GeoWatch: Online detection of Geo-Correlated Information Trends In Social Networks

Ceren Budak Theodore Georgiou Divyakant Agrawal Amr El Abbadi
Department of Computer Science UCSB
Santa Barbara, CA 93106-5110, USA
{cbudak, teogeorgiou, agrawal, amr}@cs.ucsb.edu

ABSTRACT

Detecting information trends in online social networks is an important problem that has attracted the attention of both the industry and the research community in recent years. Global trends, information items that are trendy in the entire social network, can be detected using existing data streams techniques. However, detecting global trends is only the first step in understanding online social networks. As The First Law of Geography states "Everything is related to everything else, but near things are more related than distant things". This spatial significance has implications in various applications, trend detection being one of them. To this end, in this paper we propose a new algorithmic tool, *GeoWatch*, to detect geo-trends. *GeoWatch* is a data streams solution that detects correlations between topics and locations in a sliding window in addition to providing tools for analyzing topics and locations independently. The degree of correlation as well as the sliding window size can be set to arbitrary values thus enabling a flexible framework. *GeoWatch* has theoretical guarantees for detecting all trending correlated pairs while requiring only sublinear space and running time. Experiments on Twitter show that in addition to providing perfect recall, *GeoWatch* has near-perfect precision. As the Twitter analysis demonstrates, *GeoWatch* successfully filters out topics without geo-intent and detects various local interests such as emergency events, local political demonstrations or cultural events.

1. INTRODUCTION

Geography plays an important role in various aspects of our lives. As the first law of geography states "Everything is related to everything else, but near things are more related than distant things" [34]. Even though, with the advent of the Web and later online social networks, the "virtual" distance between the Web users have dramatically decreased, research shows that geographical locality still matters. Ugander et al. [37] study the social graph of active Facebook users and show that not only are friendships predominantly across users within the same country, but friendships between countries are also highly modular, and apparently influenced by geography. This locality in friend relation formation is also seen in use of language and sentiment [30] as well as topical interests [16].

In addition to capturing interests and intentions of users in different localities, geographical signals can also be used to extract relevant information from the public in crisis management [24]. Therefore, it is a critical task to develop social network analysis tools that have a geographical focus. Most research in this area is restricted to offline measurements to geographically characterize social networks [7, 30]. Recently, there has been more effort in online analysis of geo-trends in social networks [24, 32]. However, these works focus on defining frameworks in which data is simply geographically categorized while the task of discovering geo-intent by considering the correlation between locations and topics is not addressed. Given the large scale of data shared in online social networks, there is need for algorithmic solutions that capture geo-intent and detect informational trends in a scalable fashion. Our goal in this paper is to provide such an algorithmic tool that uses sublinear space and running time with approximation guarantees.

Trends in social networks are of high significance and a major point of interest in both the industry [18, 19] and the research community [23, 4, 32]. In this paper, our goal is to detect trends of true geographical nature rather than simply identifying frequent elements in various locations. A topic of global importance incidentally also has a high frequency of occurrence in different localities. Distinguishing such a topic from one that is trending in only certain localities is not possible without considering the *correlations* between places and topics. Therefore, in this work we focus on the problem of identifying the correlation of information items with different geographical places. Items that are trendy in general not for a specific location carry no geographical significance and therefore are irrelevant from the perspective of our study.

We propose *GeoWatch*; an algorithmic tool for detecting geotrends in online social networks by reporting *trending* and *correlated* location-topic pairs. *GeoWatch* also captures the temporality of trends by detecting geo-trends along a sliding window. To the best of our knowledge, this is the first work that detects spatial information trends in social networks by capturing correlations in a multi dimensional data stream. In addition, with the use of different window sizes, trends of different time granularity can be detected. *GeoWatch* has provable accuracy guarantees even though it requires sublinear memory and amortized running time. Such a scalable algorithmic tool can be used in real large-scale social networks to reliably detect local interests or even crisis events in a timely manner. Our analysis on Twitter data set shows that such geo-trend detection can be very important in detecting significant events ranging from emergency situations such as earthquakes to locally popular flash crowd events such as political demonstrations or simply local events such as concerts or sports events. The fast detection of

emergency events such as the March 11 Japan earthquake indicate the possible value of *GeoWatch* in crisis management.

To the best of our knowledge, this is the first work that detects spatial information trends in social networks by capturing correlations in a multi-dimensional data stream. In Section 2, we start by summarizing related work. Section 3 provides analysis on the Twitter data set used in this study. Next in Section 4 we introduce the characteristics that an ideal geo-trend detection tool should have and show that an exact solution is not scalable. Therefore, we propose an approximate solution, called *GeoWatch*, and provide proofs of accuracy and sublinear memory and running time requirements. This proposed framework is experimentally evaluated in Section 5. Finally Section 6 concludes the paper.

2. RELATED WORK

This study is in the intersection of social networks research, data streams research and geo-analysis. Here we will provide an overview of recent studies related to these topics:

Social Networks Analysis: In recent years, there has been a number of studies that focused on information trends from various perspectives [2, 15, 21, 23, 4]. Kwak et al. [21] study and compare trending topics in Twitter reported by Twitter with those in other media, showing that the majority of topics are headline or persistent news. In [23] Leskovec et al. study temporal properties of information by tracking "memes" across the blogosphere. Teitler et al. [32] collect, analyze, and display news stories shared in Twitter on a map interface, capturing geographical characteristics of social networks data. However, unlike our work they focus on identifying tweet clusters based on locations and not trend detection. Hong et al. [16] focus on user profiling from a geographical perspective by modeling topical interests through geo-tagged messages in Twitter. This problem is orthogonal to the problem studied in our paper as it focuses on user-centric modeling in an offline manner while *GeoWatch* detects trends in an online fashion. MacEachren et al. [24] aim to identify significant events in different localities for crisis management. This work provides a high level framework while we provide an efficient algorithmic tool with accuracy guarantees. Another framework in detecting spatiotemporal topics have been introduced in the context of online blogs [10].

