graph denoted as $G_f = (V, \overrightarrow{E_f})$. The *cascade graph* represents the dynamic activities that are taking place in a social network (such as users sharing a URL or joining a group). A cascade graph is an acyclic directed graph denoted as $G_c = (V, \overrightarrow{E_c}, T)$ where V is the set of users, $\overrightarrow{E_c}$ is a set of directed edges where a directed edge $e = (u, v)$ from user u to user v represents the propagation of something from u to v, and T is a function whose input is an edge $e \in \overrightarrow{E_c}$ and output is the time when the propagation along edge e happens.

While the static relationship graph is easy to construct from a social network, the dynamic cascade graph is non-trivial to construct because there maybe multiple propagation paths from the cascade source to a node. So far there is no consensus on cascade graph construction in prior literature. In this paper, we use a construction method that is similar to the method described in [35]. We next explain our construction method through a Twitter example. Consider the follower graph in Figure 1(a). Let (u, t) denote a user u performing an action, such as posting a URL on u's Twitter profile, at time t. Suppose the following actions happen in the increasing time order: (A, t_1), (B, t_2), (D, t_3), (C, t_4), (E, t_5), where $t_1 < t_2 < t_3 < t_4 < t_5$. Suppose (A, t_1) denotes that A posts a URL on his Twitter profile, and all other actions (namely (B, t_2), (D, t_3), (C, t_4), and (E, t_5)) are reposting the same URL from A.

The cascade graph regarding the propagation of this URL is constructed as follows. First, A is the root of the cascade graph because it is the origin of this cascade. Second, B reposting A's tweet (which is a URL in this example) at time t_2 must be under A's influence because there is only one path from A to B in the follower graph in Figure 1(a). Therefore, in the cascade graph in Figure 1(b), there is an edge from A to B with time stamp t_2. Note that each repost (or retweet in Twitter's terminology) contains the origin of the tweet (A in this example). Third, however, D reposting A's tweet at time t_3 could be under either A's influence (because there is a path from A to D in the follower graph in Figure 1(a) and $t_1 < t_3$) or B's influence (because there is a path from B to D in the follower graph as well and $t_2 < t_3$). Note that even if D sees A's tweet through B's retweet, the repost of A's tweet on D's profile does not contain any information about B and only shows that the origin of the tweet is A. In this scenario, we assume that D is partially influenced by both A and B, instead of assuming that D is influenced by either user B or A, because this way we can retain more information with respect to the corresponding follower graph. Therefore, there is an edge from A to D and another edge from B to D in the cascade graph shown in Figure 1(b), where the time stamps of both edges are t_3. Similarly, we add the edge from B to C with a time stamp t_4 and the edge from D to E with a time stamp t_5 in the cascade graph.

B. Cascade Encoding

The first step in cascade encoding is to encode the constructed cascade graph as a binary sequence that represents the structure of the cascade graph. Graph encoding has been studied for a wide range of problems across several domains such as image compression and DNA profiling [19], [30]. The general goal of graph encoding is to transform large geometric data into a succinct representation for efficient

storage and processing. However, our goal here is to encode a given cascade graph in a way that its structural information is captured. Towards this end, we use the following graph encoding algorithm inspired by Dyck Path encoding [37].

We first conduct a depth-first traversal of the constructed cascade graph starting from the root node, which results in a spanning tree. To result in a unique spanning tree, at each node in the cascade graph, we sort the outgoing edges in the increasing order of their time stamps, *i.e.*, sort the outgoing edges $e_1, e_2, \cdots, e_k$ of a node so that $T(e_1) < T(e_2) < \cdots < T(e_k)$; and then traverse them in this order. For each edge, we use 1 to encode its downward traversal and 0 to encode its upward traversal. Figure 1(c) shows the traversal of the cascade graph in Figure 1(b) and the encoding of each downward or upward traversal. The binary encoding results from this traversal process is 11011000. Let C represent the binary code of a cascade graph $G = (V, \overrightarrow{E})$. Then the length of the binary code $|C|$ is at most twice the size of the edge set $|\overrightarrow{E}|$, *i.e.*, $|C| \leq 2|\overrightarrow{E}|$. Furthermore, let $C[i]$ be the i-th element of the binary code and $I(C[i])$ be an indicator function so that $I(C[i]) = 1$ if $C[i] = 1$, and $I(C[i]) = -1$ if $C[i] = 0$. Because each edge is exactly traversed twice, one downward and one upward, we have:

$$\sum_{i=1}^{|C|} I(C[i]) = 0.$$

The second step in cascade encoding is to convert the binary sequence, which is obtained from the depth-first traversal of the cascade graph, into the corresponding run-length encoding. A *run* in a binary sequence is a subsequence where all bits in this subsequence are 0s (or 1s) but the bits before and after the subsequence are 1s (or 0s), if they exist. By replacing each run in a binary sequence with the length of the run, we obtain the run-length encoding of the binary sequence [22]. For example, for the binary sequence 11011000, the corresponding run-length encoding is 2123.

