Spatio-Temporal Meme Prediction: Learning What Hashtags Will Be Popular Where

Krishna Y. Kamath
Texas A&M University
College Station, TX 77843
kykamath@cs.tamu.edu

James Caverlee
Texas A&M University
College Station, TX 77843
caverlee@cse.tamu.edu

ABSTRACT

In this paper, we tackle the problem of predicting what online memes will be popular in what locations. Specifically, we develop data-driven approaches building on the global footprint of 755 million geo-tagged hashtags spread via Twitter. Our proposed methods model the geo-spatial propagation of online information spread to identify which hashtags will become popular in specific locations. Concretely, we develop a novel reinforcement learning approach that incrementally updates the best geo-spatial model. In experiments, we find that the proposed method outperforms alternative linear regression based methods.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Information networks*

Keywords

social media; spatial impact; spatio-temporal analysis

1. INTRODUCTION

The widespread adoption of GPS-enabled tagging of social media content provides new access to the fine-grained spatio-temporal logs of user activities. For example, the Foursquare location sharing service has enabled 2 billion “check-ins” [10], whereby users can link their presence, notes, and photographs to a particular venue. The mobile image sharing service Instagram allows users to selectively attach their latitude-longitude coordinates to each photograph; similar geo-tagged image sharing services are provided by Flickr and a host of other services. And the popular Twitter service sees 500 million Tweets per day, of which around 5 million are tagged with latitude-longitude coordinates.

With access to the worldwide geo-spatial footprints of social media users, we focus on the problem of *predicting what online memes will be popular in what locations*, which has important implications for a variety of systems and applications, including targeted advertising, location-based services, social media search, and content delivery networks. In particular, we focus our investigation on a sample of 755 million geo-tagged Tweets with precise latitude-longitude coordinates collected over the course of 18 months. Specifically we consider the propagation of hashtags across Twitter, where a hashtag is a simple user-generated annotation prefixed with a #. Hashtags serve many purposes on Twitter, from associating Tweets with particular events (e.g., #ripstevejobs and #fukushima) to sharing memes and conversations (e.g., #bestsportsrivalry and #ifyouknowmeyouknow).

Our goal is to develop techniques based on Twitter hashtag propagation which can be used to predict hashtags that will be popular at any location. For example, can we accurately predict which hashtags will be popular in San Francisco over the next two hours? Can the same model also predict which hashtags will be popular in a small town like College Station, Texas? Can we identify which hashtags that have been popular in New York in the past two hours but will drop in interest? Building robust models that can accurately predict the spatio-temporal popularity of online memes like hashtags can aid in design of systems and applications, including content delivery networks, social media search, location-based services like Google Now, and geo-targeted advertising.

Toward answering these questions, we develop in this paper a reinforcement learning-based approach that builds upon two competing hypotheses of information spread over geo-spatial networks.

- **Spatial Affinity:** The first hypothesis, based on the Tobler’s first law of geography [23], states that the information spread between two locations is impacted by the distance between two locations. For example, according to this hypothesis hashtags spread faster between San Francisco and Mountain View, since they are closer to each other; but slower between San Francisco and Austin.
- **Community Affinity:** The second hypothesis is that the “world is flat” and information spreads based on virtual communities enabled by the prevalence of the Internet. In this hypothesis, distance is less important than are the strength of these virtual ties between locations; e.g., under this hypothesis San Francisco and Austin may be considered closer in terms of common interest (and hence, hashtags should flow more rapidly between the

CIKM’13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.
Copyright 2013 ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505515.2505579.

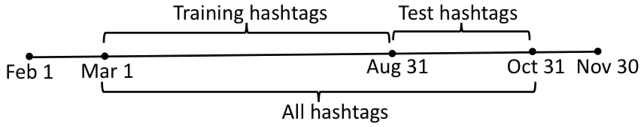


Figure 1: Hashtags datasets.

two), rather than Austin and its more proximate neighbor Houston.

We investigate a series of features inspired by these two hypotheses for predicting which hashtags will be popular in a specific location at a specific time. Since the best features may vary for each location, we additionally propose a reinforcement-learning based method whereby the best model is determined is location specific. In our experimental evaluation over 755 million geo-tagged Tweets, we find that reinforcement learning algorithm that selects the single best feature function for a location performs the best. This model is able to predict close to 70% of future hashtags occurrences accurately.

2. RELATED WORK

The area of information diffusion is well studied with most work focussed on study of diffusion through social and information networks, e.g., [11, 14, 15, 16, 18, 25]. But, our work in particular builds on two lines of research: Twitter analysis and geo-spatial analysis of social media.

Twitter Analysis: Most papers studying Twitter have focused on understudying its properties as a social network and had tried to analyze information diffusion as a effect of the underlying social network [12, 17, 18, 25]. Hence, similar models have been applied to study hashtag propagation on Twitter's social network [21, 6]. In related research, people have studied approaches to predict the popularity of hashtags in a given time frame in [24], sentiment detection on Twitter [8], topic tracking on twitter streams [19], and so forth.

Geo-spatial Analysis of Social Media: The emergence of location-based social networks like Foursquare, Gowalla, Google Latitude, and so on, has motivated several studies related to large-scale geo-spatial analysis like [22, 20, 2, 3, 9, 13]. In a recent paper [4] authors dealt with the spatial analysis of Youtube videos, in which they observed the highly local nature of videos based on the propagation patterns of Youtube videos. On Twitter people have studied geo-spatial analysis in the context of inferring geographic information from tweets like predicting user locations [5] and spatial modeling to geolocate objects [7].

3. TWITTER DATA COLLECTION

For our analysis we collected data using the Twitter Streaming API. We used the API between February 1 and November 30, 2011 to get a sample of around 755 million geo-tagged tweets which contains around 10 million unique hashtags. Every tweet is tagged with a latitude and longitude indicating the location of the user at the time of the posting and the tuple <`hashtag`, `time`, `latitude`, `longitude`> corresponds to a particular hashtag occurrence.

