TrendFusion: Trends and Influences Among Geographically Colocated Large User Communities

Shaymaa Khater[1], Hicham G. Elmongui[1,2], Denis Gračanin[1]
[1] Department of Computer Science,Virginia Tech,USA
[2] Department of Computer and Systems Engineering, Alexandria University, Egypt
skhater@vt.edu, elmongui@alexu.edu.eg, gracanin@vt.edu

Abstract

Trending topics are words or phrases that are frequently mentioned in social media. Trends differ from one geographical location to another, and often reflect the most popular events that happen in the world. Limited work has been done to analyze the relation between trends and geography.However,the current information diffusion models are not suitable at the geographical locations level as they were targeting users. In this paper, we present TrendFusion, an innovative model used to analyze and predict the localized diffusion of trends in social networks. In other words, we try to answer the following questions: Can we predict in which locations will a trend be appearing in the future? Can we predict when will it happen?

We demonstrate this by comparing our model to the widely accepted General Threshold model. Results show that TrendFusion accurately predicts places in which a trend will appear, with 98% recall and 80% precision.

1 Introduction

A plethora of users embracing the online social media has greatly impacted the way how information propagates. The online social media give the users the ability to create, discuss and disseminate information. As information diffuses from one user to another, some topics become of interest to only small groups of users, thus do not become widely adopted and could fade away quickly. On the other hand, some topics are of interest to many users, who in turn influence more and more users. When this snowball effect reaches a sufficiently large group or community of users, a topic is considered a trending topic. Most of the time the trending topic can be discovered by a certain word or phrase that is repeated in most of the users interactions.

Studying trending topics in online social networks is crucial since these topics represent the ideas and interesting themes that are currently discussed by a large number of users and, more or less, the general public. Trending topics can be observed at a global level or can be localized to the users in certain locations, such as the residents of a certain city. Studying localized trending topics helps to understand the interests of residents of the targeted local area. So far, trends have been studied as a way to detect real-world events discussed in social media [5, 13], emerging topics [41], or news of interest for the online community. On the other side, geography plays an important role in various aspects of our lives. However, proliferation of online social networks has significantly decreased the virtual distance between the users. Yet, geographical locality still matters in our choice of friends [40], as well as topical interests [25].

Our assumption is that as users influence each other at the personal level, users in one location have a collective mutual influence over users in another location [10]. This is consistent with the first law of geography [24], "*Everything is related to everything else, but near things are more related than distant things*". Based on this assumption, we developed TrendFusion, a model for predicting localized trends diffusion from one localized community of users to other geographically separated communities of users. We conducted our experiments on the trending topics collected from Twitter. Accordingly, we show that observing the local trends for some locations can enable predicting other locations where these trends will appear. Other trending topics prediction models rely on analyzing the activities of the users to be able to predict a trending topic. Thus the prediction can only be in range of few hours before the topic becomes a trending one.

The most important aspect of TrendFusion is that it allows predicting trends that will appear in some location, before even the users in that location start mentioning that topic. This is extremely useful in many cases, such as marketing, choosing areas to target in political campaigns, building a proactive localized recommendation system for topics or for early prediction of protests and strikes.

Our contributions are as follows:

1. Deriving a new information diffusion model (*Snowball Cascade Model*) in social networks that is suitable to model the diffusion between geographically separated communities in social networks, rather than relying on the users social network structure. The model is then extended to account for more than one explanatory variable.

2. Developing TrendFusion, a predictive model that predict whether the trending topics will appear for some location in the future, along with the activeness time, i.e., the time it will appear.

The remainder of the paper is organized as follows. First we give an overview of the related work in Section 2. A complete description of the model and its components will be described in Section 3 followed by the experimental results in Sections 4 and 5. Finally, conclusions and future work are provided in Section 6.

2 Related Work

We first describe information propagation in social networks and then discuss the work related to the study of trending topics.

2.1 Information and Influence propagation in Social Networks

In recent years, information propagation on social networks has been attracting much attention in academic and industrial circles [30]. Understanding the mechanisms of information propagation is vital to finding the factors affecting the information propagation process. These factors, in turn, provide a better explanation for predicting information popularity [4], and initiating a viral marketing campaign [12]. This can be formulated as inferring and estimating the propagation probability that a piece of information propagates from one individual to another along social links connecting them.

Previous research had identified two factors that affect the information propagation process: the importance of the information, and the level of interactions between users. The existing research approaches for the first factor mainly consider the analysis of the messages propagation and the decay with respect to the time since the posting of the message [22]. Most of these approaches are descriptive. However, our approach is predictive.

For the second factor, the level of interactions between users, the current research efforts focus on the interactions between the users, along with the geographic, demographic, topical and contextual features that affect the propagation between the users [2, 14, 22]. For example, Galuba et al. [16] proposed a propagation model that predicts which users will tweet about which URL based on the history of past user activity. Agarwal et al. [1] studied the problem of identifying influential bloggers in the blogosphere.

As our model analyzes and predicts the localized diffusion of trends in social networks between locations, our work is different in that it doesn't take into account the social structure of the social networks. It is prohibitively complex to include the social structure connections relating the locations. Another point is that the location information posted by the user is not always available or accurate. Our work is also different from the research that studies relationship between geography and information diffusion, as in [9], as our model consider other non-geographical parameters.

2.2 Trends in Social Networks

Trending topics in Twitter are words and phrases, appearing on the main page of Twitter, that are currently popular in users' tweets, and are identified for the past hour, day and week. They represent the popular topics of conversations among the Twitter users [28]. Trends in social networks have recently been a focus of interest for many researchers.

Some researches focused on studying trends from a temporal view [28, 29]. Kwak et al. compared the trending topics generated by Twitter to other social media [28]. Leskovec et al. also studied the temporal properties of information shared in social networks by tracking memes across the blogosphere [29]. Some researches were interested in studying the structural nature of the social graph that leads to creating the trends [6, 8].

Other studies focused on studying the dynamics, the growth and the decay of the trending topics [3, 43]. Asur et al. studied the trending topics on Twitter, and provided a theoretical basis for the factors affecting the formation, persistence and decay of trends [3].

Limited work has been done to analyze the relation between trends and geography. A closely related work was presented by Kamath et al. [26] and Ferrara et al. [15]. Kamath et al. modeled the social media spread in different locations by trying to predict the top K cities in which a topic will be trending [26].

Ferrara et al. investigated the spatial and geographic dynamics that govern trending topics in Twitter. However their goal was different, as they aimed at studying what dynamics underlie the production and consumption of trends in different geographic areas In other words, they wanted to know if trends travel through the Internet, or by people physically traveling across cities [15].