Intuitively, using run-length encoding with depth-first traversal based encoding allows us to capture the branching characteristics of a cascade graph. We also tried breadth-first traversal based encoding, but it did not capture similar information that would be effective later in cascade classification. Our proposed encoding method successfully captures the branching characteristics of cascade graphs, while being simple to implement. It is noteworthy that our proposed framework can also be used with other suitable encoding methods.

C. Markov Model

We further want to model cascade encoding to capture the characteristics of cascades so that they can be used to identify the similarities and differences among different types of cascades (*e.g.*, large vs. small cascades). This model should allow us to extract structural features for different types of cascades and then use these features to classify them. Below, we first present our model and then demonstrate its usefulness in classifying cascades.

Consider the run-length encoded sequence $\hat{C}$ of a cascade graph G. We can model this sequence using a discrete random process $\{\hat{C}_k\}$, $k = 1, 2, ..., |\hat{C}|$. Basic analysis of this process reveals that there is some level of dependencies among the consecutive symbols emitted by the random process. In other words, it would be unreasonable to assume that the process is independent or memoryless. Meanwhile, to balance between capturing some of the dependencies within the process and to simplify the mathematical treatment of this encoded sequence, we resort to invoking the Markovian assumption [6]. As we discuss later, this assumption can be reasonably justified by analyzing the autocorrelation function of the underlying process $\{\hat{C}_k\}$. For a first order Markov process, this implies the following assumption: $Pr[\hat{C}_n = c_n|\hat{C}_1 = c_1, \hat{C}_2 = c_2, ..., \hat{C}_{n-1} = c_{n-1}] = Pr[\hat{C}_n = c_n|\hat{C}_{n-1} = c_{n-1}]$. Equivalently:

$$Pr[c_1, c_2, ..., c_n] = Pr[c_1]Pr[c_2|c_1]...Pr[c_n|c_{n-1}]. \quad (1)$$

In other words, we invoke the Markovian assumption about the underlying cascade process and its shape and structure, which is represented by the encoded sequence $\hat{C}$. Given the Markovian assumption with homogeneous time-invariant transition probabilities, $\hat{C}$ can be represented using a traditional Markov chain. The Markov chain framework allows us to quantify the probability of an arbitrary sequence of states by using Equation 1. Each element of the state transition matrix of a Markov chain is equivalent to a sub-sequence of $\hat{C}$, which in turn is equivalent to a subgraph of the corresponding cascade. We can generalize a Markov chain model by incorporating multiple consecutive transitions as a single state in the state transition matrix, which will allow us to specify arbitrary sized subgraphs of cascades. Such generalized Markov chains are called multi-order Markov chains and are sometimes referred to as full-state Markov chains. The order of a Markov chain represents the extent to which past states determine the present state.

Autocorrelation is an important statistic for selecting appropriate order for a Markov chain model [6]. For a given lag t, the autocorrelation function of a stochastic process, X_m (where m is the time or space index), is defined as:

$$\rho[t] = \frac{E\{X_0 X_t\} - E\{X_0\}E\{X_t\}}{\sigma_{X_0}\sigma_{X_t}}, \quad (2)$$

where $E(\cdot)$ represents the expectation operation and σ_{X_i} is the standard deviation of the random variable at time or space lag i. The value of the autocorrelation function lies in the range $[-1, 1]$, where $|\rho[t]| = 1$ indicates perfect correlation at lag t and $\rho[t] = 0$ means no correlation at lag t. The order of Markov chain model is generally selected equal to the largest non-negative lag for which the value of autocorrelation function jumps out of the 95% confidence envelope [24].

The number of possible states of a Markov chain increase exponentially with an increase in the order of the Markov chain model. For the n-th order extension of a Markov chain with k states, the total number of states is k^n. For a set of cascade encoding sequences, let $\mathbb{T}$ denote the set of selected orders

as per the aforementioned criterion. We select the maximum value in $\mathbb{T}$, denoted by T_{max}, as the order of a single Markov chain model that we want to employ.

D. Cascade Classification

We now show how to use the aforementioned Markov chain model for cascade classification.

1) Feature Selection: The essence of our modeling approach is to capture the shape and structure of a cascade through the states of the multi-order Markov model. Each state in the Markov chain represents a likely sub-structure of cascades. Thus, we can use these states to serve as underlying features that can be used to characterize a given cascade and to determine the class that it might belong to. However, as mentioned earlier, the number of states in a Markov chain increase exponentially for higher orders and so does the complexity of the underlying model. Furthermore, higher order Markov chains require a large amount of training data to identify a subset of states that actually appear in the training data. In other words, a Markov chain model trained with limited data is generally sparse. Therefore, we use the following two approaches to systematically reduce the number of states in the Markov chain of order T_{max}.