To support location-based analysis, we divide the globe into square grids of equal area using Universal Transverse Mercator (UTM), a geographic coordinate system which uses a 2-dimensional Cartesian coordinate system to map locations on the surface of the globe [1]. The issue with using an angular co-ordinate system like latitudes and longitudes is that distance covered by a degree of longitude differs as we move towards the pole. In addition, the distance covered by moving a degree in latitude and longitude is same only at the equator. Hence, it is hard to break globe into grids using this system. UTM on the other hand gives us a system of grids that closely matches distances in metric system making our analysis easier. While varying the choice of grid size can allow analysis at multiple levels (e.g., from state-sized cells to neighborhood-sized ones), we adopt a middle ground by dividing the globe into squares of 10km by 10km. Some grid cells will naturally be densely populated, others will be sparse. Let this set of distinct locations, each corresponding to a square, be represented by the set L.

To avoid sparsely represented hashtags, we consider only hashtags with at least 5 occurrences in a location and consider only hashtags with at least 250 total occurrences across all locations. Since some hashtags may have begun their Twitter life before the first day of our sample (February 1) while others may have continued on after the last day (November 30), we consider both February and November as buffer months. Hence, we capture the full lifecycle of hashtags starting on or after March 1 and ending by October 31, which focuses our study to hashtags which have both their birth and death within the time of study (and as a result, removes cyclical hashtags like "#ff" and "#nofollow"). As illustrated in Figure 1, we additionally divide the set of all hashtags into two sets: a training set based on hashtags from March to August; and a test set based on September to October. Hashtags that start in training but continue into test are ignored. In this way, the training set contains 1466 complete hashtag propagations and the test set contains 515.

4. SPATIO-TEMPORAL MEME PREDICTION

Let H be a set of hashtags and L the set of distinct locations. Then for a hashtag $h \in H$ let o_l^h be the number of occurrences of the hashtag that have been observed in a location $l \in L$, and let e_l^h be the number of occurrences of the hashtag that are expected in l. We now define the problem of selecting top$-k$ hashtags for a location as **hashtag subset selection problem**.

DEFINITION 4.1. ***(Hashtag Subset Selection Problem):*** *Given an integer k, hashtag subset selection problem for a location l is the task of determining set of top$-k$ hashtags $S_l \subseteq R$ such that the total number of expected hashtags for S_l is maximized, i.e.,*

$$S_l = \underset{\{S \subseteq R \ | \ |S|=k\}}{\arg\max} \sum_{h \in S} e_l^h \tag{1}$$

To understand the hashtag subset selection problem better, consider the example shown in Figure 2. It shows propagation of two hashtags (pink and blue) in Dallas and Austin at time t. The number of observed and unobserved occurrences for these hashtags at a time t is indicated by the area below

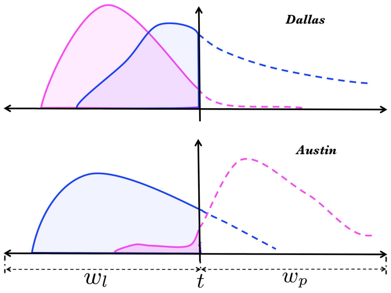


Figure 2: Example of trail propagation in two locations

shaded region with solid lines and a unshaded region with dotted lines respectively. Now, given that we only know the shaded regions under complete lines at t, the hashtag subset selection problem is the task of identifying k hashtags that will have maximum area under dotted lines. If $k = 1$, the solution to this problem would be $S_{Dallas} = \{\text{Blue}\}$ and $S_{Austin} = \{\text{Pink}\}$.

Feature Functions: If we know the area under dotted lines, i.e. e_l^h, then the solution to this problem is trivial. But, since we don't have that information at t, we have to develop methods to estimate this value. Let $\hat{e}_l^h$ be a score representing the value of e_l^h. Depending upon the method used to estimate this score, it could be anything - an integer predicting the number of expected occurrences or a value $\in [0, 1]$. The only condition is that a higher score for a location should indicate that this location sees more occurrences than a location with lower score. Then using (1), we redefine the hashtag subset S_l in terms of $\hat{e}_l^h$ as:

$$S_l = \underset{\{S \subseteq R \ | \ |S|=k\}}{\arg\max} \sum_{h \in S} \hat{e}_l^h \tag{2}$$

Like mentioned earlier, the score, $\hat{e}_l^h$, for a location l and hashtag h, can be determined using several techniques. Let F be the set of feature functions used to estimate the value of e_l^h, where, $f_i \in F$ is defined as $f_i : L \times H \rightarrow \mathbb{R}$. For example, a simple way to estimate expected hashtags in a location would be to use the notion that a hashtag that is popular in that location at current time will continue to be popular there during future. Like, lets say a hashtag (#redskins) about a football game in Washington D.C that is popular right now can be expected to remain popular next hour too. Concretely, calling this the greedy approach we can define the feature function corresponding to this, $f_{\text{greedy}} \in F$, as:

$$f_{\text{greedy}}(l, h) = o_l^h$$

where f_{greedy} just gives the number of occurrences of h that have been observed in l.

Learning Algorithms: Every feature function in F estimates a different value of $\hat{e}_l^h$, i.e., for a given location-hashtag pair we have $|F|$ estimates for $\hat{e}_l^h$. But, for a given location-hashtag pair, we can only use one value of $\hat{e}_l^h$ in (2). So, we formulate the task of determining a single value from a set of $|F|$ values as a learning problem. In particular, we propose a set of learning algorithms, $\mathcal{L}$, that use the set of feature functions F and a location-hashtag pair to estimate the value for $\hat{e}_l^h$. The learning algorithm can either combine all the estimated values in some ratio to get a new value of $\hat{e}_l^h$ or use some heuristic and select one of the values that it thinks is the best estimate. For example, we can estimate a new value for $\hat{e}_l^h$ using linear regression as:

$$\hat{e}_l^h = \epsilon + \sum_{f_i \in F} w_{f_i} f_i(l, h)$$

where, w_{f_i} are regression coefficients and ϵ is the error term.

In the following two sections we address two fundamental questions:

- **Feature Functions**: What feature functions F do we use to determine the value of $\hat{e}_l^h$?
- **Learning Algorithms**: What learning algorithms do we use to determine a single value from a set of $|F|$ values of $\hat{e}_l^h$?