3 TrendFusion Framework

In this section, we first describe some of the main concepts and definitions, followed by a detailed explanation for our model.

TrendFusion model relies on the information cascade concept to represent the flow of a piece of information, usually called the contagion, through a social network [11]. The cascade is usually represented as a directed acyclic graph (DAG). Figure 1 represents an example of information cascade, where
Nodes: are the entities (such as users, groups or cities). The nodes represent locations in our model.
Edges: represent the information propagation between entities.
Seeds: are the vertices that initiate the cascade.
An activation step (or a step): Every time a given trend appears at the same time at one or more entities corresponds to an activation *step*, or simply a step, in the cascade.
A Cascade: is a sequence of activation steps generated by a contagion process. The weights on the edges represent the influence of an active entity on an inactive one. The way to calculate these influences and how an inactive node responds to them are specific to each model.

The two main objectives of TrendFusion are:

1. Predict whether a trend will appear for some location based on its diffusion in other locations.

2. Predict when the trend will appear.

The problem we are trying to solve can be defined as:

- Given: A history of spatially and temporally tagged trending topics in a number of locations.

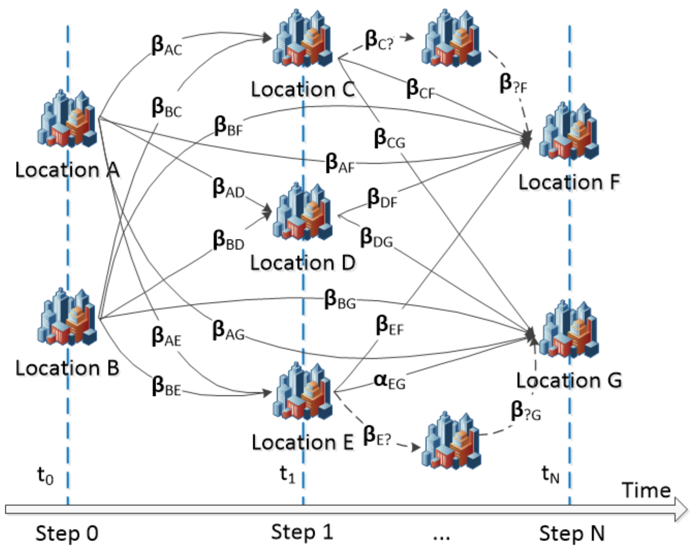


Figure 1: An information cascade represented by a Directed Acyclic Graph (DAG).

- Processing: Define a model that can extract and capture the dependency relations between these locations.
- Output: When a topic is trending in some locations, use the model to predict where and when this topic will be trending next.

3.1 TrendFusion Model

We now provide the statistical foundations of the TrendFusion model.

Generally, most information diffusion models assume that the considered entities (such as users, groups, etc.) are connected by a social graph, and that the graph structure is known beforehand. In our case, there is no such social graph connecting the locations together. Thus, before applying any known diffusion model, we need first to infer the underlying *hypothetical* graph that describes the influence between locations. Fortunately, several network inference models have been developed recently [19, 20, 33, 34, 38]. These algorithms estimate the underlying network structure given past activation times.

In TrendFusion model, we assume a fully connected graph, and estimate the transmission rates along the edges using NetRate algorithm [19]. We based our assumption of the fully connected graph on the first law of geography by Tobler [24]. We start with a fully connected graph of the locations and estimate the transmission rate between each pair of locations using NetRate. The lowest transmission rates are then omitted reducing the edges (connections) between the locations.

NetRate algorithm estimates the transmission rates, not just a binary on/off value. The algorithm takes the input in the form of information cascades. The NetRate algorithm relies on the survival theory and the concept of *hazard rate* that will be explained shortly [21].

3.2 Generating the Hazard Rate Graph

After converting the activations to different cascades of trends between locations, we compute the pairwise hazard function between these locations. The hazard rate is mostly related to the survival theory [21], and can be described as the instantaneous activation rate between two locations i and j [19], i.e., how likely is it that location j will adopt a trend at time t_j, if location i adopted that trend at time t_i (Equation 1).

$$H(t_j|t_i;\lambda_{i,j}) = \frac{f(t_j|t_i;\lambda_{i,j})}{S(t_j|t_i;\lambda_{i,j})} \tag{1}$$

where $f(t_j|t_i)$ is the conditional likelihood of transmission from location i to location j. Likelihood depends on the activation times t_i and t_j (i.e, the time the trend first appears in location i and location j), and a pairwise transmission rate $\lambda_{i,j}$. The transmission rate $\lambda_{i,j}$ models the strength of an edge (i,j), and determines how frequently information spreads from location i to location j. The most commonly used parametric models for the shape of the conditional transmission likelihood are the exponential, power-law, and Rayleigh distributions models [21]. $S(t_j|t_i)$ in Equation 1 refers to the survival function computed for the edge connecting the locations i and j. It is computed as the probability that location i does not cause location j to activate by time t_j as in Equation 2:

$$S(t_j|t_i;\lambda_{i,j}) = 1 - F(t_j|t_i;\lambda_{i,j}) \tag{2}$$

where $F(t_j|t_i)$ denotes the cumulative density function computed from the transmission likelihoods. The generated graph is used in our model as a measure for the influence between locations.

3.3 TrendFusion stages

TrendFusion consists of five stages. The first three stages can be shared across the locations of interest. Stages four and five should be repeated for each location. The five stages of TrendFusion model are shown in Figure 2.

3.3.1 Stage 1: Collect Trends From Locations

Trends should be collected from all the locations of interest. The trends are continuously collected every Δt time units. If social media does not reveal the localized trending topics, an extra step of monitoring user activities and extracting the trending topics is needed.

3.3.2 Stage 2: Store Trending Topics Stream

As the stream of the trending topics is received, they are labeled by the location/time they were received from/at. The trending topics are stored for further analysis.