First, we can combine multiple states in the Markov chain to reduce its number of states. By combining states in a multi-order Markov chain, we are essentially using states from lower order Markov chains. We need to establish a criterion to combine states in the Markov chain. Towards this end, we use the concept of *typicality* of Markov chain states. Typicality allows us to identify a typical subset of Markov chain states by generating its realizations [6]. Before delving into further details, we first state the well-known typicality theorem below: For any stationary and irreducible Markov process X and a constant c, the sequence $x_1, x_2, ..., x_m$ is almost surely (n, ϵ)-typical for every $n \leq c \log m$ as $m \to \infty$. A sequence $x_1, x_2, ..., x_m$ is called (n, ϵ)-typical for a Markov process X if $\hat{P}(x_1, x_2, ..., x_n) = 0$, whenever $P(x_1, x_2, ..., x_n) = 0$, and

$$\left| \frac{\hat{P}(x_1, x_2, ..., x_n)}{P(x_1, x_2, ..., x_n)} - 1 \right| < \epsilon, \text{ when } P(x_1, x_2, ..., x_n) > 0. \quad (3)$$

Here $\hat{P}(x_1, x_2, ..., x_n)$ and $P(x_1, x_2, ..., x_n)$ are the empirical relative frequency and the actual probability of the sequence $x_1, x_2, ..., x_n$, respectively. In other words,

$$\hat{P}(x_1, x_2, ..., x_n) \approx P(x_1, x_2, ..., x_n). \quad (4)$$

This theorem shows us a way of empirically identifying typical sample paths of arbitrary length for a given Markov chain. Based on this theorem, we generate realizations (or sample paths) of arbitrary lengths from the transition matrix of the Markov chain. By generating a sufficiently large number of sample paths of a given length, we can identify a relatively small subset of sample paths that are typical. Using this criterion, we select a subset of typical states as potential features, whose lengths vary in the range $[0, T_{max}]$. In what follows, we further short-list the Markov states from the typical subset and use them as features to classify cascades.

Second, to further reduce the number of features to be employed in a classifier, we need to prioritize the aforementioned typical Markov states. The prioritization of features can be based on their differentiation power. An information theoretic measure that can be used to quantify the differentiation power of features (Markov states in our case) is information gain [9]. In this context, information gain is the mutual information between a given feature X_i and the class variable Y. For a given feature X_i and the class variable Y, the information gain of X_i with respect to Y is defined as:

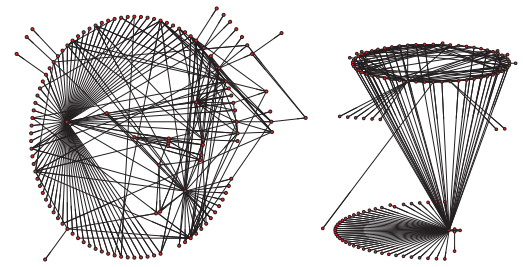
$$IG(X_i; Y) = H(Y) - H(Y|X_i), \quad (5)$$

where $H(Y)$ denotes the marginal entropy of the class variable Y and $H(Y|X_i)$ represents the conditional entropy of Y given feature X_i. In other words, information gain quantifies the reduction in the uncertainty of the class variable Y given that we have complete knowledge of the feature X_i. Note that, in this paper, the class variable Y is $\{0, 1\}$ because we apply our model to problems that require differentiating between two classes of cascades (as described later). In this study, we eventually select the top-100 features with highest information gain, as using more features did not significantly alter the results.

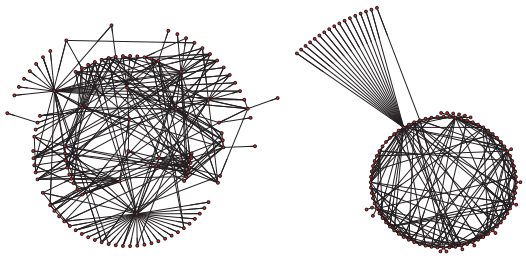
2) Classification: Let us assume that the presence of a state i is represented by a binary random variable $X_i, i = 1, 2, ..., 100$. Hence, $P(X_i = 1)$ represents the probability for the presence of state X_i. We can think of the X_is as the variables representing potential features. Thus, our training process proceeds as follows. For a given class Y of cascades, we evaluate the presence of a given feature (state) X_i in Y by analyzing a sufficiently large number of sample cascades that belong to the class Y. Subsequently, we are able to evaluate the a-priori conditional probability $P(X_i|Y)$ for each class $Y \in \{1, 2, ..., k\}$, where the number of classes k is usually very small. In our case, we are interested in the traditional binary classifier with $k = 2$. However, note that this classification methodology can be extended to the cases with $k > 2$ using the well-known one-against-one (pairwise) or multiple one-against-all formulations [20].