5. FEATURE FUNCTIONS

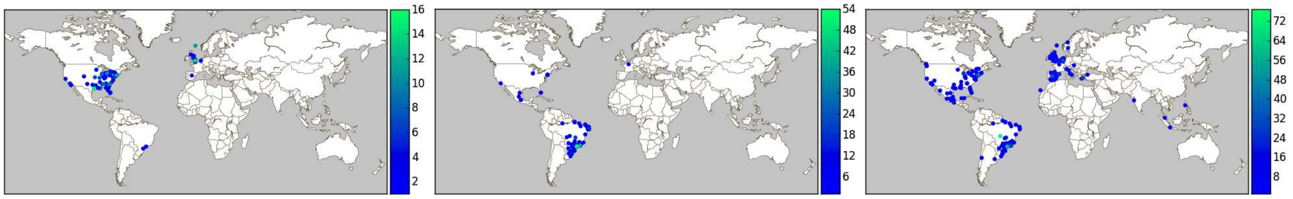
The feature functions to estimate the expected number of hashtag occurrences are guided by two major concepts of geo-spatial propagation: spatial affinity and community affinity. We first describe the feature functions based on spatial affinities where one function estimates local hashtags more accurately while other estimates global hashtags more accurately. We then describe feature functions that use community affinities to learn relationship between locations.

5.1 Spatial Affinities

In this section, we present feature functions that use spatial affinities between locations as described by the Tobler's hypothesis [23] to estimate expected number of hashtags. Tobler's hypothesis implies that the popularity of a hashtag in a location is dependent on the popularity of this hashtag in neighboring locations. So, we predict the future popularity of a hashtag in a particular location as a function of the hashtag's spatial distribution in other locations, such that the "contribution" made by the other location decreases exponentially as its distance from the particular location increases.

An advantage of using spatial affinities to estimate expected hashtag occurrences is that this approach allows us to develop different feature functions depending on our preferred hashtag type - local hashtag or global hashtag. Examples of local and global hashtags are shown in Figure 3. It shows spatial distribution for two local hashtags - #blackparentsquotes (USA and England) and #missuniverso (Brazil), and one global hashtag - #usopen (entire world), which were popular on the evening of September 19, 2011. We can imagine applications (like localized advertising) where we would want to prefer one type of hashtag over other and the feature function based on spatial affinities helps in such cases. In particular, we propose two feature functions: (i) global feature function which is suitable to estimate hashtags that are globally popular; and, (ii) local feature function which is suitable to estimate local hashtags.

Global Feature Function: This function uses spatial distribution of hashtags and estimates global hashtags more accurately than local. It is similar to the greedy feature in the sense that both these functions use hashtag's observed



(a) #blackparentsquotes (b) #missuniverso (c) #usopen

Figure 3: Distribution of three trails on the evening of September 19, 2011.

occurrences to estimate expected hashtag occurrences. But, unlike greedy, this approach doesn't use raw occurrence counts but *shifted occurrence counts*. Shifted occurrences are occurrences that are contributed to a location from other locations using Tobler's hypothesis, such that locations that are close by contribute greater number of occurrences to the location than locations that are far off. The global feature function is defined as:

$$f_{\text{global}}(l, h) = \sum_{l_i \in L} o^h_{l_i} \alpha^{-H(l_i,l)}$$

where, the sum calculates the total number of shifted footprints of h contributed by all locations to l. The exponential function helps model Tobler's hypothesis by decaying the contribution made by l_i to l depending on the distance between the two locations. The parameter α controls the rate of decay and in our experiments we set $\alpha = 1.01$.

Local Feature Function: As mentioned earlier, this feature function uses spatial distribution to estimate expected hashtag occurrences for local hashtags more accurately than global hashtags. But, instead of estimating expected count this feature function estimates a score in $[0, 1]$ that is an indicator of expected hashtag occurrences, such that, a higher score for a location indicates that more hashtags are expected at that location than a location with lower score. But, before describing this function, we first define the probability of observing a hashtag h in l, P^r_l, as:

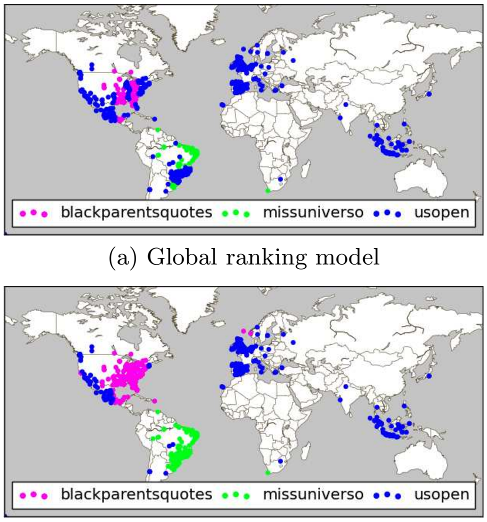
$$P^h_l = \frac{o^h_l}{\bigcup_{l_i \in L} o^h_{l_i}}$$

The score is calculated by applying Tobler's hypothesis to the hashtag observing probability. So, we define the local feature function for a hashtag h in a location l as:

$$f_{\text{local}}(l, h) = \frac{\sum_{l_i \in L} P^h_{l_i} \alpha^{-H(l_i,l)}}{\sum_{l_j \in L} \sum_{l_i \in L} P^h_{l_i} \alpha^{-H(l_i,l_j)}}$$

where, the numerator sums the shifted hashtag occurrence probability values from all locations to l. The exponential term is used to model Tobler's hypothesis such that locations that are closer to l contribute more to the score than locations that are far from l. Like before, in our experiments we set $\alpha = 1.01$.

To illustrate differences between the two spatial affinity based feature functions described in this section, consider the spatial distributions of three hashtags shown in Figure 3. We use the global and location feature selection methods to predict the expected number of occurrences for each of these hashtags. Then we mark every location with color of the hashtag that was most accurately estimated. The performance of these feature functions is shown in Figure 4. In these figures we observe the difference in approaches that the two feature functions take to estimate expected number of occurrences for local and global hashtags. The global feature function as expected estimates hashtag occurrences for global hashtags more accurately as shown by the blue locations in Brazil and USA where other local hashtags exist. The local feature function on the other hand, estimates the excepted occurrences of local hashtag pink #blackparentquotes (pink) and #missuniverso (green) hashtags more accurately.