3.3.3 Stage 3: Build Cascades

Since trending topics are continuously polled every fixed time step, it is not always clear if a trend is a beginning of a new cascade or a continuation of an old one. Therefore,

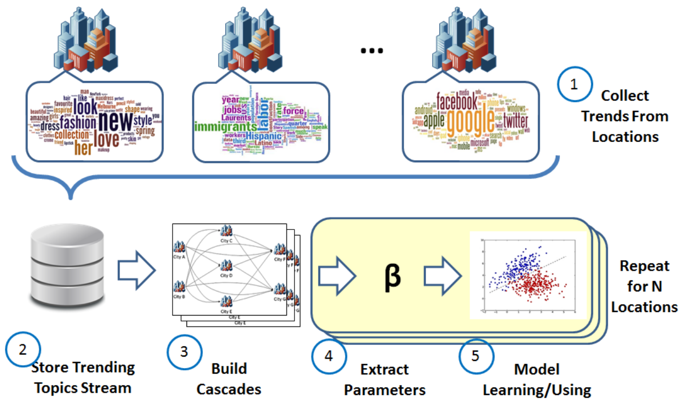


Figure 2: The stages of TrendFusion model.

a process is needed to build cascades from trending topics that are retrieved every Δt. Algorithm 1 provides the details of the cascades building process. The process begins by chronological ordering of all received spatially and temporally tagged trends (Activations List), where one activation represents a record of (*trend*, *location* and *time*). The algorithm first determines if an activation should be part of an earlier cascade or it should be considered as a seed for a new cascade. Ferrara et al. [15] state that the life time of almost all trends does not exceed 24 hours. Thus we consider a trend to be a seed for a new cascade if it was not trending for more than 24 hours. The algorithm then determines whether or not to consider this activation as a new step. If the location did not appear before in the cascade, then this is a new step. Otherwise, this is considered an update to the location activity times.

3.3.4 Stage 4: Extract Parameters

This stage is done for each location. In a given cascade, every location that appears in that cascade will have a distinctive set of parameters. The parameters are calculated mainly based on the diffusion model used as will be explained in sections 3.4 and 3.5. For example, an average distance parameter will be calculated between a given location and all its parents or ancestors depending on the diffusion model. There are four main classes of the parameters:

- Diffusion Parameters (Hazard rate): is the value representing the activation rate between any two locations calculated over all cascades.
  - Maximum hazard (*max_hazard*).
  - Sum of hazards (*sum_hazard*).
- Geographical Parameters: these features are used to examine the geospatial properties of the trending topics spread.
  - Geographical distance between locations (*shrt_dist*): indicates the shortest distance between locations and whether these distances affect the appearance of trends in these locations. For this, we have used the Haversine distance, which is commonly used to measure the distance between locations based on the spherical shape of the Earth (as compared to Euclidian distance) [37]. Average distance between locations (*avg_dist*) are also calculated.
  - Coverage (*cvr*): is a spread over geographical area of a trend S at time t. The area which the trend covers is determined by getting the area of bounding box in which the trend appeared. For the bounding box area, we determined the bounding locations (north east, north west, south east, south west) in which each trend appear. We then calculated the area using the Haversine distance between the boundaries

Algorithm 1 Build Cascades

```
Procedure BuildCascadesFromActivations
Input ActivationsList $al$
begin
  // An activation $a$ is a record $a = (trend, location, time)$
  ActivationsList $alo \leftarrow$ Order $al$ by time
  for all Activation $a$ in $alo$ do
    if $a.trend$ appeared in ($a.time$ - 24 hours) then
      $cas \leftarrow$ last cascade of $a.trend$
      if $a.location$ appeared in $cas$ then
        Add $a.time$ to instances of $a.location$ in $cas$
      else if $a.time$ equals time of last step in $cas$ then
        Add $a$ to last step of $cas$
      else
        Add new step to $cas$ containing $a.location$
      end if
    else
      Create new cascade $cas$
      Add new step to $cas$ containing $a.location$
    end if
  end for
end
```

- Historical Parameters: these parameters describe the path characteristic of each trend through all locations. Their values are based on previous cascades.
  - Trending topics similarity between locations (sim_{tt}) [26]: the similarity parameter is used to measure the trending topics similarity between locations. For measuring the similarity, we used the Jaccard coefficient between the sets of trends observed at each location, as shown in Equation 3:

$$sim_{tt}(location_i, location_j) = \frac{|M_{location_i} \cap M_{location_j}|}{|M_{location_i} \cup M_{location_j}|} \quad (3)$$

where $M_{location_i}$ is the set of trends appeared in $location_i$. A similarity score of 1 means that all trends are common between the two locations. A score of 0 means that no trends are in common between the two locations. Average similarity is calculated over all trends.

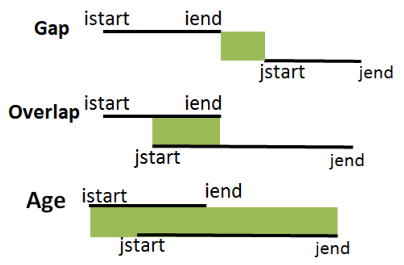


Figure 3: Time tracking of trends' appearances in locations i, j.

  - Average gap (*avg_gap*): for each trend appearing between two locations, the gap is calculated as the time difference between the end time in location i and its appearance in location j. It is calculated over all trends.
  - Overlap time: for two locations i and j the overlap time is calculated as the difference between trend's end time in location i and its appearance in location j, given that ($t_{i.end} > t_{j.start}$). Average overlap time is generated over all cascades.
  - Average trend age (*avg_age*): quantifies the average time of trend's appearance within the social network.

  Figure 3 illustrates calculating time differences between trends' appearances in locations.

- Trend Parameters: these features include information about the relationship between locations based on the current cascade.
  - Trend's rank (*sum_rank*): as Twitter provides a trends box that contains the top 10 trending topics, ranked according to their popularity. The trend's rank differ when the trend list is updated every 5 minutes.
    - Maximum rank (*sum_rank*): the highest rank reached by each trend in each cascade. The sum of trend's ranks over all cascades is also computed.
    - Weighted sum of trend's rank (*weighted_sum_rank*): indicates whether or not the trend's rank has effect on the transmission rate. It is calculated as a sum of trend's ranks multiplied by the hazard rate between two locations.
  - Number of parents / ancestors (*num_parents* / *num_ancestors*): the number of parents and ancestors' locations for each location/cascade.

3.3.5 Stage 5: Model Learning/Using

As locations are different, a distinct predictive model is needed for each location. The model should learn the parameters extracted from the previous stage and should be used to predict if a new cascade will appear in that location. For this, we utilized two diffusion models. We first use the widely used General Threshold model (GT) as our baseline, and then we present our information diffusion model, the Snowball Cascade (SC) model.The differences between the two models will be described in details in the coming sections.