We can jointly use multiple features to differentiate between two sets of cascades belonging to different classes. In particular, given the top features with respect to information gain, we can classify cascades by deploying a machine learning classifier. In this study, we use a Bayesian classifier to jointly utilize the selected features to classify cascades. Naïve Bayes is a popular probabilistic classifier that has been widely used in the text mining and bio-informatics literature, and is known to outperform more complex techniques in terms of classification accuracy [38]. It trains using two sets of probabilities: the prior, which represents the marginal probability $P(Y)$ of the class variable Y; and the a-priori conditional probabilities $P(X_i|Y)$ of the features X_i given the class variable Y. As previously explained, these probabilities can be computed from the training set.

Now, for a given test instance of a cascade with observed features X_i, $i = 1, 2, ..., n$, the *a-posteriori* probability



(a) Example cascade # 1



(b) Example cascade # 2

Fig. 2. Visualization of real-world Twitter cascades. Radial layout on left and circular layout on right.

$P(Y|X^{(n)})$ can be computed for both classes $Y \in \{0, 1\}$, where $X^{(n)} = (X_1, X_2, ..., X_n)$ is the vector of observed features in the test cascade under consideration:

$$P(Y|X^{(n)}) = \frac{P(X^{(n)}, Y)}{P(X^{(n)})} = \frac{P(X^{(n)}|Y)P(Y)}{P(X^{(n)})} \quad (6)$$

The naïve Bayes classifier then combines the a-posteriori probabilities by assuming conditional independence (hence the "naïve" term) among the features.

$$P(X^{(n)}|Y) = \prod_{i=1}^{n} P(X_i|Y). \quad (7)$$

Although the independence assumption among features makes it feasible to evaluate the a-posteriori probabilities with much lower complexity, it is unlikely that this assumption truly holds all the time. For our study, we mitigate the effect of the independence assumption by pre-processing the features using the Karhunen-Loeve Transform (KLT) to uncorrelate them [11].

IV. Experimental Results

In this section, we evaluate the effectiveness of our proposed model for cascade size prediction in online social networks. Below, we first describe the data set used for evaluation, then define evaluation metrics, and finally discuss evaluation results.

A. Data Collection

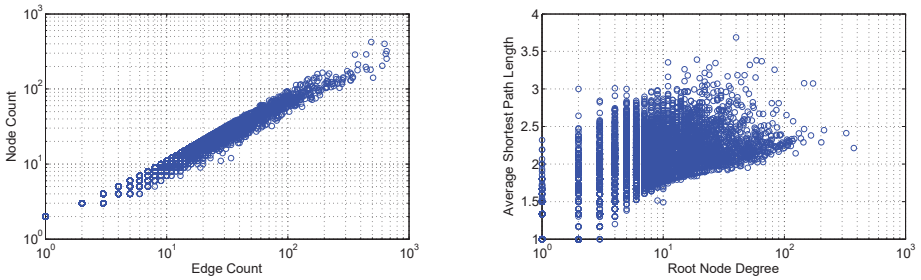
Among the popular online social networks, Twitter is one of the social networks that allows systematic collection of public data from its site. Therefore, we chose to study the shape and structure of cascades appearing on Twitter. For our study, we separately collected two data sets from Twitter. The first data set was collected using Twitter's *streaming API*, which allows the realtime collection of public tweets matching one or more filter predicates [2]. We focused on tweets related to the Arab Spring event, which represents an ideal case study because it spans several months. To collect tweet data pertaining to a country, we provided relevant keywords as filter predicates. For example, we used the keywords 'Libya' and 'Tripoli' to collect tweets related to Libya. In total, we collected tweets for 8 countries over a period of a week in March 2011. Using Twitter's streaming API, we collected more than 8 million tweets from more than 200 thousand unique users.

As mentioned in Section III-A, we cannot accurately construct cascade graphs without information about whom the users are following. Twitter provides follower information for a given user via a separate interface called REST API [2]. REST API employs aggressive rate limiting by allowing clients to make only a limited number of API calls in an hour. In our tweet data set, we encountered more than $200,000$ unique users and we were required to make at least one request per user to get the follower list. To overcome this limitation, we utilized dozens of public proxy servers to parallelize calls to Twitter's REST API. Using this methodology, we collected follower lists of all users in less than a month.