(a) Global ranking model

(b) Local ranking model

Figure 4: Ranking trails using geospatial distribution

5.2 Community Affinities

The approaches proposed so far took into account only the geographical distances between two locations to estimate expected hashtag occurrences. In this section, we move beyond geographical distances and look at an alternative approach that considers the impact of virtual communities that exist over Internet. In particular, we present feature functions that use community affinities between locations that may not necessarily be close in terms of geographical distance. In particular, we propose two feature functions that differ in the way community affinities between two locations is measured: (i) common hashtags feature function uses community affinities measured based on the set of common hash-

Spatial Affinities			Community Affinities		
City	**Distance (miles)**	**Affinity**	**City**	**Distance (miles)**	**Affinity**
San Antonio	79	0.54	Los Angeles	1,373	0.96
Houston	167	0.79	Washington D.C	1,520	0.94
Dallas	191	0.92	New York	1,732	0.92

Table 1: Comparison between spatial and common hashtag affinities for Austin

tags shared between locations; and, (ii) hashtag transmission feature function uses community affinities between locations measured based on the hashtags that a location might have transmitted to another.

Both these approaches learn affinities between locations based on historical hashtag propagations. To do this we use the training set described in Section 3. Let H^T be the set of all hashtags observed in the training set and $H^T_l \in R$ the set of hashtags observed in location l. Then, we define a prior probability of observing a hashtag in l as:

$$P^T_l = \frac{|H^T_l|}{|H^T|}$$

We define $\mathcal{C}_{l_i \to l_j} \in [0, 1]$ as the measure of community affinity between locations l_i and l_j such that, $\mathcal{C}_{l_i \to l_j} = 1.0$ indicates that a hashtag in l_i will definitely occur in l_j and $\mathcal{C}_{l_i \to l_j} = 0.0$ indicates that a hashtag in l_i will not occur in l_j.

Common Hashtags Feature Function: In this approach we measure community affinities between locations based on the information about common hashtags observed between a pair of locations. The intuition behind this approach is that if locations are connected by virtual communities then they must share common hashtags. Ex: techies in San Francisco and techies in Austin though geographically apart will share common hashtags. For the pair of locations l_i and l_j we define the common hashtag affinity $\mathcal{C}^{com}_{l_i \to l_j}$ when a hashtag has occurred in l_i, as:

$$\mathcal{C}^{com}_{l_i \to l_j} = \frac{|H^T_{l_i} \cap H^T_{l_j}|}{|H^T_{l_i}|}$$

Note that there might be cases where $\mathcal{C}^{com}_{l_i \to l_j} \neq \mathcal{C}^{com}_{l_j \to l_i}$ as the number of hashtags observed in these locations might be different ($|R_{l_i}| \neq |R_{l_j}|$). We now define common hashtag feature function using affinities learned from common hashtags observed in locations as:

$$f_{\mathrm{com}}(l, h) = \sum_{l_i \in L - l} P^T_{l_i} \; P^h_{l_i} \; \mathcal{C}^{com}_{l_i \to l_j}$$

where, the sum calculates the total influence other locations have on l to make a hashtag h popular.

Hashtag Transmission Feature Function: After looking at affinities based on common hashtags observed in locations, we now look at affinities based on a set of hashtags that a location might have transmitted to another. In particular, with this approach we are interested in learning affinities that can reflect temporal relationships between locations. We define the affinity, $\mathcal{C}^{tran}_{l_i \to l_j}$, measured this way as hashtag transmission affinity and it indicates the chance that a hashtag observed in a particular location will be observed in another location in future. For example, in Figure 2, observing pink hashtag that is popular in Dallas during the estimation window become popular in Austin during the prediction window, we can learn the temporal relationship between these two locations. We define $\mathcal{C}^{tran}_{l_i \to l_j}$, as:

$$\mathcal{C}^{tran}_{l_i \to l_j} = \frac{|\{h \mid t^h_{l_j} > t^h_{l_i} \quad \forall h \in H^T_{l_i} \cap H^T_{l_j}\}|}{|H^T_{l_i}|}$$

where, t^h_l is the location l's traction time for h. The numerator in this definition is the size of set of hashtags that gained traction in l_i before l_j. Like in case of affinities based on common hashtags, there might be cases where $\mathcal{C}^{tran}_{l_i \to l_j} \neq \mathcal{C}^{tran}_{l_j \to l_i}$. Similar to common hashtag feature function, the hashtag transmission feature function is defined as:

$$f_{\mathrm{tran}}(l, h) = \sum_{l_i \in L - l} P^T_{l_i} \; P^h_{l_i} \; \mathcal{C}^{tran}_{l_i \to l_j}$$

An example of how community affinities differs from spatial affinities is shown in Table 1. In this table we compare the community (common hashtag) and spatial affinities for Austin, Texas. We observe that though Austin is spatially closer to some of the other big cities in Texas, the hashtags observed there are more similar to the hashtags observed in Los Angeles, Washington D.C and New York.

6. LEARNING FEATURE FUNCTIONS

In the previous sections we proposed five feature functions to estimate $\hat{e}^h_l$ - first, the greedy feature function, next, two feature functions that used the hypothesis that distance between two locations played an important role in making hashtags popular, and finally two feature functions that used a contradictory hypothesis that is wasn't distance but virtual communities on Internet that impact hashtag popularities.

The next task is to reduce $|F|$ values of $\hat{e}^h_l$ to a single value that can be used in (2). A simple approach now would be to evaluate which of these feature functions determines the value of $\hat{e}^h_l$ most accurately and select it. In reality though, we might observe that a single feature function might not be suitable for all locations. Instead the demography of a place might dictate selection of a particular function that is best for this place. For example, metropolitan areas like those around Austin might prefer community feature functions, while smaller towns surrounding Dallas might prefer the spatial feature functions. In addition, it is possible that some locations might not prefer one feature function over the other but a combination of these feature functions in some ratio. Hence, in this section to deal with these issues we concentrate on two things: (i) introduce evaluation metrics to measure the performance of feature functions for a location; and, (ii) describe algorithms that use these metrics to learn the best feature function or the best ratio for combining the feature functions for a location.