3.4 The General Threshold (GT) Model

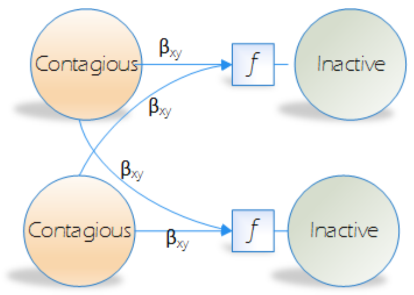
One widely used cascade model is the Generalized Threshold model (GT) [27, 35]. Its primary focus is on the information diffusion within users in a social network. Conceptually, as any other information diffusion cascade model, the GT model tries to predict whether or not a certain piece of information will get adopted by different nodes in a social network. Generally, there are three types of nodes: active, contagious, and inactive. Given a piece of information, *inactive* nodes are those nodes that did not adopt that information yet, *active* nodes are the nodes that adopted it already, and the *contagious* nodes are the nodes that are trying to influence other nodes of adopting it. Once a node is activated, it will remain active till the end of the cascade.

Initially, other than the seed nodes, all the other nodes are considered inactive. The seed nodes are those nodes that initially introduce that information to the network. At the beginning of the cascade, seed nodes are activated. Once a set of nodes is activated, they become contagious and only try once to collectively influence other inactive nodes. Once they are done, they are no longer contagious but they remain active, i.e., they will no longer try to influence other nodes.

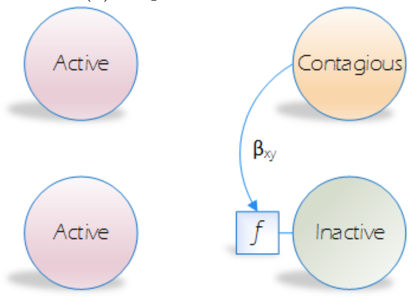
Figure 4 shows an example of two steps for four nodes. In Figure 4a, two nodes are contagious, both trying to influence the two inactive nodes. The βs on the edges are influence rates between the corresponding nodes. The function box in the GT model is just a summation operation followed by a condition to check that the sum is below a certain threshold. The threshold is a specific property of each node, i.e. the threshold is different from one node to the other. If the sum exceeds that threshold, the node becomes contagious; as in the second step shown in Figure 4b. The second step shows that one of the inactive nodes, got infected and became contagious itself, and the other one was not affected. The two contagious nodes in step one, became active in step two. This means that they are already infected but will not try to influence other node anymore.

Formally, the GT model is described as follows. Consider a directed graph $G = (V, E)$, where V is the set of vertices representing locations, and E is the set of weighted edges, with weights w_{uv} representing the influence rate of the edge $e_{uv} \in E$ from location u to location v. Let N_v be the set of vertices with edges going into v, and S_t be the subset of N_v that are active at time t. For every vertex v there is an activation function $f()$, such that at time t, if $f(S_t) > \theta_v$, vertex v becomes active at time $t + 1$. In the original model, the value of θ_v is randomly chosen from a uniform distribution in the interval [0, 1]. In our evaluation, we rely on statistical classifiers to estimate the likelihood value of θ_v.

Although the GT model was performing well in modeling the information diffusion between users, we find that it was

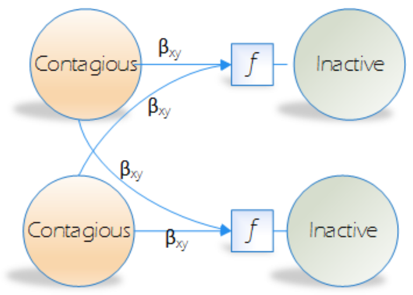


(a) Step 1 in GT model

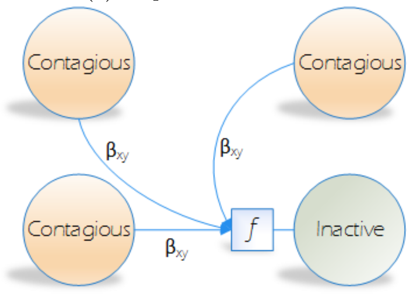


(b) Step 2 in GT model

Figure 4: Steps of General Threshold model



(a) Step 1 in SC model



(b) Step 2 in SC model

Figure 5: Steps of Snowball cascade model

not suitable to model the diffusion between locations with large communities of users (i.e., cities), as will be demonstrated later by our results. Thus, we are proposing a new cascade model (*Snowball Cascade Model*) that generalizes the GT model. The new model, along with the differences between the GT model and our model are described in details in Section 3.5.

3.5 The Snowball Cascade Model

The central part of TrendFusion is a new cascade model, Snowball Cascade (SC) Model. We consider that the assumption in the GT model that the influence of a node (location) only happens one time is unrealistic. Thus, in the SC model, unlike the GT model, an activated node will always be contagious, i.e., it will keep trying to influence other nodes.The rational behind the continuous influence is simple: As long as a topic is trending in a location, this interest can affect other locations. The GT model will fail to differentiate between the influence of quick and outstretched trends. Thus in the SC model, the number of active nodes in the system that are trying to spread the influence will grow over time. Active nodes try to influence other nodes which, if activated, become contagious and try to influence other nodes, and so on. This snowball effect is the reason behind the model name.

Yet another difference between the two models is that in SC model, the edge weights are vectors rather than scalars. The vector values change from one activation to the other. This is different from the GT model, where the edge weights are required only to be fixed scalars. The vectors on the edges represent the set of parameters that might affect the influence between a contagious location and inactive location at a given step of a cascade.

The SC model is also different from the well known VARMA model [31]. As the VARMA model is used for analysis and predicting future values of a time-series. Unlike the VARMA model, the SC model is an information diffusion model, not a multi-variate time series model. The SC model defines how the nodes influence each other at every time step. It also defines how to combine different parameter values from active nodes to construct the parameter vector. In fact, many parameters, e.g. the distance, in the influence vectors cannot be regarded as a time-varying processes.

Figure 5 shows an example of a cascade in SC model. The $\boldsymbol{\beta}$ values on the edges represent vectors containing the influence rates along with other parameters that are described in 3.3.4. The main difference between Figures 4 and 5 is in the second step, where in the SC model, the contagious nodes remain contagious, and keep on trying to influence other inactive nodes till the end of the cascade.

Formally, the SC model can be described as follows. Consider a directed graph $G = (V, E)$, where V is the set of vertices representing locations, and E is the set of weighted edges, with weights $\boldsymbol{\beta}^t_{uv}$ of edge $e_{uv} \in E$ representing the influence rate from location u to location v at time step t. Let N_v be the set of vertices with edges going into v, and S_t be a subset of N_v with the vertices that are active on or before time t. For every vertex v there is an activation function $f()$, such that at time t, if $f(\boldsymbol{\beta}^t_{u_0 v}, \boldsymbol{\beta}^t_{u_1 v}, \ldots, \boldsymbol{\beta}^t_{u_n v}) > \theta_v \; \forall u_i \in S_t$, vertex v becomes active at time $t+1$. The value of θ_v can be learned for each location by a binary classifier.