Geographical information is important for recommendation systems in the context of social networks as well. This motivation has driven a considerable effort in the research community. A recent tutorial on location-based social networks (LBSN) discuss various research problems in this context [40]. One such problem [39] relates to identifying interesting activities to perform in a location. In that sense it is similar to finding interesting topics being discussed in the network. However, this technique is based on collaborative filtering and collective matrix factorization methods and therefore is not an online solution. Such a technique cannot respond rapidly to fast changing information trends.

Since our goal is to provide and evaluate a tool for detecting geo-trends in social networks, it is an important sub-task to geo-tag the social content in an accurate manner. Geo-tagging has been successfully addressed through NLP [13] and LDA [1] in the context of the unstructured web. However, this task introduces new challenges in the context of social networks. Various studies have focused on geo-tagging social networks data [7, 38, 12, 3]

Data Streams: There is a large family of data streams problems that are related to (but not the same as) our problem definition. One significant problem studied in the context of data streams is the *frequent elements problem* [8]. The algorithms for answering frequent elements queries are broadly divided into two categories: *sketch-based* and *counter-based*. In the *sketch-based* techniques [6, 9, 20], the entire data stream is represented as a summary *sketch* which is updated as the elements are processed. The *counter-based* techniques [28, 25, 11] monitor a subset of the stream elements and maintain an approximate frequency count. Although our method relies on frequent element detection, it also requires identifying correlations in a data set in an online manner.

There has been some effort in detecting correlations in multi dimensional data streams. In particular, in [29] the authors address the problem of fraud detection in Internet advertising networks. The proposed solution models discovering single-publisher attacks as a new problem of finding correlations in multidimensional data consisting of two dimensions; the publisher, and the IP address of a machine. In order to detect fraudulent behavior, they aim to detect *correlated pairs* where a correlated pair is defined as one where the IP is a *frequent (or heavy-hitter)* element for the publisher, and the publisher is a *frequent* element for the IP. Since this technique is a count-based solution, it only allows insertions and not deletions. Therefore, unlike our work, it uncovers correlated items in the *entire* data stream. For detecting trends in social networks, capturing temporal aspects is crucial. Therefore, the solution introduced in [29] is not applicable as is. Moreover, this work makes the assumption that the traffic characteristics of non-fraudulent publishers and IPs are stable within the analyzed window. Such an assumption is not applicable in online social networks where information trends are highly temporal. Similar to [29], *GeoWatch* keeps track of correlations in a multidimensional data stream but unlike SLEUTH [29], *GeoWatch* is a sketch based solution that allows for a sliding window implementation.

In another relevant work, Lappas et al. [22] study the notion of burstiness in documents in a spatiotemporal [27, 35] manner. While their methodology also captures the notion of geography and time, it focuses on data burstiness and not geo-intent. The streaming version of the problem does not provide guarantees of optimality (or sub-optimality) for the maximal window approach. In addition, geography in that context is defined based on a bounding box and not an actual location (such as city, country) which we believe is the natural aggregation-level in defining geographical interests.

3. GEOGRAPHICAL TWITTER ANALYSIS

For our experiments we used Twitter updates from February 1st to June 18th 2011. The data is extracted through Twitter's public API (GardenHose) and therefore constitutes $\sim 10\%$ of the overall Twitter updates of that time period. The average number of tweets per day is 14.2M (with a total of 2 billion for the whole period). After geo-tagging, a procedure described below, we obtained a total of 378,941,219 labeled datapoints, out of which 63M also include a hashtag. The number of users in our data set is 46M. The geographical data was obtained from [26], which contains complete hierarchical information and coordinates for approximately 50,000 cities from all the countries and regions of the globe.

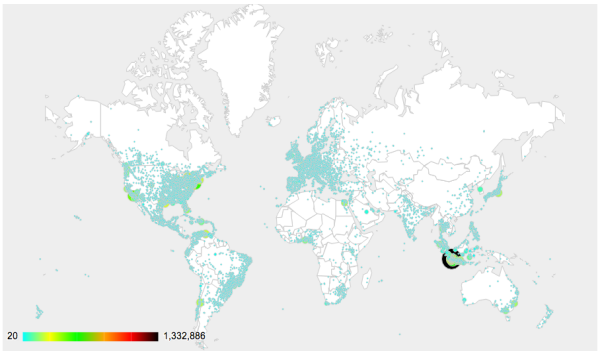
Geo-tagging Twitter Content: There are two types of geographical information that can be associated with a given tweet: the location the tweet is shared from (*geo-origin*) and the location that the tweet is about (*geo-focus*). While our technique can be applied for both cases, we will focus on detecting geo-trends when loca-

tion is based on *geo-origin* in the rest of this paper. To identify the *geo-origin* of a tweet, we utilize both the tweet and the user location. Tweet location is provided explicitly by the Twitter API in the form of a latitude and longitude pair. In certain cases the city name might also be available. Even though this signal offers a highly accurate estimation of the tweet location, it is sparse. Only 1.5% of tweets' *geo-origin* are identified through this method. The second signal, user location, is a user provided free-form text that carries more noise [38]. We extract this information by parsing the location string and identifying pairs of (longitude, latitude), (city-name, region-abbreviation), (city-name, region-name), (city-name, country-abbreviation) and (city-name, country-name). Since there are cases where a region or a country might have the same city name for more than one locations we choose as the best match the one with the largest population. After obtaining the location of a user, all his/her untagged tweets are tagged with this location, which increases the number of tweets that are tagged according to their *geo-origin* in our data set to 13%.