6.1 Evaluation Metrics

We now describe two evaluation metrics which we use in learning the best feature function or best combination of feature function for a given location. The value for each of these metrics is in the range $[0, 1]$ with 0.0 indicating the worst performance and 1.0 indicating the best performance. Given a location l, we denote the best set of top$-k$ hashtags at this location as $S_l{}^\star$ and the set of top$-k$ hashtags selected by our ranking models as S_l (without the $\star$ on top). The two evaluation metrics are:

Accuracy: This metric measures the similarity between $S_l{}^\star$ and S_l. This measure is similar to other set comparsion metrics like the Jaccard index. It is defined as:

$$\mathcal{A}_l = \frac{S_l{}^\star \cap S_l}{k}$$

such that, if the sets are identical accuracy is 1.0 and 0.0 if they are disjoint.

Impact: While accuracy measures the similarity between the sets, it doesn't measure the effect of selecting a particular set of hashtags over another. For example, it is possible that two disjoint sets of hashtags might observe same number of total hashtag occurrences after they are selected, resulting in the same performance. Hence, we define a metric called hashtag subset's impact defined as:

$$\mathcal{I}_l = \frac{\sum_{h \in S_l} e_l^h}{\sum_{h \in S_l{}^\star} e_l^h}$$

which measures the ratio between the number of hashtag occurrences that were observed for hashtags in S_l to those in $S_l{}^\star$. The impact value 0.0 signifies no impact while 1.0 signifies best impact.

6.2 Learning Algorithms

We next describe learning algorithms to determine a single value for $\hat{e}_l^h$ from $|F|$ values for it estimated using the feature functions. We build a different model for each location $l \in L$ and to build these models we use the training and test hashtag sets described in Section 3, which contains complete propagations for all hashtags. In particular, we present two learning algorithms depending on how the learning algorithm assigns best feature function to a location: (i) linear regression algorithm which determines the weights for a linear combination of feature functions for a location; and, (ii) reinforcement learning algorithm which determines the single best feature function for a location.

Learning with Regression: We first describe a learning algorithm using linear regression to determine a single value for $\hat{e}_l^h$, where a different model is built for each location $l \in L$. To build these models we use the training and test hashtag sets described in Section 3. We know the complete propagation for a hashtag in the training and test sets. Consider the matrices X_l and Y_l:

$$X_l = \begin{pmatrix} 1 & f_1(l, h_1) & f_2(l, h_2) & \cdots & f_{|F|}(l, h_{|H|}) \\ 1 & f_1(l, h_1) & f_2(l, h_2) & \cdots & f_{|F|}(l, h_{|H|}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & f_1(l, h_1) & f_2(l, h_2) & \cdots & f_{|F|}(l, h_{|H|}) \end{pmatrix}$$

$$Y_l = \begin{pmatrix} e_l^{h_1} & e_l^{h_2} & \cdots & e_l^{h_{|H|}} \end{pmatrix}^T$$

where, X_l matrix has $|H|$ rows, one for each hashtag in the training set. Every row contains $1 + |F|$ values each, except that for the first column, corresponding to the expected value for the hashtag calculated using the feature function corresponding to the column. The column matrix Y_l has $|H|$ rows with each value equal to the real expected value determined from the training set.

The values for the matrices is calculated using learning (w_l) and prediction (w_p) windows as shown in Figure 2. Note that, the expected value in Y_l increases as we increase the prediction window, i.e. using a prediction window of 4 hours will have more hashtag occurrences than a window of 2 hours. Similarly, the observed hashtag occurrences used by feature functions to determine X_l varies as the learning window is varied. The impact of varying these windows on the learning algorithms is evaluated later in the experiment section. Using these matrices, we define Y_l as a linear function of X_l,

$$Y_l = X_l \beta_l + \mathcal{E}_l \tag{3}$$

where, β_l is a column matrix called parameters matrix and $\mathcal{E}$ is the matrix of error terms. The parameters matrix contains the weights using which the various feature functions should be combined to determine $\hat{e}_l^h$ from $|F|$ estimates for it. The parameters matrix can be estimated by linear regression using the equation (3). We for a new hashtag h we can determine the expected occurrences for it using:

$$\hat{e}_l^h = \hat{\beta}_0 + \sum_{i=1}^{|F|} \hat{\beta}_i f_i(l, h)$$

Learning with Reinforcement: In the previous method we used linear regression to combine the values of expected hashtag occurrences estimated by all the feature functions. We now describe an approach that uses reinforcement learning to determine this value. By reinforcement learning we mean that during every time interval the learning algorithm makes some prediction, then in the next time interval it updates its model based on its performance before making future predictions.

The learning algorithm is run independently for every location at regular time intervals. Let the weight $W_l^f(t)$, for every location-feature function pair, represent the value that the learning algorithm uses to select a feature function for a given location at time t. During every time interval we select a feature function that we expect will perform best using $W_l^f(t)$. We then evaluate the performance of all of all the feature functions using some metric (accuracy or impact) and update the $W_l^f(t)$ accordingly. So, the idea is that after a few observations the algorithm learns which feature function is best suited for a location.

We describe two methods of reinforcement learning depending upon how $W_l^f(t)$ is updated and used to select a feature function: (i) Deterministic method which selects the best feature function at any time; (ii) Randomized method which uses a probability to select a feature function.

Deterministic Method: This method selects the single best feature function for a location. In this method the weight $W_l^f(t)$ for every location-feature function represents the cumulative loss for the function until time t:

$$W_l^f(t) = W_l^f(t-1) + (1 - \mathcal{A}_l^f)$$

then, for the next interval we select the feature function with the lowest cumulative loss until now, i.e. $f = \arg\min_{f \in F} W_l^f(t)$.