According to the definitions of the two models, the GT model can be considered as a special case of the SC model, where the $\boldsymbol{\beta}$ vectors are reduced to a fixed scalar (infleunce rate), and the $\boldsymbol{\beta} = 0$, for all nodes that are already active before time t.

4 Evaluation

We now describe the methodology used to generate our dataset, and then we describe in details the results of every stage in our model.

4.1 Trending Topics Dataset

To build our dataset we used Twitter APIs [39] to collect all trending topics appearing on Twitter for a period of 30 days, starting from August 2014 until September 2014, in 48 of the most populated US locations (cities). Twitter provides a trends box that contains the top 10 trending hashtags or phrases at any given moment, ranked according to their popularity. These trending topics, along with their rank is updated every 5 minutes. Each user can monitor the trends at the worldwide, country, or city level.

We deployed a crawler to get the trends every 5 minutes for the 48 cities. We also collected all trends reported by Twitter for the United States and the whole world. To mask the effect of global trends in our experiments, we filtered out the trends for the cities, that appeared in the U.S. trends or the global trends. We ended up collecting more than 400K different trends. The data is stored as tuples of the form: (woeid, $trend_0, trend_1, \ldots, trend_9$, date/time) where *woeid* is Yahoo Where On Earth ID (WOEID) [44] and $trend_0, \ldots, trend_9$ are the top 10 trends. Figure 6 shows the histogram of the distances between the 48 cities, where the x-axis represent the upper limit of each distance bin in miles.

4.2 Applying TrendFusion Framework Steps

As mentioned earlier, the steps presented in Algorithm 1 are used to convert the data collected from previous step into cascades. We then use the MATLAB®implementation of NetRate algorithm [18] to build the influence graph for all locations. This implementation assumes linear DAG for cascades, i.e., it assumes that each step in the cascade consists only of one location. However, the Snowball Cascade model allows multiple locations per cascade step. So the algorithm was modified slightly to account for this difference. The modified NetRate is used to generate three graphs, one for each assumed distribution for the hazard rate. The graphs from NetRate are then used with the cascades to generate the training and testing examples for each location.

For each location, we generate the training file containing the examples for the first 22 days of the data and a testing file containing the remaining data. The extracted parameter was based on the Snowball Cascade model. We also used the GT model as a baseline, so training and testing data was also generated for it. Each of the parameter vectors is augmented by one class and one dependent variable.

Given a cascade, when generating the training examples for a given location, an example is generated for each step in the cascade before that location appears in it. For example, if a location appeared in step n, we generate $n-1$ examples for each step before that location appeared. If the location doesn't appear in the cascade, then the number of examples generated will be equal to the number of steps in the cascade. The class values are set to be the *appearing* or *not appearing*, depending on whether or not the location appeared in the cascade. If the class value is appearing, then the dependent variable value is set based on the lag value between the time at the cascade step and the time the trend appeared in the location.

We used the parameter vectors for each cascade for individual trends to train three classifiers:

- Logistic Regression (LR), a probabilistic statistical classification model [32].
- Stochastic Gradient Descent (SGD) classifier [17].
- Random Forest (RF), ensemble learning method classifier [7].

We used Weka [23] and R [36] statistical packages to train the three classifiers and afterwards use them to predict whether or not the trend will appear in the designated location.

4.3 Experiments

The evaluation includes five experiments:

1. Predict trends based on individual steps.
2. Predict trends considering each cascade as a whole.
3. Determine the effect of each parameter on the classification process.
4. Determine the average time a topic can be predicted to be trending before it actually becomes trending.
5. Predict when a trend will appear.

5 Results and Discussion

We evaluated the performance of TrendFusion by running our training and testing examples through the three classifiers. Each example represents a step in a cascade. We used the widely adopted GT model as a baseline to compare its performance with TrendFusion. We recorded two quality measures in our experiments, *recall* and *precision*.

Here *recall* indicates the ratio of the number trends we were able to predict to the total number of actual trends. Similarly, *precision* indicates the quality of our prediction, i.e., the ratio of the number of topics that actually become trending in our predictions to the total number of topics we predicted will be trending.

In the first experiment, we considered the output from each individual example. This means that at each step, we take a decision regardless of other steps in the cascades. Figure 7 shows the recall and precision values obtained by TrendFusion and the GT models using the three classifiers. It is clear that TrendFusion was giving the same performance across the different classifiers with a recall value of around 0.71 and precision around 0.84. This means on average 84% of the predicted trend will be actually trending.

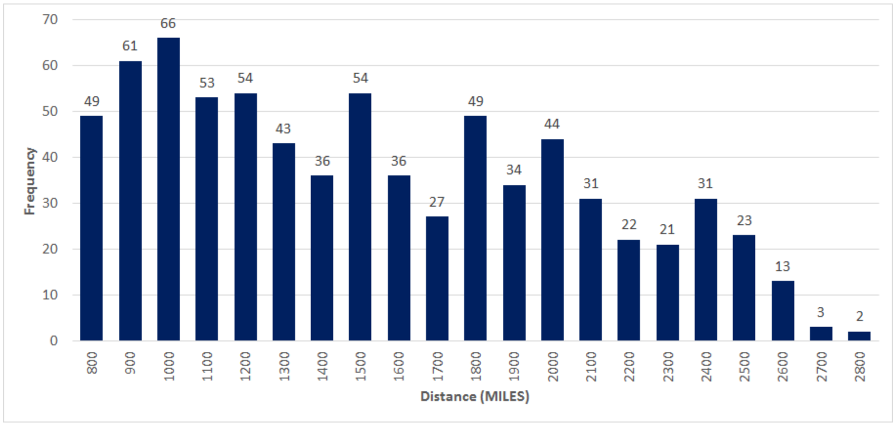


Figure 6: Histogram of the distances between the 48 cities.

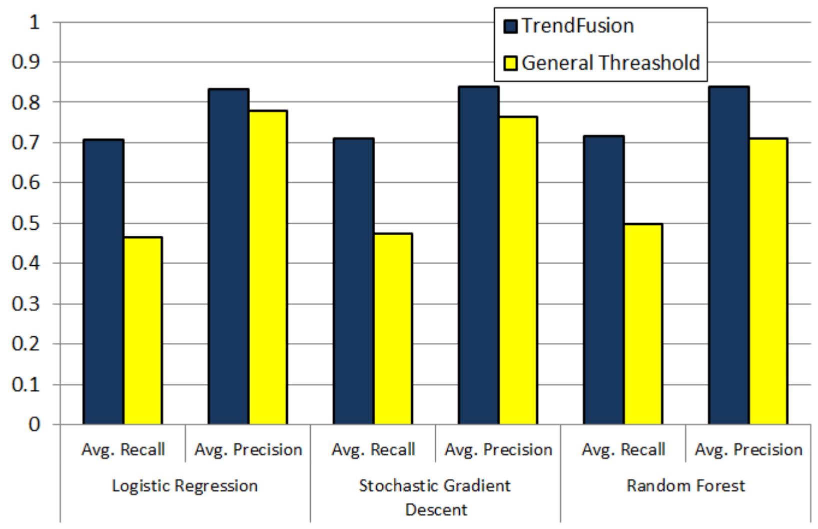


Figure 7: Average precision and recall for TrendFusion and GT models considering cascade steps.