B. Data Characteristics

Twitter provides a "re-tweet" functionality which allows users to re-post the tweet of other users to their profiles. The reference to the user with original tweet is maintained in all subsequent re-tweets. There is no information on intermediate users in re-tweets. Using the follower graph, we constructed cascade graphs for all sets of re-tweets which are essentially cascades. Therefore, the overall graph is a union of all cascades in our data. In Figure 2, we visualize two cascades in our data set using the radial layout method [1]. In a radial layout, we choose the user with original tweet as a center vertex (or root vertex in general) and the remaining vertices are put in concentric circles based on their proximity to the center vertex. In Figure 2(a), we observe that the degree of vertices typically decreases as their distance from the root vertex increases. On the contrary, in Figure 2(b), we observe that subsequent vertices have degrees comparable to the root vertex. We aim is to capture such differences in an automated fashion using our proposed model. Next, We analyze the structural features of the cascades in our collected data set in terms of their degree and path properties.

We first jointly study the number of edges and the number of nodes for all cascades in our data set. The cascade graphs in our data set are connected and each user in the cascade graph has at least one inward or outward edge. Therefore, the number of edges in a cascade graph $|E|$ has the lower bound: $|E| \geq |V| - 1$, where $|V|$ is the number of users participating in the cascade. Figure 3(a) shows the scatter plot between edge and node counts for all cascades in our data set. Note that we use the logarithmic scale for both axes. From this figure, we observe that the scatter plot takes the form of a strip whose thickness represents the average number of additional edges for each node. The average thickness of this



(a) Edge and node counts (b) Root node degree and average path length

Fig. 3. Cascade graph attributes in our data set.

strip approximately corresponds to having twice the number of edges compared to the number of nodes.

1) Path Properties: Another important characteristic of a cascade is the degree of the root node (user who initiated the cascade), which typically has the highest degree compared to all other nodes in a cascade graph. In our data set, the root node has the highest degree compared to all other nodes in cascade graphs for more than 92% of the cascades. The degree of the root node essentially represents the number of different routes through which cascade propagates in an online social network. Note that these paths may merge together after the first hop; however, we expect some correlation between the degree of root node and the number of unique routes through which a cascade propagates. One relevant characteristic of a graph is average (shortest) path length (APL), which denotes the average of all-pair shortest paths [5].

$$APL = \sum_{\forall i,j \in V, i \neq j} \frac{d(i,j)}{|V|(|V|-1)},$$

where $d(i,j)$ is the shortest path length between users i and j. We expect the average path length of a cascade to be proportional to the degree of the root node. Figure 3(b) shows the scatter plot of the root node degree and the average path length. As expected, we observe that cascades with higher root node degrees tend to have larger average path lengths.

C. Evaluation Metrics

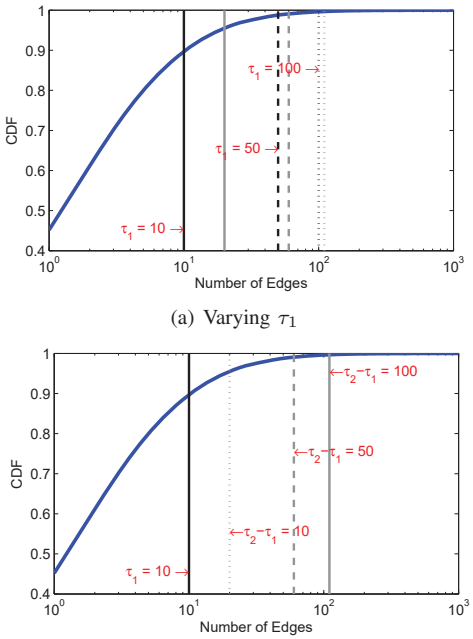
We now evaluate the classification effectiveness of M^3 in terms of the standard Receiver Operating Characteristic (ROC) metrics [13]. Below, |Positives| = |True Positives| + |False Positives| and |Negatives| = |True Negatives| + |False Negatives|.

$$\textit{Detection Rate} = \frac{|\text{True Positives}|}{|\text{True Positives}| + |\text{False Negatives}|}$$

$$\textit{False Positive Rate} = \frac{|\text{False Positives}|}{|\text{False Positives}| + |\text{True Negatives}|}$$

$$\textit{Accuracy} = \frac{|\text{True Positives}| + |\text{True Negatives}|}{|\text{Positives}| + |\text{Negatives}|}$$

To ensure that the classification results are generalizable, we divide the data set into k folds and use $k-1$ of them for training and the left over for testing. We repeat these experiments k times and report the average results in the following text. This setup is called stratified k-fold cross-validation procedure [38]. For all experimental results reported in this paper, we use the value of $k = 10$. We observed qualitatively similar results for other k values.