Randomized Method: Instead of picking a feature function using cumulative loss as in the previous approach, in this method we select a feature function using a probabilistic approach. Let $\mathcal{P}_l^f(t)$, such that $\sum_{f \in F} \mathcal{P}_l^f(t) = 1$, be the probability of choosing a feature function from F for location l at time t. We initialize these probabilities to $\frac{1}{|F|}$. The weight $W_l^f(t)$ for every location-feature function is then used to determine probabilities for the next iteration. Like before, this weight is updated during every iteration as:

$$W_l^f(t) = W_l^f(t-1) \centerdot \beta^{(1-\mathcal{A}_l^f)}$$

where, $\beta \in [0, 1]$. By using this function of β, as the accuracy for a feature function decreases the weight corresponding to that function decreases. The probability for choosing a feature function is then updated as:

$$\mathcal{P}_l^f(t+1) = \frac{W_l^f(t)}{\sum_{f \in F} W_l^f(t)}$$

7. EXPERIMENTS

We now evaluate the feature functions along with the learning algorithms described in this paper. In the first set of experiments we analyze performance of feature functions using accuracy and impact, and then the analyze the effect of varying various learning parameters. We then evaluate some characteristics of the learning algorithms. For the experiments we use the dataset described in Section 3.

7.1 Performance of Learning Algorithms

In this section, we evaluate the performance of the feature functions and the learning algorithms using the metrics - accuracy and impact - described earlier in the paper. We start by evaluating the performance of the the feature functions and the learning algorithms on fixed parameters and then evaluate the performance of the learning algorithms by varying parameters like number of top hashtags (k), the length of learning window and the length of prediction window.

An example of how the methods are evaluated, using evaluation metrics, is shown in Table 2. In this example, we predicted the subset of hashtags for New York on September 20, 2011. We predicted these hashtags at 20:00 UTC for the next 2 hours using a learning window of 6 hours. The first 3 columns show the prediction made by the 3 feature functions - greedy, local spatial affinity and community affinity based on hashtag transmission. The last column shows the best set of hashtags or the gold set. The hashtags in bold indicate that they were one of the correct hashtags predicted. In this example, we observe one of the drawbacks of greedy approach - it's inability to predict hashtags which it hasn't observed yet locally (in NY). The feature function using local spatial affinity does slightly better, in the sense it predicts mostly local hashtags, but misses out on hashtags that are popular globally like dudesthatsayno***, terriblenamesfor*** and so on. On the other hand, the feature function using community affinity based on hashtag transmission predicts 4 of the 5 hashtags correctly and performs the best. We also see that the performance of the feature function measured using the evaluation metrics we defined gives an indication of their actual performance.

Prediction Method	**Accuracy**	**Impact**
Greedy	0.55	0.55
Global	0.64	0.64
Local	0.60	0.60
Common Hashtags	0.62	0.62
Hashtag Trans.	0.63	0.63
Linear Regression	0.32	0.32
Deterministic Method	**0.68**	**0.69**
Randomized Method	0.68	0.67

Table 3: Performance of various feature functions and learning algorithms

We then evaluated the performance of all the feature functions and learning algorithms as shown in the example. We evaluated the methods using $w_p = 2$ hours, $w_l = 6$ hours and $k = 10$. The performance of the methods is shown in Table 2. Among the feature functions, we observe that the function that uses global spatial affinities performs the best. It has an accuracy and impact of 64%, implying that this method on average predicts 64% of hashtag occurrences for 2 hours in future correctly. In addition, as expected, the learning algorithms perform better than the individual feature functions with the method that uses reinforcement learning performing the best. The performance of this method could be attributed to the fact that it learns the best feature function for a location and uses that during predictions.

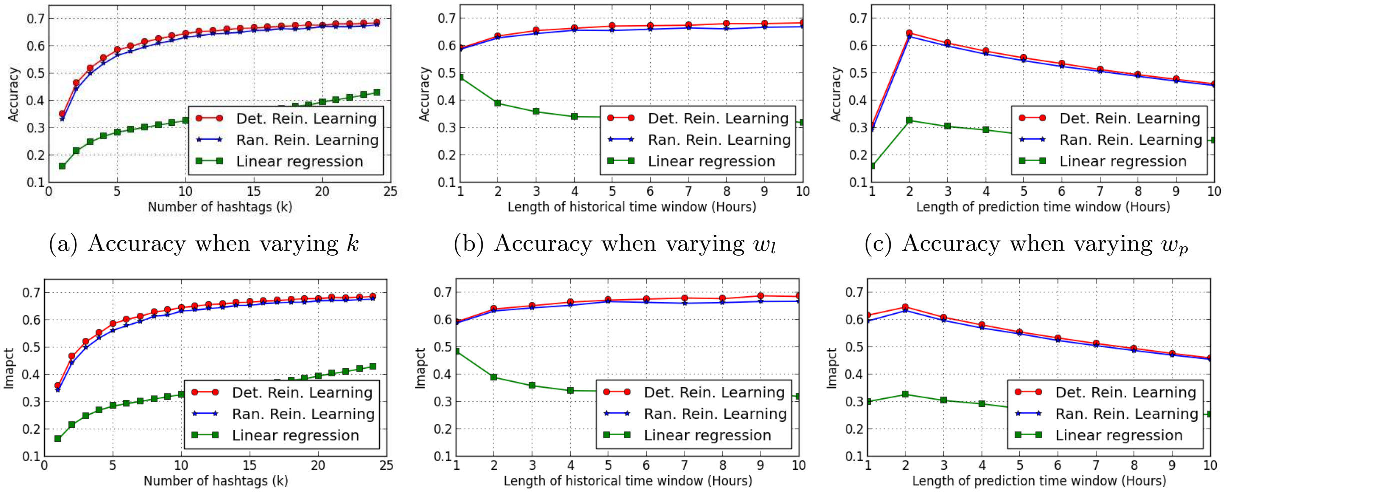
Performance With Varying k: For this experiment, we evaluated the performance on various learning algorithms by varying the number of top hashtags (k). We set the learning window length $w_l = 6$ hours, prediction window length $w_p = 2$ hour and then varied the value for k. The results of this experiment evaluated using accuracy and impact are shown in Figure 5(a) and Figure 5(d) respectively. The figures show the performance of the ranking algorithms as we vary the value of k from 1 to 25.