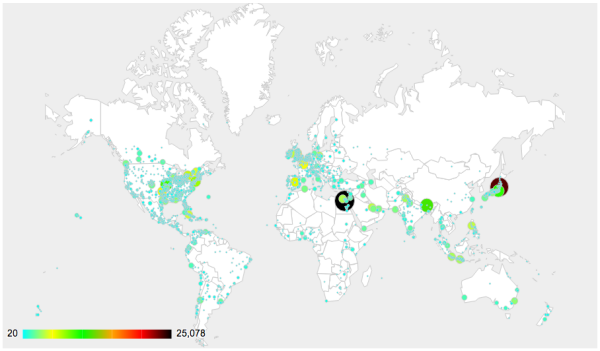
Geo-tagging social networks data is an active area of research. For instance, Cheng et al. [7] study the problem of identifying city-level location of Twitter users based on a probabilistic framework that relies purely on tweet content in identifying user locations. However, this solution requires a large number of tweets per person for high accuracy and therefore identifying the location of a large fraction of the population is not possible. In addition, this solution is a batch process while our goal is to detect trends in a streaming fashion necessitating even geo-tagging to be performed in an ad-hoc manner. Other Twitter geo-tagging studies [38, 12] suffer from low accuracy with a median error of 479 km per user. Instead of investing in Bayesian models with a large margin of error, in this work we use simple reliable methods to extract place names from tweets and user profiles. Such a solution results in identifying the location of a relatively smaller set of users while providing high accuracy.

Geo-tagging is an important task for other social networks like Facebook as well. In a recent work, Backstrom et al. [3] predict the addresses of 1.6 million Facebook users based on the addresses of 700,000 other users. Their methodology that leverages from the friendship graph correctly places 57.4% of users within 25 miles of their provided locations. We do not leverage from this methodology in this paper for various reasons. For one, their methodology requires an expensive preprocessing phase. Since our goal is to solve the geotrend detection in an online manner, such a technique is troublesome in a highly dynamic setting with new users and follow relations being formed frequently. Secondly, Facebok and Twitter have different natures, therefore it is not clear whether the technique would be as beneficial in the context of Twitter. While it is an interesting research problem to compare the nature of geo-information in Twitter and Facebook, this is not the research question addressed in this paper. Finally, we do not leverage from friend relations in our methodology, we believe such a solution, not relying on having access to full graph, is more accessible for implementation.

Geographical Distribution of Twitter Updates: In order to provide an overview of the geographical characteristics of our data set, we present heat maps of locations that tweets originate from (Figure 1(a)) and locations tweets are about (Figure 1(b)). In both graphs, we plot every city associated with more than 10 tweets using the GeoMap tool of Google Charts. The color and size of cities are proportional to the number of tweets. The two figures resemble each other but there are certain distinctions. For instance, Japan is denser in Figure 1(b) due to the Japan earthquakes that took place within the time period captured in our data set. On the contrary, a drop in significance can be observed for countries such as Indonesia when comparing the tweets *in* cities to tweets *about* cities. This difference is due to the fact that Indonesia is a highly active country in Twitter [17], while there are no important events taking place in its cities that would result in people mentioning them. Note that we also analyzed the number of users per location and the results were similar to Figure 1(a).



(a) Tweets *in* Cities



(b) Tweets *about* Cities

Figure 1: Heat Map for # of tweets in/about cities of the world

Characterizing geo-correlation of twitter "friends" : To further demonstrate the usefulness of geo-analysis we analyzed the correlation between friends and their location. Instead of using the static "following" relation to define friendship, we denote the users which mutually interact (mention each other) as "friends". Due to space limitations, we omit various details of this analysis and note that approximately 57.4% of "friends" reside in the same city while about 62% of users have at least one friend in the same city. These results show *locality* in friendship relations in Twitter confirming earlier work [37]. Research shows that this *local* behavior in friendship formation extends also to more dynamic behavior, i.e. topical interests [16]. In these circumstances, it becomes vital to detect such local interests. Given that interests evolve over time, it is also crucial to carry this task in an online manner. This task is exactly what *GeoWatch* addresses.

4. DETECTING GEO-TRENDS

In this section, our goal is to identify the characteristics that comprise a useful geo-trend detection tool. We aim to define which locations, topics as well as correlations are necessary and sufficient to report to provide a rich geo-trend detection tool. These charac-

teristics lead to the three main premises of our algorithmic design.

A basic geo-trend detection tool should provide a high level overview of the popularity of *locations* and *topics*. Such a tool should answer queries such as *"What fraction of the mentions in the current time window are about topic t_x (or from location l_i)?"* efficiently and accurately. This notion can be formalized by the following premise:

PREMISE 1. *The frequency of any topic t_x and any location l_i in the current time window should be reported in an accurate and timely fashion.*

This premise ensures tracking global trends in the social network. Not only can one identify the interesting topics but also keep track of most active geographical locations in the network. This task can be achieved by traditional heavy hitters approaches and has already been addressed to a large extend in recent research. In this paper, we aim to reach beyond that and identify *geo-trends* that provide the link between the topics and locations by capturing the correlation between the two. Consider a stream consisting of pairs (l_i, t_x) where l_i is the *geo-origin* of a tweet and t_x is the topic of the tweet. In this context, geo-trends can be captured through the following premise:

PREMISE 2. *All* significantly correlated *location-topic pairs can be retrieved at any particular time in an efficient and accurate manner. A location-topic pair (l_i,t_x) is* significantly correlated *if at least ϕ fraction of all mentions* from *location l_i are about topic t_x and at least ψ fraction of all mentions about topic t_x are* from *location l_i.*

Consider the stream $\{(l_1,t_1),(l_2,t_1),(l_3,t_1),(l_1,t_2),(l_1,t_3),(l_2,t_3),(l_2,t_3)\}$. Assume that $\phi = \psi = 0.5$. The only correlation that will be reported based on Premise 2 is (l_2,t_3). For instance correlation (l_1,t_2) is not reported even though l_1 is a heavy hitter for t_2 since t_2 is not central to the interests of l_1, at least based on the threshold setting $\phi = 0.5$. A similar filtering can be observed for the correlation (l_3,t_1) since t_1 is a global trend, appearing equally in all three locations and hence, l3 is not special in any geographic sense for topic t1. Through this premise, the geo-trend detection tool captures the interests of different localities and provides means for serving important applications such as crisis management.