(a) Varying τ_1

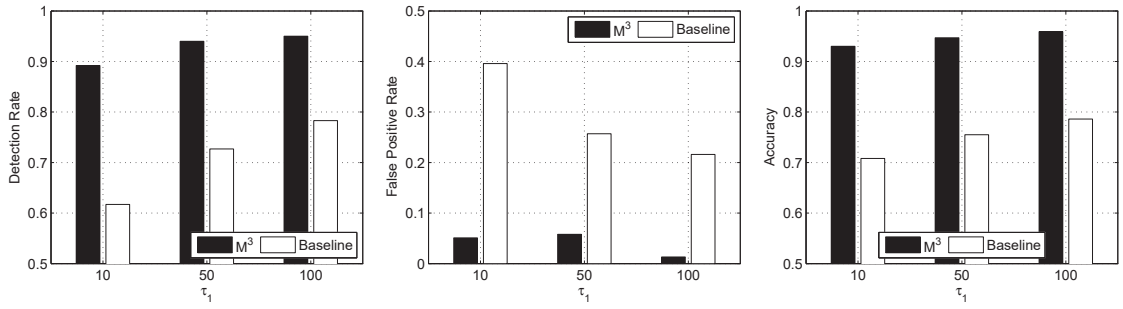
(b) Varying $\tau_2 - \tau_1$

Fig. 4. Setup for cascade size prediction.

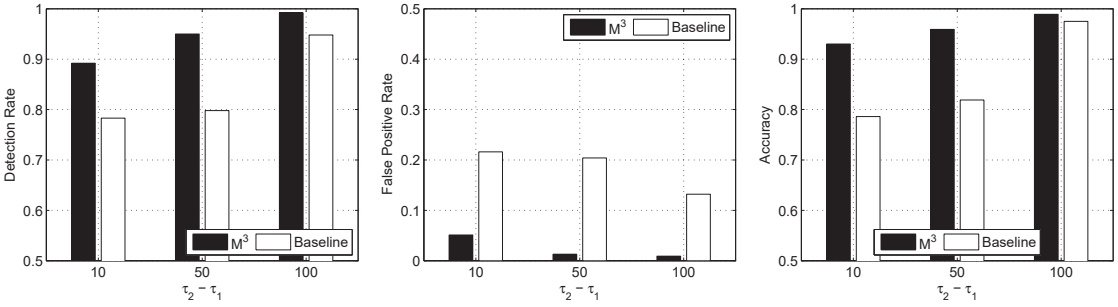
D. Discussions

We now present the evaluation results of M^3 using our Twitter data set. We compare the classification performance of M^3 based scheme with a baseline scheme that uses the following well-known graph features [4] with the Naïve Bayes classification algorithm: edge growth rate, number of nodes, degree of the root node, average shortest path length, diameter, number of spanning trees, clustering coefficient, and clique number. These features summarize the structural information of cascade graphs.

In this paper, we treat the cascade size prediction problem to an equivalent cascade classification problem: given a cascade with τ_1 edges, classify it into two classes: the class of cascades that will have less than τ_2 edges over their lifetime and the class of cascades that will have greater than or equal to τ_2 edges over their lifetime. We use the initial τ_1 edges to train both the cascade size prediction scheme based on M^3 and



(a) Varying values of τ_1 and $\tau_2 - \tau_1 = 10$



(b) Varying values of $\tau_2 - \tau_1$ and $\tau_1 = 10$

Fig. 5. Classification results of M^3 and the baseline scheme for different τ_1 and $\tau_2 - \tau_1$ values.

the baseline scheme that is based on cascade graph features. For extensive evaluation, we vary the values of τ_1 and τ_2. Because the distribution of the number of edges in our data set is skewed, *i.e.*, most cascades having only a few edges over their lifetime, the larger the values of τ_1 and $\tau_2 - \tau_1$ are, the more imbalanced the two classes are. To mitigate the potential adverse effect of class imbalance [21], we employ instance re-sampling to ensure that both classes have equal number of instances before the cross-validation evaluations. Below we discuss the classification accuracies of both schemes as we vary the values of τ_1 and τ_2.

Impact of Varying τ_1. Figure 4(a) shows the evaluation setup as we vary the values of $\tau_1 \in \{10, 50, 100\}$, while keeping $\tau_2 - \tau_1$ fixed at 10. The solid, dashed, and dotted vertical black lines corresponds to $\tau_1 = 10, 50,$ and 100. The solid, dashed, and dotted vertical grey lines all correspond to $\tau_2 - \tau_1 = 100$. The value of τ_1 impacts the classification results because it determines the number of edges in each cascade that are available for training. Therefore, larger values of τ_1 generally improve training quality of both cascade size prediction schemes and lead to better classification accuracy.