As described before, accuracy measures the similarity between the set of hashtags selected by our algorithms and the best set of hashtags for that interval, while impact measures how close we are to the best possible algorithm when it comes to the number of observed hashtag occurrences. We know that the distribution of hashtags at a location follows a zipfian distribution with few trails accounting for most occurrences. Hence, the problem of selecting top$-k$ hashtags becomes harder when k is small. The result of this distribution is reflected in the performance of our ranking algorithms as well, where we observe that the performance of you al-

Greedy	**Local (Spatial)**	**Hashtag Trans. (Community)**	**Actual Hashtags** (% of hashtag occ.)
cgi2011	teamenzomusic **takewallstreet** **cgi2011** miscellaney epatcon	**cgi2011** **dudesthatsayno***** **terriblenamesfor***** foino20desetembro **takewallstreet**	faze3 (0.29) terriblenamesfor*** (0.29) cgi2011(0.14) dudesthatsayno*** (0.14) takewallstreet (0.14)

Accuracy = 0.20
Impact = 0.10
	Accuracy = 0.40
	Impact = 0.29
		Accuracy = 0.80
		Impact = 0.71	

Table 2: Top hashtags identified using different feature functions for New York on September 20, 2011 at 20:00.



(a) Accuracy when varying k (b) Accuracy when varying w_l (c) Accuracy when varying w_p

(d) Impact when varying k (e) Impact when varying w_l (f) Impact when varying w_p

Figure 5: Ranking Model Performance

gorithm improves as the value of k increases. The zipfian distribution also explains the flattening of the curve after around $k = 10$. The hashtags selected by the algorithms after this value of k don't result in significant increase in impact as the observed occurrences of these hashtags is small, resulting in the flattening of the curve. Of the the learning algorithms, the algorithms that used reinforcement learning performed better than the algorithm that used linear regression to estimate the value of expected hashtag occurrences.

Performance With Varying w_l: To evaluate the performance of our ranking algorithms for varying lengths of learning time window, we set the prediction time window $w_p = 2$ hours and $k = 10$. We varied w_l from 1 hour to 10 hours in 1 hour intervals. The results from this experiment using accuracy is shown in Figure 5(b) and using impact is shown in Figure 5(e).

We observe that the performance of the learning algorithms that use reinforcement is better than the algorithm that uses linear regression. But, there is no significant difference between the two methods that use reinforcement learning. Initially, as the length of learning window increases we see in improvement in accuracy (and impact) for all the algorithms. But accuracy beings to level out as the length of estimation window continues to increase. We believe the performance of the algorithms improves during initial increase in learning window because with a longer window they are able to analyze larger number of hashtag occurrences which helps them make better decisions during prediction. But, as the window continues to increase they observe older hashtags propagations, which results in evening out or even decreasing performance. The window that is best suitable for estimation might depend on the network on which the social network are propagating and the nature of hashtag themselves. In case of hashtag propagation on Twitter we found a window of 6 hours was best suited for hashtag prediction.

Performance With Varying w_p: We next evaluated the performance of our learning algorithms for varying lengths of prediction time window. We set the learning time window $w_l = 6$ hours and $k = 10$. We then varied w_p from 1 hour to 10 hours in 1 hour intervals. The results from this experiment using accuracy is shown in Figure 5(c) and using impact is shown in Figure 5(f).

Like in earlier experiments we observe that learning algorithms that use reinforcement perform better than the linear regression algorithm. In particular, we observe that the performance of the algorithms peaks when the prediction window is 2 hours and then decreases with the increase in length of prediction window. This result shows the sensitivity of the prediction window, because unlike w_l which had

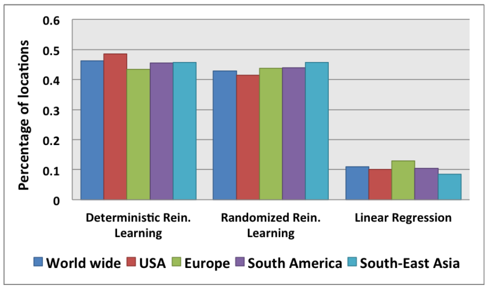
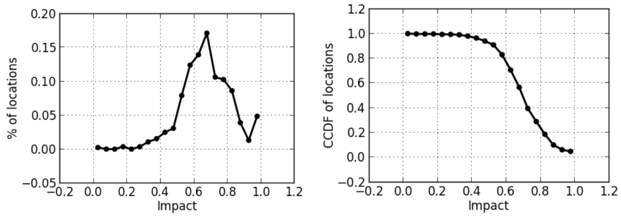


Figure 6: Distribution of preferred learning algorithm in various locations by geographical areas (Impact).



(a) Distribution of impact scores (b) CCDF of impact scores

Figure 7: Analysis of impact scores for various locations. Using our learning algorithms we were able to achieve a impact of at least 60% for more than 80% of the locations.

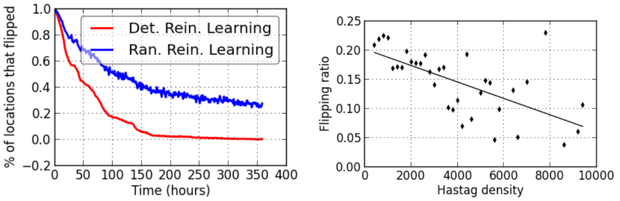
a region in which the performance didn't change, in case of w_p the performance of the model decreases almost linearly with time.

7.2 Learning Analysis

In this section we analyze learning algorithms in detail. We first analyze the impact scores obtained using these algorithms and then analyze the rate at which the learning algorithms assign feature function to locations. We then analyze these algorithms further by defining a metric called flipping ratio which measures the uncertainty of a learning algorithm in assigning feature functions.

Analysis of Impact Scores: We next analyze the impact scores for all the locations in our dataset. For this analysis we use impact scores obtained using the three algorithms that were compared in the previous section. Every location is assigned the best algorithm specific to it. We divided all the locations into 4 regions - United States (0.33%), Europe (0.34%), South America (0.25%) and South-East Asia (0.08%). The number in bracket indicates the percentage of locations in the region. The distribution of the algorithms is shown in Figure 6. In spite of varying number of locations in each region we observe that distribution of learning models is similar. All the regions have almost equal number of locations that prefer either deterministic or randomized algorithm and a small number of locations prefer linear regression.