Note that, we rely on parameters ϕ and ψ rather than relying on the definition of statistically significant correlation. Statistical analysis can compute the association strength between a pair of location and topic by comparing their *expected* and *observed* frequencies. The χ^2 statistic is a classical method that is widely used for this type of analysis. While, the notion of statistical significance [14] is an interesting and useful concept, the application of statistical methods such as χ^2 test would rejected most null hypotheses, i.e. a location-topic pair not being correlated, due to the large sample size [5]. This would result in unmanagable or even meaningless correlations being detected. Therefore, we believe leaving the choice of ϕ and ψ to be determined based on the specific application is more practical and useful compared to the detection of statistically significantly correlated location-topic pairs. However, we still believe further investigation in the line of identifying new statistical methods for capturing correlations is an interesting problem and aim to work on this as future work.

One of the important characteristics of a useful trend detection tool is its ability to filter out *insignificant* information. Therefore, given the large number of locations and topics as well as their zipfian distribution of popularity, a scalable and useful trend detection tool should also filter out *unpopular* correlations. Consider a hypothetical location l_i consisting of only one user who uses a highly uncommon hashtag h_x. If there are no restrictions on the significance of locations the pair (l_i,h_x) would be reported as a correlated pair. Given the Zipfian nature of popularity of locations and topics, it is easy to see that the list of correlations involving such locations would grow large. In order to avoid reporting an unmanageably large list of location-topic correlations, there should be a lower bound on the importance of a given location for it to be reported by the geo-trend detection tool. This leads us to the final premise of our algorithm:

PREMISE 3. *Geo-trend detection should identify a list of "all" and "only" the locations that are at least θ-frequent in the current time window and limit the reported correlations to such locations.*

A θ-frequent location in a window of N elements is a location that occurs at least θN times where $0 \leq \theta \leq 1$. Through this premise, geo-trend detection is guaranteed to capture significant locations while also keeping the number of reported locations at a manageable size. Such a requirement also filters out locations for which there is not enough data to infer the geographical interest. Given that Premise 2 dictates a correlation to be reported *only* if *both* the location and the topic are heavy-hitters for each other, Premise 3 also ignores unpopular topics by eliminating unpopular locations. This is because unpopular topics cannot be frequent for popular locations; the only locations tracked for correlations.

So far we defined *geo-trends* where locations represent the *geo-origin* of information. A similar definition could be constructed to detect the correlations in a stream of pairs (l_j,t_y) where l_j is the *geo-focus* of a tweet and t_y is the topic of that particular tweet. In this case, from Premise 2, the location-topic pair (l_y,t_y) is significantly correlated if at least ϕ fraction of all mentions *about* location l_i are also about topic t_x and at least ψ fraction of all mentions about topic t_x are *about* location l_i. Also note that the correlations reported are filtered due to Premise 3 meaning no correlation whose geo-focus is a location with less than $\theta * N$ occurrences in a window of N elements is reported. This way the list of correlated pairs are kept at a manageable size.

In the following sections we will first provide the formal problem definition that addresses the three premises introduced here. Next we will prove that all location-topic pairs need to be tracked for an exact solution which introduces scalability challenges which are addressed through our proposed technique, *GeoWatch*, that requires sublinear memory and processing time.

4.1 Problem Definition

We denote the set of all topics as $T = \{t_1,t_2,...,\}$ and the set of all locations as $L = \{l_1,l_2,...\}$, $|T|$ and $|L|$ denote the number of topics and locations respectively. Since tweets are restricted to at most 140 characters long, Twitter users use hashtags to convey their thoughts in a compact manner [31]. Therefore, we choose *hashtags* to capture topics in this study. As for the definition of locations, we focus on *cities* since this resolution allows capturing local interests while not being too small as to result in meaningless correlations. This choice also allows us to map our findings to real events that happen in different cities of the world. In the following sections we will assume that the number of distinct hashtags and locations

are known in advance and do not change. However, in a highly dynamic setting such as social networks, the set of topics itself is also dynamic. We note that our solution also works for such cases by simply creating larger sketches as the data range grows [20].

Given a stream S of location-topic *pairs* of the form (l_i, t_j) and three user defined frequency thresholds θ, ϕ, and ψ in the interval [0, 1]; our goal is to keep track of **(i)** the frequencies $F(l_i)$ ($F(t_x)$) of all locations l_i (topics t_x) and **(ii)** all pairs (l_i, t_x) s.t. $F(l_i) > \lceil \theta N \rceil$, $F(l_i, t_x) > \lceil \phi F(l_i) \rceil$, and $F(l_i, t_x) > \lceil \psi F(t_x) \rceil$ in the current time window. Here $F(l_i, t_x)$ is the number of *pairs* on topic t_x from location l_i; $F(l_i)$ is the number of the *pairs* from l_i in the current time window; and $F(t_x)$ is the number of *pairs* on t_x. The window size can be set in terms of maximum number of elements or an actual time window such as an hour or a day. In the latter case, the number of elements N in the current window is defined by the user. Since frequency of each topic and location is tracked, Premise 1 is satisfied. As all the correlated pairs are determined, Premise 2 is captured by definition. And finally, by setting the requirement $F(l_i) > \lceil \theta N \rceil$ we address Premise 3.