Figure 5(a) plots the detection rate, false positive rate, and accuracy of M^3 and the baseline scheme for varying $\tau_1 \in \{10, 50, 100\}$, while keeping $\tau_2 - \tau_1$ fixed at 10. Overall, we observe that M^3 consistently outperforms the baseline scheme with the peak precision of 96% at $\tau_1 = 100, \tau_2 - \tau_1 = 10s$. Generally, we observe that the classification accuracies of both schemes decreases as the value of τ_1 is increased. The standard ROC threshold plots of M^3 shown in Figure 6(a) also confirm this observation.

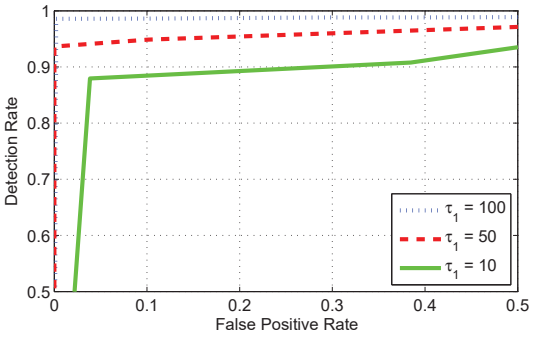
Impact of Varying $\tau_2 - \tau_1$. Figure 4(b) shows the evaluation setup as we vary the values of $\tau_2 - \tau_1 \in \{10, 50, 100\}$, while keeping τ_1 fixed at 10. The solid vertical black line corresponds to $\tau_1 = 10$. The solid, dashed, and dotted vertical grey lines correspond to $\tau_2 - \tau_1 = 10, 50,$ and 100, respectively. The value of $\tau_2 - \tau_1$ also impacts the classification results because it determines the separation or distance between the two classes. Therefore, larger values of $\tau_2 - \tau_1$ generally lead to better prediction accuracy.

Figure 5(b) plots the detection rate, false positive rate, and accuracy of M^3 and the baseline scheme for varying values of $\tau_2 - \tau_1$. Once again, we observe that M^3 consistently outperforms the baseline scheme with the peak precision of 99% at $\tau_2 - \tau_1 = 100, \tau_1 = 10$. We also observe that the classification accuracies of both methods improve as the value of $\tau_2 - \tau_1$ is increased. The standard ROC threshold plots of M^3 shown in Figure 6(b) also confirm this observation.

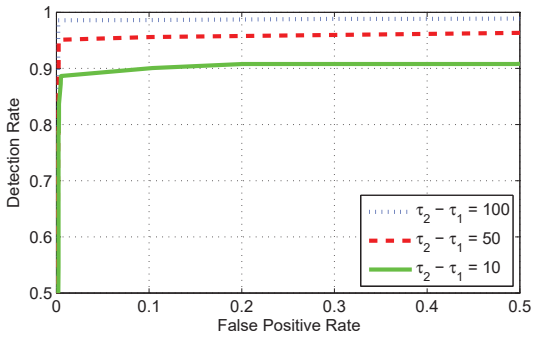
V. Conclusion

In this paper, we first propose M^3, a multi-order Markov chain based model for cascade size prediction in online social networks. The key insight behind M^3 is that large and small cascades have different initial propagation characteristics such as shape and structure. M^3 captures these differences by automatically extracting distinguishing graph signatures that can be used to discriminate between large and small cascades. The experimental results using a real-world Twitter data set showed that M^3 significantly outperforms the baseline scheme in terms of prediction accuracy. M^3 based cascade size prediction scheme consistently achieved more than 90% prediction accuracy in different experimental scenarios.

We envision future work along the following directions. First, M^3 can be used to solve other cascade classification problems that can benefit from their structural information. For example, M^3 can be used to differentiate spam and normal



(a) Varying τ_1



(b) Varying $\tau_2 - \tau_1$

Fig. 6. ROC threshold plots of M^3 for different τ_1 and $\tau_2 - \tau_1$ values.

activity cascades in online social networks. Second, we plan to explore randomized cascade encoding methods such as those based on random walks on graphs [14], [34]. Finally, we used M^3 in the context of online social networks in this paper; however, our model is generally applicable to cascades in other contexts as well such as sociology, economy, psychology, political science, marketing, and epidemiology. Applications of our model in these contexts are interesting future work to pursue.

Acknowledgment

Zubair Shafiq's work is partially supported by the National Science Foundation under Grant Numbers CNS-1464110, CNS-1524329, and CNS-1617288. Alex Liu's work is partially supported by the National Science Foundation under Grant Numbers CNS-1318563, CNS-1524698, and CNS-1421407, the National Natural Science Foundation of China under Grant Numbers 61472184 and 61321491, and the Jiangsu Innovation and Entrepreneurship (Shuangchuang) Program.