The distribution of impact scores and its complementary cumulative distribution function is shown in Figure 7(a) and Figure 7(b) respectively. As described earlier impact in a way measures how close the learning algorithm selected for a location is close to the ideal algorithm that can be designed for that location. So, a impact of 1.0 signifies the algorithm as good as the best algorithm. We observed that more than half of the locations, for which we made predictions, we were able to achieve an impact of at least 0.70.



(a) Learning rate comparison (b) Flipping ratio Vs Location Density

Figure 8: Deterministic algorithm learns faster than the randomized version and the hashtag density of a location impacts the rate at which it learns.

Analysis With Learning Rate: In this experiment we compare the rate at which the two reinforcement algorithms, we described in Section 6.2, learn feature function to be assigned to a location. The result of this experiment is shown in Figure 8(a). In this figure, the learning time is shown in x-axis and the percentage of locations that flipped their decision in the current interval is shown in y-axis.

We observe that the deterministic algorithm is faster than the randomized algorithm. The flipping nature of these algorithms could be attributed to the way in which they select feature functions. The randomized algorithm selects a feature function based upon probabilities estimated from the feature function weights while the deterministic algorithm is much simpler in the sense it makes a decision based upon the cumulative loss. These probabilities are non-zero for more than one feature function resulting in the algorithm flipping more. This issue is not observed in case of the deterministic algorithms making it much more stable and hence faster. In spite of the simple nature of deterministic algorithm we observe that its performance as better than that of the randomized algorithm. For hashtag propagation in Twitter we saw that we were able to assign feature function to locations using about a weeks data (flatting of red curve in Figure 8(a)).

Analysis With Flipping ratio: We first describe flipping ratio and then analyze the learning algorithms using it. In our experiments test set is broken into time intervals of equal size. The learning algorithms select a feature function every interval. Then, flipping ratio measures the uncertainty of a learning algorithm by determining the number of times the algorithm changes its decision from that made in previous interval. It is defined as:

$$\text{Flipping Ratio} = \frac{\#\text{ of decision changes}}{\#\text{ of intervals in test set}}$$

where, an ideal learning algorithm with flipping ratio 0.0 will pick a feature function for a location in its first attempt, while the worst learning model with flipping ratio 1.0 will change its decision every interval.

We analyzed the correlation between the density of location and its flipping ratio. Since, we can't get the exact density for every location, we assume hashtag occurrences at a location as an indicator of the actual density. One of the issue with this assumption is that hashtag occurrence counts might not be a good indicator of actual density. For example, there could be dense locations with poor Internet connectivity resulting in low occurrences, while college towns with low density might have large number of occurrences. But, this assumption doesn't impact applications using hashtag subset selection, because the hashtags selected by our models are still reflective of the user activity online and not the actual density. The correlation between density of a location and its flipping ratio is shown in Figure 8(b). We see that flipping ratio decreases with increase in density of a place. In other words, the ability of a learning algorithm to assign feature function to a location increases as the number of hashtag occurrences at that location increases. This is an important result because, the earlier and more accurately we can assign feature function to a location with high density the better performance of our algorithm is.

8. CONCLUSION

In this paper, we proposed and evaluated approaches that predict where and when a online meme will be popular. In particular, we developed models based on the two competing hypotheses of information spread over geo-spatial networks - spatial affinity and community affinity. We then evaluated these models over a collection 755 million geo-tagged Tweets and found a model that can predict future hashtags occurrences with a 70% accuracy. In our future work, we are interested in analyzing how these approaches scale under large amount of data arriving at rapid rate.

9. ACKNOWLEDGMENTS

This work was supported in part by NSF grant IIS-1149383. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsors.

10. REFERENCES

[1] Universal transverse mercator coordinate system, November 2012.

[2] L. Backstrom, J. Kleinberg, R. Kumar, and J. Novak. Spatial variation in search engine queries. WWW '08.

[3] L. Backstrom, E. Sun, and C. Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. WWW '10.

[4] A. Brodersen, S. Scellato, and M. Wattenhofer. Youtube around the world: geographic popularity of videos. WWW '12.

[5] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: a content-based approach to geo-locating twitter users. CIKM '10.

[6] E. Cunha, G. Magno, G. Comarela, V. Almeida, M. A. Gonçalves, and F. Benevenuto. Analyzing the dynamic evolution of hashtags on twitter: a language-based approach. LSM '11.

[7] N. Dalvi, R. Kumar, and B. Pang. Object matching in tweets with spatial models. WSDM '12.

[8] D. Davidov, O. Tsur, and A. Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. COLING '10.

[9] J. Ding, L. Gravano, and N. Shivakumar. Computing geographical scopes of web resources. VLDB '00.

[10] Foursquare. About foursquare, April 2012.

[11] M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10.

[12] B. A. Huberman, D. M. Romero, and F. Wu. Social Networks that Matter: Twitter Under the Microscope. *Social Science Research Network Working Paper Series*, Dec. 2008.

[13] 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, New York, NY, USA, 2012. ACM.

[14] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. KDD '03.

[15] D. Kempe, J. Kleinberg, and É. Tardos. Influential nodes in a diffusion model for social networks. In *IN ICALP*, 2005.

[16] G. Kossinets, J. Kleinberg, and D. Watts. The structure of information pathways in a social communication network. KDD '08.

[17] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? WWW '10.

[18] K. Lerman and R. Ghosh. Information contagion: an empirical study of the spread of news on digg and twitter social networks. *CoRR*, 2010.

[19] J. Lin, R. Snow, and W. Morgan. Smoothing techniques for adaptive online language models: topic tracking in tweet streams. KDD '11.

[20] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. An empirical study of geographic user activity patterns in foursquare. ICWSM' 11.

[21] 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. WWW '11.

[22] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo. Socio-spatial properties of online location-based social networks. ICWSM' 11.

[23] W. Tobler. A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46(2), 1970.

[24] O. Tsur and A. Rappoport. What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities. WSDM '12.

[25] J. Yang and J. Leskovec. Modeling information diffusion in implicit networks. ICDM' 2010.