4.2 Exact Solution

An exact solution that solves the problem described in Section 4.1 requires keeping track of all possible pairs in a given window. We will prove this statement, by focusing on Premise 2 alone. The full solution that also satisfies Premises 3 and 1 is at least as hard.

THEOREM 4.1. *Any exact solution for the problem of detecting geo-correlated trends in a sliding window requires keeping exact and complete information about all pairs in the given window.*

PROOF. Given a stream $S = \{..., t_{i+1}, t_{i+2}, ..., t_{i+m}, ...\}$ and a window size m, construct a 2-dimensional stream as follows, $S' = \{..., (l_1, t_{i+1}), (l_1, t_{i+2}), ..., (l_1, t_{i+m}), ...\}$, by appending some location l_1 as the first value for all pairs. An answer to the query about correlations at time step $i+m$ in the constructed stream with thresholds ϕ and $\psi = 1 - \frac{1}{m}$ and $\theta = 1$ can be directly translated into an answer to a query about frequent elements in the original stream with threshold ϕ. Therefore, answering the correlated geo-trend query in S' is equivalent to answering frequent elements query in S which requires complete information about all elements. □

Next we focus on the implications of Theorem 4.1. There are over 50K cities and over 2.3M unique hashtags in our dataset which results in over 115 billion different possible pairings. It is also important to consider the rate at which information is shared in social networks. For instance, there are on average 140 million tweets shared on Twitter per day [36]. It is easy to see that as the number of topics and locations become large, the exact solution of keeping track of all possible pairs of locations and topics becomes infeasible. Therefore, we next propose our method with sub-linear memory and processing requirements.

4.3 GeoWatch

Given the infeasibility of the exact solution, we now propose *GeoWatch* that requires a sublinear memory and amortized running time while still providing accuracy guarantees. The main idea behind *GeoWatch* is to limit the number of monitored locations by tracking those that are at least θ-frequent and to further limit the number of monitored topics by tracking a topic t_x *only if* t_x is ϕ-frequent for at least one location and track ψ-frequent locations for each such topic. Given that there can be at most $\lceil \frac{1}{\theta} \rceil$ θ-frequent locations at a given time, each of which can have up-to $\lceil \frac{1}{\phi} \rceil$ topics that are ϕ-frequent, the number of elements to track can be bounded by a small number. As we will demonstrate later on, in order to provide accuracy guarantees, *GeoWatch* relaxes the number of locations to track from $\lceil \frac{1}{\phi} \rceil$ to $\lceil \frac{1}{\phi - \varepsilon} \rceil$ where $\varepsilon \ll \theta$.

4.3.1 GeoWatch Data Structures

An overview of the structure of *GeoWatch* is provided in Figure 2. In this section we briefly describe *GeoWatch* and its subcomponents. As can be seen from Figure 2, *GeoWatch* consists of two main components. *Location-StreamSummary-Table* is a hashtable that contains a $StreamSummary_{l_i}$ structure for each location l_i that has a current estimated relative-frequency of at least θ. Note that the estimated relative-frequency is never an underestimation, therefore all location with at least θ relative-frequency are guaranteed to be in *Location-StreamSummary-Table*. In order to provide a solution in a sliding window where deletions as well as insertions of elements need to be supported, *Location-StreamSummary-Table* also includes a sketch structure. This sketch structure is maintained to keep track of frequencies of locations in a sliding window by allowing both insertion and deletion operations [20]. $StreamSummary_{l_i}$ monitors the ϕ-frequent topics for location l_i. Since deletions need to be supported to maintain the list of ϕ-frequent topics as well, this summary structure is also maintained through a sketch-based solution. Consider a case where a pair (l_i, t_x) that expired is to be deleted from the data structures. In this case, $StreamSummary_{l_i}$ should only be updated to reduce $F(l_i, t_x)$ if (l_i, t_x) occurred after $StreamSummary_{l_i}$ was created. Therefore $StreamSummary_{l_i}$ also includes a time-stamp TS_{l_i} recording the time it was created. In the case where the window size is set based on the maximum number of elements rather than real time, the time-stamp will be based on a discrete notion of time that is based on the sequence number of mention pairs in stream S.

The second component given in Figure 2 is the *Topic-StreamSummary-Table*, a hashtable that monitors the topics that are potentially correlated with at least one location and a sketch structure to keep track of the topic frequencies. Through such an implementation, Premise 1 can also be addressed. The topics in this table are determined by the topics that appear in at least one $StreamSummary_{l_i}$ for location l_i that is θ-frequent in the current window. For each such topic t_x in *Topic-StreamSummary-Table*, there is a data-structure pair $< Count_{t_x}, StreamSummary_{t_x} >$ where $count_x$ is the number of locations t_x is ϕ-frequent for and $StreamSummary_{t_x}$ monitors the ψ-frequent locations for topic t_x. $StreamSummary_{t_x}$ will be maintained as long as $count_x$ is positive. As soon as this number reaches 0 for topic t_x, the structure $StreamSummary_{t_x}$ is deleted freeing the space used by $< Count_{t_x}, StreamSummary_{t_x} >$. Similar to stream summary structure for locations, $StreamSummary_{t_x}$ includes a time-stamp TS_{t_x} of when $StreamSummary_{t_x}$ was created.