References

[1] Graphviz - graph visualization software. http://www.graphviz.org.
[2] Twitter API documentation. https://dev.twitter.com/docs.
[3] P. A. D. amd L. A. Adamic and A. Friggeri. The Anatomy of Large Facebook Cascades. In *AAAI International Conference on Weblogs and Social Media (ICWSM)*, 2013.
[4] B. Bollobas. *Modern graph theory*. Springer Verlag, 1998.
[5] A. Bondy and U. Murty. *Graph Theory*. Springer, 2008.
[6] P. Bremaud. *Markov Chains*. Springer, 2008.
[7] M. Cha, A. Mislove, and K. P. Gummadi. A measurement-driven analysis of information propagation in the Flickr social network. In *ACM WWW*, 2009.
[8] J. Cheng, L. A. Adamic, P. A. Dow, J. Kleinberg, and J. Leskovec. Can cascades be predicted? In *World Wide Web Conference (WWW)*, 2014.
[9] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.
[10] K. Dave, R. Bhatt, and V. Varma. Modelling action cascades in social networks. In *AAAI Conference on Weblogs and Social Media*, 2011.
[11] R. Dony. *The Transform and Data Compression Handbook, Chapter 1*. CRC Press, 2001.
[12] P. A. Dow, L. A. Adamic, and A. Friggeri. The Anatomy of Large Facebook Cascades. In *AAAI ICWSM*, 2013.
[13] T. Fawcett. ROC Graphs: Notes and Practical Considerations for Researchers. Technical report, HP Laboratories, 2004.
[14] D. Figueiredo, P. Nain, B. Ribeiro, E. de Souza e Silva, and D. Towsley. Characterizing continuous time random walks on time varying graphs. In *ACM SIGMETRICS/Performance*, 2012.
[15] W. Galuba, K. Aberer, D. Chakraborty, Z. Despotovic, and W. Kellerer. Outtweeting the Twitterers - Predicting Information Cascades in Microblogs. In *3rd Workshop on Online Social Networks (WOSN)*, 2010.
[16] V. Gomez, H. J. Kappen, and A. Kaltenbrunner. Modeling the structure and evolution of discussion cascades. In *ACM HT*, 2011.
[17] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *ACM KDD*, 2010.
[18] D. Gruhl, R. Guha, R. Kumar, J. Novak, and A. Tomkins. The predictive power of online chatter. In *ACM KDD*, 2005.
[19] S.-Y. Hsieha, C.-W. Huanga, and H.-H. Choub. A DNA-based graph encoding scheme with its applications to graph isomorphism problems. *Applied Mathematics and Computation*, 203:502–512, 2008.
[20] C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.
[21] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.
[22] N. S. Jayant and P. Noll. *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Prentice Hall, 1984.
[23] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *proceedings of KDD*, 2003.
[24] S. A. Khayam and H. Radha. Markov-based modeling of wireless local area networks. In *ACM Mobicom Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2003.
[25] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *ACM WWW*, 2010.
[26] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs. In *SIAM International Conference on Data Mining (SDM)*, 2007.
[27] J. Leskovec, A. Singh, and J. Kleinberg. Patterns of influence in a recommendation network. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2006.
[28] X. Li. Informational cascades in IT adoption. *Communications of the ACM*, 47(4), 2004.
[29] M. Miller, C. Sathi, D. Wiesenthal, J. Leskovec, and C. Potts. Sentiment flow through hyperlink networks. In *AAAI ICWSM*, 2011.
[30] M. Reid, R. Millar, and N. D. Black. Second-generation image coding: An overview. *Second-Generation Image Coding: An Overview*, 29:3–29, 1997.
[31] T. Rodrigues, F. Benevenuto, M. Cha, K. P. Gummad, and V. Almeida. On word-of-mouth based discovery of the web. In *ACM IMC*, 2011.
[32] E. M. Rogers. *Diffusion of Innovations*. Cambridge University Press, 2003.
[33] D. M. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on Twitter. In *ACM WWW*, 2011.
[34] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences (PNAS)*, 105(4):1118–1123, 2008.
[35] E. Sadikov, M. Medina, J. Leskovec, and H. Garcia-Molina. Correcting for missing data in information cascades. In *WSDM*, 2011.
[36] J. A. Starr and I. C. MacMillan. Resource cooptation via social contracting: Resource acquisition strategies for new ventures. *Strategic Management Journal*, 11:79–92, 1990.
[37] Y. Sun. The statistic "number of udu's" in Dyck paths. *Discrete Mathematics*, 287(1-3):177–186, 2004.
[38] I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011.
[39] Z. Zhou, R. Bandar, J. Kong, H. Qian, and V. Roychowdhury. Information resonance on Twitter: Watching Iran. In *SOMA*, 2010.