On the other hand, the GT model recall values were in the range between 0.47, 0.48 and 0.5 for the LR, SGD and RF classifiers respectively, which means that it misses around half of the trends. The precision values were 0.78,0.75 and 0.71 for the same three classifiers.This means that the slight increase in the recall was accompanied with more false positive predictions.The results therefore show that the GT model is not suitable for modeling the diffusion of trending topics between locations.

In the second experiment, we evaluated each cascade as a whole, getting one decision for the whole cascade. For a given location, we set the class value to be *appearing* for the cascades in which the location appeared, and *not appearing* for the cascades in which the location didn't appear. The classification is performed on each step, then the predicted values are reduced to one value for the whole cascade. If the predicted value at any of the steps is *appearing*, we consider the combined prediction as *appearing*, as if doing a *logical OR*. The reason behind this way of classification is that the *class* is assigned at each step based on the fact whether or not the location appeared later in the cascade. So at an early step in reality, that might not have any influence on a given location that appeared later in the cascade, the class is still assigned as *appearing*. This is due to the fact that we do not have ground truth data.

A false positive prediction is considered to be made in a cascade where a given location didn't appear, if at any step an *appearing* class is predicted. The logic of this classification process is detailed in Algorithm 2.

Figure 8 shows the average recall and average precision values for the TrendFusion and GT models for the second experiment with the same three classifiers as before. The average recall values for TrendFusion improved greatly. This means that in the first experiment, TrendFusion made wrong *not appearing* predictions at the beginning of the cascades that are neutralized in this experiment by a later correct *appearing* prediction. Values for *recall* are 0.96, 0.98 and 0.99 for LR, SGD and RF classifiers, respectively.

On the other hand, *precision* dropped slightly to around 0.8 for the LR and SGD classifiers and to 0.71 for the RF classifier. This also means that one wrong *appearing* prediction at any step of cascade in which a given location did not appear, will cause the overall prediction to be considered wrong. Although, the average recall is slightly improved for the GT model, it still in the range of 0.5 to 0.56 for the three classifiers. The average precision also dropped as expected to the values of 0.65, 0.65 and 0.51 for LR, SGD and RF classifiers, respectively. This still point out that

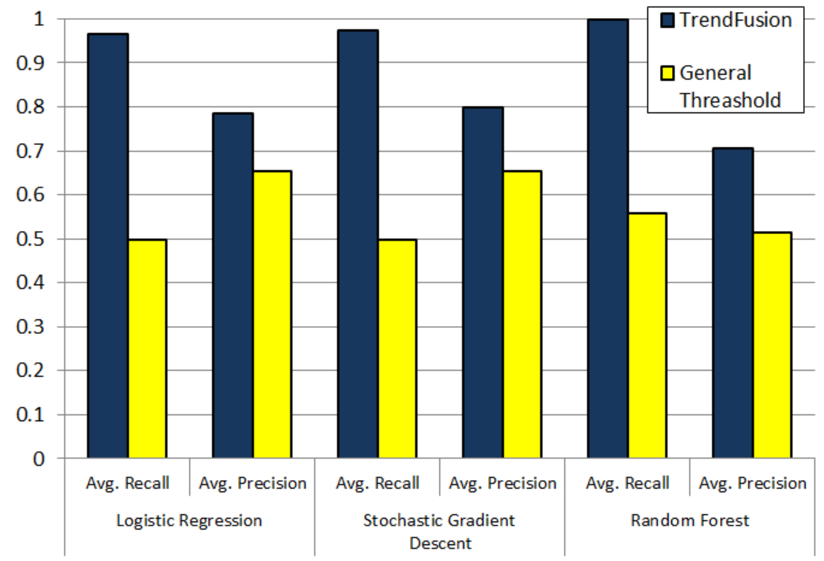


Figure 8: Average precision and recall for TrendFusion and GT models considering all cascades.

even though that the GT model was good in modeling information diffusion in a social graph at the users level, it is not suitable to model the trending topics diffusion between locations.

These two experiments were conducted using the transmission rates generated by the modified NetRate algorithm assuming exponential distribution. We also examined the two distribution models (power-law and Rayleigh) to decide the shape of the conditional transmission likelihood, and to analyze the effect of changing them on the classification process. The experiments were repeated using the other two distributions. The results were very consistent with the results obtained for exponential distribution. The variation in the results obtained in all experiments did not exceed 1%.

The third experiment was conducted to measure the effect of each parameter on the classification process. This is achieved by ranking all the parameters according to their average information gain. Figure 9 shows the rank of each parameter used in the classification process.

We observed the following:

- Geography matters: it is clear from Figure 9 that locations that are geographically near each other are most likely to influence each other in the social context.

- The similarity in interests and diffusion parameters are also of high importance: locations that are similar in the trending topics in the past, are more likely to have the same trends later on. Also, cities with high combined diffusion rate to a given city, will have high probability to affect it.

- Trend parameters are the least important: although locations may be influencing each other, the rank of the trending topic in one location is not affecting its rank in the other location. This might be due to the fact that each location has different interests in topics. This also means that it does not really matter in how many locations did a topic appear in, to be influential to other locations, it might just give an indication of how globally important is that topic.

- The remaining parameters were equally important.

The fourth experiment explored the average time a topic can be predicted to be trending before it actually becomes trending. Figure 10 shows the average time before a trend can appear. The x-axis represents the lag time between the beginning of the cascade and the time a trend will occur. The y-axis represent the time before a trend is predicted as trending. This shows that we are able to predict the topics on average 3 hours before they actually trend.