An important sub-component of *GeoWatch* that is leveraged in both *Location-StreamSummary-Table* and *Topic-StreamSummary-Table* is the sketch structure. This structure consists of a hashtable, *S[m][h]*, along with h hash functions. Given a range of elements from 1 to M, an item k in this range has a set of h associated counters and these counters are increased (or decreased) when encountering an insert (or delete) operation of element k. Clearly, the values for m and h should be set such that the collisions are minimized and guarantees can be given for bounds on overestimation. It has been shown that, $\frac{e}{\varepsilon} . \ln(\frac{-M}{\ln p})$ counters are needed to estimate each item

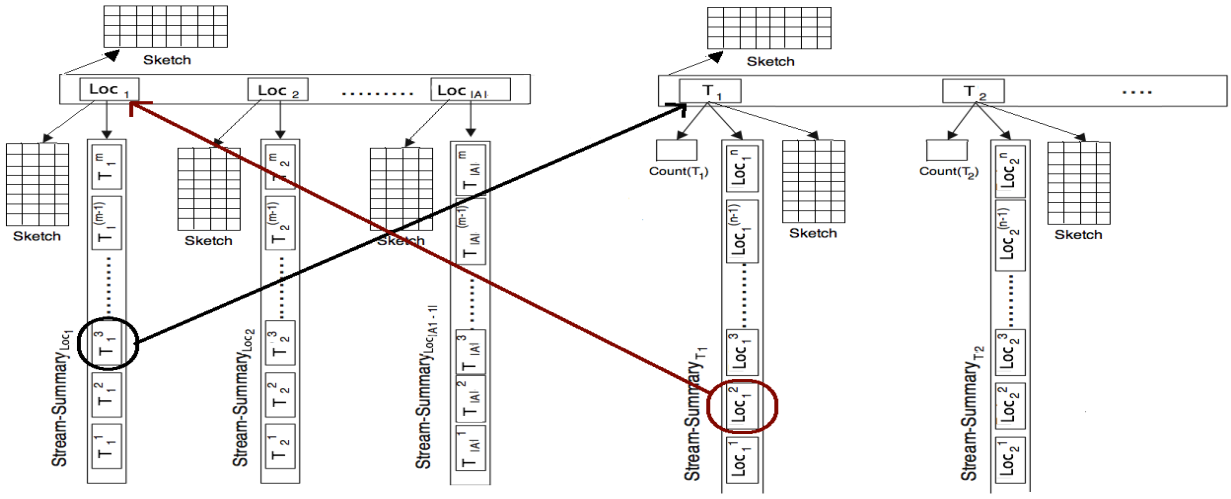


Figure 2: Overview of *GeoWatch* Data Structures: *Location-StreamSummary-Table* (on the left) keeps track of ϕ-frequent topics for θ-frequent locations. *Topic-StreamSummary-Table* (on the right) keeps track of ψ-frequent locations for each topic that is ϕ-frequent for at least one location. Here the third most important topic for location Loc_1 is T_2 and the second most important location for topic T_2 is Loc_1

with error no more than εN in a window of size N with probability p by setting $m = \frac{e}{\varepsilon}$ and $h = \ln(\frac{-M}{\ln p})$ [20].

Given that the ϕ-frequent topics for a given location l_i are tracked only after l_i becomes θ-frequent and a topic t_x is tracked only after it becomes ϕ-frequent for at least one location, we need to show how *GeoWatch* satisfies Premises 3, 1 and 2. To this end, we first give the intuition as to how these premises are still satisfied under our approximation. Premises 3 and 1 are relaxed to allow for a small error ε and to be guaranteed probabilistically. For this purpose, *GeoWatch* requires two additional parameters ε and p in addition to the parameters θ, ϕ and ψ as described in Section 4.1. The parameter ε captures the allowed error rate while p captures the probability of remaining within this error rate.

In reference to Premise 1, instead of guaranteeing to capture the relative frequency of each topic and location exactly, *GeoWatch* guarantees that for *any* topic t_x and any location l_i, its true relative-frequency is overestimated by no more than ε with probability p but never underestimated. Note that theoretically, the ε and p values used to determine the error for locations and topics could potentially be distinct values. In this paper, for ease of presentation we choose the same ε and p values for locations and topics. Also, in reference to Premise 3, even though an exact counter for each location is not kept, through the use of the sketch structure in *Location-StreamSummary-Table*, *GeoWatch* guarantees detecting *all* locations l_i s.t. $F(l_i) \geq \theta N$. It also guarantees that no location l_j s.t. $F(l_j) < (\theta - \varepsilon)N$ is reported. Lastly, the relative frequencies of locations are overestimated by no more than ε with probability p but never underestimated.

In reference to Premise 2, *GeoWatch* guarantees capturing all *trending* correlated pairs of locations and topics rather than all correlated pairs. Here the notion of *trending* refers to non-decreasing significance. Most importantly, *GeoWatch* satisfies this premise deterministically which guarantees perfect recall values. While it is important to capture correlations in general, the more important task is to detect *trending* correlations, i.e. correlations that have an increasing value over time. For instance, consider two hypothetical correlations *(Los Angeles, #405Traffic)* and *(Los Angeles, #earthquake)*. Traffic in 405 freeway in Los Angeles is a general topic of interest resulting in a stable interest in the topic. In contrast, a recent hypothetical earthquake would result in *increasing* interest and therefore increasing value of correlation. While capturing both cases is important, it is crucial to *guarantee* capturing the latter. Even though *GeoWatch* is only guaranteed to capture the trending correlations, as we will demonstrate in Section 5 it in fact captures all correlated pairs for various θ, ϕ and ψ settings. Similarly, even though there are no guarantees on the precision performance, as we show in Section 5, *GeoWatch* provides near-perfect precision.

4.3.2 GeoWatch Operations

There are three operations that are allowed at a given point; insert, remove and report operations. Each incoming stream element of the form (l_i, t_x) needs to be inserted into the data structure. As the sliding window moves along, expired mentions should be removed. Note that a sliding window can be set either in terms of number of elements to be maintained or the period of time defined in terms of minutes, hours, days etc. The pseudo-code for insert and remove operations are provided in Algorithms 1 and 2. Due to space limitations, we omit the pseudocode for the report algorithm that goes through the structures and reports correlated pairs.

In Algorithm 1, lines (1-15) perform updates due to the occurrence of l_i. Lines (1-8) capture the steps that need to be taken to incorporate the addition of the new mention in location l_i. If t_x becomes ϕ-frequent for location l_i after this insertion, *Topic-StreamSummary-Table* needs to be updated to increase the number of locations t_x is trendy for. If this count was zero before this operation, a new $StreamSummary_{t_x}$ will be created with timestamp ts and counter 1. Since the number of items increase with an insert operation, it is possible that a location whose frequency is stable becomes θ-infrequent. Lines (9-12) remove such items and consequently updates the *Topic-StreamSummary-Table* for topics that were ϕ-frequent for such locations. Decreasing $Count_{t_y}$ also entails removing $StreamSummary_{t_x}$ if the counter becomes 0. Similarly,

Algorithm 1 Insert (l_i, t_x, ts)

1: $F(l_i) \leftarrow F(l_i) + 1$
2: **if** l_i turned θ-frequent **then**
3: Create $StreamSummary_{l_i}$ with timestamp ts for location l_i
4: **if** l_i is θ-frequent **then**
5: $F_{l_i}(t_x) \leftarrow F_{l_i}(t_x) + 1$
6: **if** t_x turned ϕ-frequent for l_i **then**
7: $StreamSummary_{l_i} = StreamSummary_{l_i} \cup \{t_x\}$
8: Increase $Count_{t_x}$
9: **for all** l_j turned θ-infrequent **do**
10: **for all** $t_y \in StreamSummary_{l_j}$ **do**
11: Decrease $Count_{t_y}$
12: Delete $StreamSummary_{l_j}$
13: **for all** t_y turned ϕ-infrequent for location l_i **do**
14: $StreamSummary_{l_i} = StreamSummary_{l_i} \setminus \{t_y\}$
15: Decrease $Count_{t_y}$
16: $F(t_x) \leftarrow F(t_x) + 1$
17: **if** $t_x \in$ *Topic-StreamSummary-Table* **then**
18: $F_{t_x}(l_i) \leftarrow F_{t_x}(l_i) + 1$
19: **if** l_i turned ψ-frequent for t_x **then**
20: $StreamSummary_{t_x} = StreamSummary_{t_x} \cup \{l_i\}$
21: **for all** l_j turned ψ-infrequent for t_x **do**
22: $StreamSummary_{t_x} = StreamSummary_{t_x} \setminus \{l_j\}$

since the number of mentions in location l_i increased, there could be topics whose frequency is stable and yet became ϕ-infrequent. Such cases are handled through lines (13-15). Starting from line 16, the changes to *Topic-StreamSummary-Table* are performed to capture the mention about topic t_x. First, the value of t_x is increased irrespective of the topic being tracked or not to satisfy Premise 1. Next if t_x is already being tracked, $StreamSummary_{t_x}$ is updated to capture the new mention from location l_i.

In Algorithm 2 we present the steps that need to be taken upon a remove operation. Here Lines (1-11) are for incorporating the reduction in the mentions from l_i while Lines (12-17) are for incorporating the deletion of t_x. Note that when an element is deleted the total number of elements in the given window decreases. In this case, there could potentially be a location l_j whose frequency is stable yet becomes θ-frequent. In order to avoid checking the frequency of each currently θ-infrequent location with every remove operation which would hurt the efficiency of *GeoWatch*, we omit the creation of such $StreamSummary_{l_j}$. Even if such a summary were to be created, the set of topics in it would be empty. Therefore there is no penalty in omitting this action, the next time there is a mention from l_j, this stream summary will be created. The same is true for topics becoming ϕ-frequent for l_i, or locations becoming ψ-frequent for t_x. All such operations are omitted for efficiency purposes, while preserving precision and the described guarantees.

It is a important task to obtain bounds on memory and running time as well as the performance guarantees for *GeoWatch*. Next we present such proofs, starting with runtime bounds for insert and remove operations, and then with the memory requirements for satisfying Premises 3-to-2. Finally, we prove that *GeoWatch* is guaranteed to capture all *trending* correlated pairs.

4.3.3 Running Time and Memory Requirements

Memory Requirements: A feasible geo-trend detection solution should be sub-linear in its space usage given the large scale of data.

Algorithm 2 Remove (l_i, t_x, ts)

1: $F(l_i) \leftarrow F(l_i) - 1$
2: **if** l_i is θ-frequent **then**
3: **if** $TS(StreamSummary_{l_i}) \leq ts$ **then**
4: $F_{l_i}(t_x) \leftarrow F_{l_i}(t_x) - 1$
5: **if** t_x turned ϕ-infrequent for l_i **then**
6: $StreamSummary_{l_i} = StreamSummary_{l_i} \setminus \{t_x\}$
7: Decrease $Count_{t_x}$
8: **if** l_i turned θ-infrequent **then**
9: **for all** $t_y \in StreamSummary_{l_i}$ **do**
10: Decrease $Count_{t_y}$
11: Delete $StreamSummary_{l_i}$
12: $F(t_x) \leftarrow F(t_x) - 1$
13: **if** $t_x \in$ *Topic-StreamSummary-Table* **then**
14: **if** $TS(StreamSummary_{t_x}) \leq ts$ **then**
15: $F_{t_x}(l_i) \leftarrow F_{t_x}(l_i) - 1$
16: **if** l_i turned ψ-infrequent for t_x **then**
17: $StreamSummary_{t_x} = StreamSummary_{t_x} \setminus l_i$

In this section we provide proofs that *GeoWatch* is sub-linear in both the number of locations and topics.