In Figure 10, we noticed a drop at value 17 of x-axis. We investigated the possible reasons for this drop. We found that the number of trends that appeared in new cities after 17 hours are relatively much less than different hours. To find out the reason for that, we used the facts presented by Upbin [42] that shows the average Twitter activity by hour. Upbin showed that the user activity is highest between 9 AM and 2 PM, and lowest between 1 AM and 6 AM. According to this, We assumed that most trends are formed during the high activity intervals. The first four horizontal lines in the Figure 11 represent different timezones in the US. The upper represent Eastern time, then Central, Mountain, and finally Pacific. The red peaks represent high activity time at each timezone. The blue troughs represent low activity intervals. The lower line represent the combined activities, and it shows that the highest activity in the US happens around 1 PM Eastern, and the lowest activity happens around 6 AM Eastern. The difference between these numbers is 17 hours, thus the trends will not be trended within this gap, hence the drop in number of trends that happen after 17 hours.

Algorithm 2 Classify Cascade

```
Procedure ClassifyCascade
Input Location $l$
      Cascade $cas$
begin
  // Determine the class for the whole cascade
  $count_{true\_positive} \leftarrow 0$
  $count_{false\_positive} \leftarrow 0$
  $count_{true\_negative} \leftarrow 0$
  $count_{false\_negative} \leftarrow 0$
  if $l$ appears in $cas$ then then
    $class \leftarrow appearing$
  else
    $class \leftarrow notappearing$
  end if

  // Collective classification for all steps
  for all step $s$ in $cas$ do
    $prediction \leftarrow$ classify_at($s$)
    if $prediction$ is $appearing$ then
      if $class$ is $appearing$ then
        $count_{true\_positive} \leftarrow count_{true\_positive} + 1$
      else
        $count_{false\_positive} \leftarrow count_{false\_positive} + 1$
      end if
      return
    end if
  end for

  // At this stage, $prediction$ should be $not\ appearing$
  if $class$ is $not\ appearing$ then
    $count_{true\_negative} \leftarrow count_{true\_negative} + 1$
  else
    $count_{false\_negative} \leftarrow count_{false\_negative} + 1$
  end if
end
```

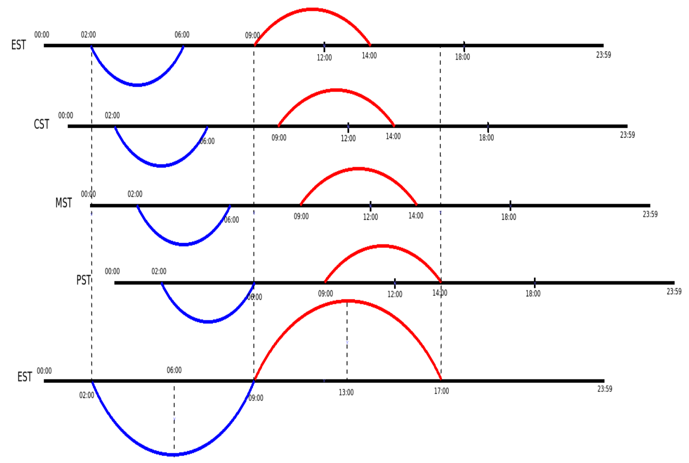


Figure 11: Activity times over 24 hours for users on Twitter. Red: highly active window, Blue: low active window.

In the fifth experiment we tried to predict when a trend will appear. The training and testing examples in this case are labeled by the time lag between each step and the step at which the trend appeared in a given city. We trained a linear

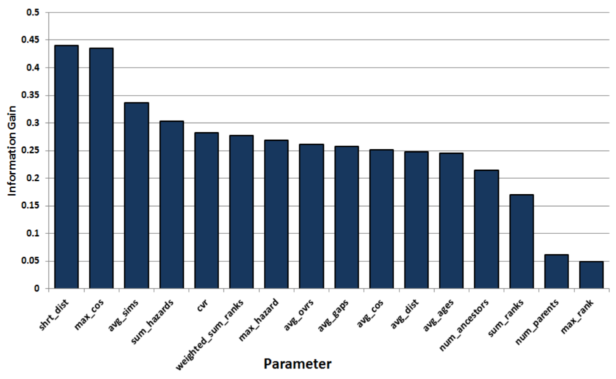


Figure 9: Rank of each parameter used in the classification process.

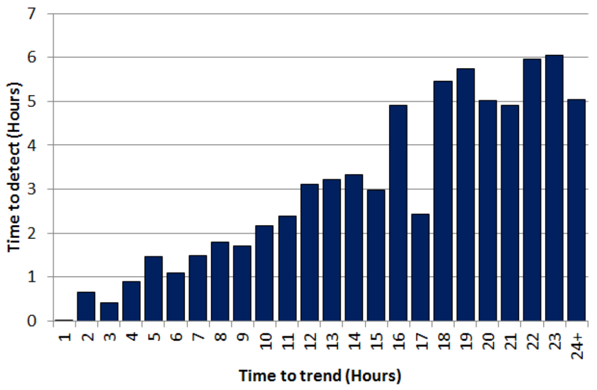


Figure 10: Lag analysis for predicted trends.

regression model and used it to try to predict when will the trend happen. Figure 12 shows a histogram where the bins (x-axis) represent the error in prediction in hours. The results shows that most of the predictions were around zero error. The bimodal peaks is probably due to the activity windows described in Figure 11, where the high activity interval makes the trends travel faster, and the low activity window makes the trends be delayed in traveling.

6 Conclusion

We proposed TrendFusion, a model for predicting the localized trends diffusion in social networks. Our goal was to develop a model that will allow us to predict whether a trend will be appearing on some location in the future, and if it will appear, when it would appear. We showed that the diffusion models designed for modeling information spread between users are not suitable for modeling trends diffusion across locations, where no real friendship relations exist. The main aspect of TrendFusion is a new information cascade model, Snowball Cascade (SC) model. The model assumes that an activated node in a graph will always be contagious.

We applied our proposed models on trending topics obtained from Twitter for 48 of biggest US cities. We demonstrated the effectiveness of our model and compared it to the General Threshold (GT) model, a widely accepted diffusion

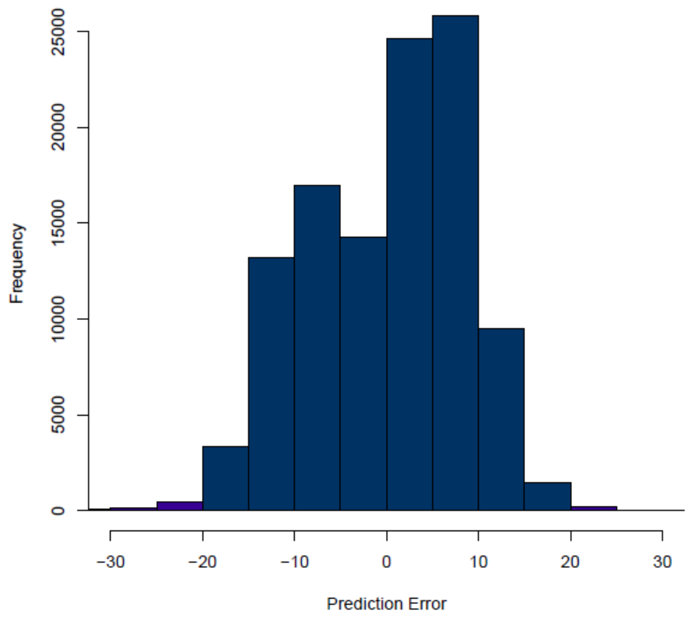


Figure 12: Prediction error histogram

model. TrendFusion outperformed GT model by achieving the recall and precision of prediction of trends by 98% and 80%, respectively.

TrendFusion is also capable of predicting the time at which the trend will appear. TrendFusion successfully predicted trends before they actually become trending by up to 24 hours. The root mean squared error (RMSE) in TrendFusion time prediction is less than six hours.

References

[1] N. Agarwal, H. Liu, L. Tang, and P. S. Yu. Identifying the influential bloggers in a community. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 207–218, New York, 2008. ACM.

[2] S. Aral and D. Walker. Identifying influential and susceptible members of social networks. *Science*, 337(6092):337–341, 2012.

[3] S. Asur, B. A. Huberman, G. Szabó, and C. Wang. Trends in social media : Persistence and decay. *CoRR*, abs/1102.1402, 2011.

[4] P. Bao, H.-W. Shen, J. Huang, and X.-Q. Cheng. Popularity prediction in microblogging network: A case study on sina weibo. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 177–178, 2013.

[5] H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on twitter. In *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[6] S. Bharathi, D. Kempe, and M. Salek. Competitive influence maximization in social networks. In *Proceedings of the 3rd International Conference on Internet and Network Economics*, WINE'07, pages 306–311, Berlin, Heidelberg, 2007. Springer-Verlag.

[7] G. Biau. Analysis of a random forests model. *J. Mach. Learn. Res.*, 13(1):1063–1095, Apr. 2012.

[8] C. Budak, D. Agrawal, and A. El Abbadi. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 665–674, New York, 2011. ACM.

[9] E. Casetti. Innovation Diffusion as a Spatial Process, by Torsten Hägerstrand. *Geographical Analysis*, 1:318–320, 1969.

[10] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *in ICWSM 10: Proceedings of international AAAI Conference on Weblogs and Social*, 2010.

[11] W. Chen, L. V. Lakshmanan, and C. Castillo. Information and influence propagation in social networks. *Synthesis Lectures on Data Management*, 5(4):1–177, 2013.

[12] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 199–208, New York, 2009. ACM.

[13] A. Crooks, A. Croitoru, A. Stefanidis, and J. Radzikowski. #earthquake: Twitter as a distributed sensor system. *Transactions in GIS*, 17(1):124–147, 2013.

[14] W. Feng and J. Wang. Retweet or not?: Personalized tweet re-ranking. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 577–586, New York, 2013. ACM.

[15] E. Ferrara, O. Varol, F. Menczer, and A. Flammini. Traveling trends: Social butterflies or frequent fliers? *CoRR*, abs/1310.2671, 2013.

[16] W. Galuba, K. Aberer, D. Chakraborty, Z. Despotovic, and W. Kellerer. Outtweeting the twitterers - predicting information cascades in microblogs. In *Proceedings of the 3rd Wonference on Online Social Networks*, WOSN'10, 2010.

[17] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 69–77, 2011.

[18] M. Gomez-Rodriguez. Netrate implementation, 2014. [Online; accessed July 2015].

[19] M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pages 561–568, 2011.

[20] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. *TKDD*, 5(4):21, 2012.

[21] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf. Modeling information propagation with survival theory. *CoRR*, abs/1305.3616, 2013.

[22] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 241–250, New York, 2010. ACM.

[23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.

[24] B. Hecht and E. Moxley. Terabytes of tobler: Evaluating the first law in a massive, domain-neutral representation of world knowledge. In *Spatial Information Theory*, volume 5756 of *Lecture Notes in Computer Science*, pages 88–105. Springer Berlin Heidelberg, 2009.

[25] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsiouliklis. Discovering geographical topics in the twitter stream. WWW '12, pages 769–778, New York, 2012. ACM.

[26] K. Y. Kamath, J. Caverlee, Z. Cheng, and D. Z. Sui. Spatial influence vs. community influence: Modeling the global spread of social media. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 962–971, 2012.

[27] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 137–146, 2003.

[28] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600, 2010.

[29] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 497–506, New York, 2009. ACM.

[30] J. Leskovec, M. Mcglohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs. In *SDM*, 2007.

[31] H. Lutkepohl. Chapter 6 forecasting with {VARMA} models. volume 1 of *Handbook of Economic Forecasting*, pages 287 – 325. Elsevier, 2006.

[32] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, 1 edition, 1997.

[33] S. A. Myers and J. Leskovec. On the convexity of latent social network inference. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Canada.*, pages 1741–1749, 2010.

[34] P. Netrapalli and S. Sanghavi. Learning the graph of epidemic cascades. *SIGMETRICS Perform. Eval. Rev.*, 40(1):211–222, June 2012.

[35] N. Pathak, A. Banerjee, and J. Srivastava. A generalized linear threshold model for multiple cascades. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 965–970, Dec 2010.

[36] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. [Online; accessed June 2015].

[37] R. W. Sinnott. Virtues of the Haversine. *Sky and Telescope*, 68(2):159+, 1984.

[38] T. M. Snowsill, N. Fyson, T. De Bie, and N. Cristianini. Refining causality: Who copied from whom? In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 466–474, New York, 2011. ACM.

[39] TwitterAPI_v1.1. https://dev.twitter.com/. [Online; accessed June 2015].

[40] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the facebook social graph. *CoRR*, abs/1111.4503, 2011.

[41] S. Unankard, X. Li, and M. Sharaf. Location-based emerging event detection in social networks. In *Web Technologies and Applications*, volume 7808 of *Lecture Notes in Computer Science*, pages 280–291. 2013.

[42] B. Upbin. What are the best times to share on facebook and twitter? http://www.forbes.com/sites/bruceupbin/2012/05/09/when-to-make-stuff-go-viral-online [Online; accessed June 2015].

[43] C. Wang and B. Huberman. Long trend dynamics in social media. *EPJ Data Science*, 1(1), 2012.

[44] Yahoo!-GeoPlanet. "https://developer.yahoo.com/geo/geoplanet/". [Online; accessed July 2015].