THEOREM 4.2. *The method requires* $O(\frac{e}{\varepsilon*(\theta-\varepsilon)}(\ln(-\frac{|T|}{\ln(p)}) + \frac{\ln(-\frac{|L|}{\ln(p)})}{\phi-\varepsilon}) + \frac{1}{(\theta-\varepsilon)(\phi-\varepsilon)(\psi-\varepsilon)})$ *memory.*

PROOF. There are two main substructures: location table and topic table. The location table consists of the main sketch structure that tracks occurrences of locations in the window and requires $m_l * h_l$ counters. In order to fulfill Premise 3 that entails estimating the frequency of locations with error no more than εN with probability p, $m_l = \frac{e}{\varepsilon_l}$ and $h_l = \ln(-\frac{|L|}{\ln p_l})$ [20]. At a given time there are up to $\lceil \frac{1}{\theta-\varepsilon_l} \rceil$ locations being tracked for which a list of top topics should be maintained. For each of these $\lceil \frac{1}{\theta-\varepsilon_l} \rceil$ locations, $m_{lt} * h_{lt}$ counters are required for the sketch structure s.t. $m_{lt} = \frac{e}{\varepsilon_{lt}}$ and $h_{lt} = \ln(-\frac{|T|}{\ln p_{lt}})$ since pairs also need to be maintained to satisfy Premise 2. For each location, up to $\lceil \frac{1}{\phi-\varepsilon_{lt}} \rceil$ topics are tracked.

The second main substructure is for keeping track of important topics. The topics table consists of the main sketch structure that tracks occurrences of topics in a given window and requires $m_t * h_t$ counters. In order to fulfill Premise 1 that entails capturing topic frequencies correctly, these values should be set as $m_t = \frac{e}{\varepsilon_t}$ and $h_t = \ln(-\frac{|T|}{\ln p_t})$. For each tracked topic, a list of locations needs to be tracked. Since there are at most $\lceil \frac{1}{\theta-\varepsilon_l} \rceil$ locations tracked and for each location there are at most $\lceil \frac{1}{\phi-\varepsilon_{lt}} \rceil$ topics tracked, there are at most $\lceil \frac{1}{\theta-\varepsilon_l} \rceil \lceil \frac{1}{\phi-\varepsilon_{lt}} \rceil$ distinct topics in the topic table. For each of those topics, $m_{tl} * h_{tl}$ counters are required for the sketch structure s.t. $m_{tl} = \frac{e}{\varepsilon_{tl}}$ and $h_{tl} = \ln(-\frac{|L|}{\ln p_{tl}})$ since pairs also need to be maintained to satisfy Premise 2. In addition, there are at most $\lceil \frac{1}{\psi-\varepsilon_{tl}} \rceil$ locations tracked and for each topic. Adding all those together, and simplifying the system by setting all $\varepsilon_{\{l,t,lt,tl\}} = \varepsilon$ and $p_{\{l,t,lt,tl\}} = p$, in total, the memory requirement sums up to $O(\frac{e}{\varepsilon*(\theta-\varepsilon)}(\ln(-\frac{|T|}{\ln(p)}) + \frac{\ln(-\frac{|L|}{\ln(p)})}{\phi-\varepsilon}) + \frac{1}{(\theta-\varepsilon)(\phi-\varepsilon)(\psi-\varepsilon)})$. □

Running time requirements: There are two possible update operations at a given time: an insert or a remove of a location-topic pair. Both of these operations have amortized log-linear running time. Due to space limitations, we skip the proof for the *remove* operation and note that it is very similar to the proof provided for the *insert* operation as provided below:

THEOREM 4.3. *The amortized running time for an insert operation in GeoWatch is* $O(log(-\frac{|T|}{log(p)}) + log(-\frac{|L|}{log(p)}))$

PROOF. The steps that need to be taken for an insert are given in Algorithm 1. Line 1 requires updating the sketch structure which entails $h = log(-\frac{|L|}{log(p)}))$ operations. Lines 2-3 create an empty stream structure if l_i becomes θ-frequent with the insertion of the new item. This clearly is a constant time operation. In the case where l_i was (or became) θ-frequent (Lines 4-8), $StreamSummary_{l_i}$ needs to be updated to include the addition of t_x. This entails updating $StreamSummary_{l_i}$ and possible insertions/deletions of the ϕ-frequent topics for l_i. Even with a conservative setting for the sketch structure that assumes all topics can be mentioned at a given location, the sketch update requires $h = log(-\frac{|T|}{log(p)})$ operations and the updates to the substructure is amortized-constant time. For the locations that have become θ-infrequent (Lines 9-12), the deletion operation is also constant time, however, with a non-constant number of such topics, the number of operations can become quite large. Since a location can only be deleted as many times as it is inserted to the stream summary and since by construction, a location l_j is inserted into the summary only when there is a tuple (l_j, t_y), we can conclude that the deletion operation has amortized constant time. Lines (13-15) requires amortized constant time for the same reason. In order to keep track of frequent global-level topics, sketch structure for topics is updated regardless of the topic being tracked or not requiring $h = log(-\frac{|T|}{log(p)})$ operations (Line 16). If topic t_x is being tracked (Lines 17-22), $StreamSummary_{t_x}$ has to be updated which entails updating the sketch structure for t_x(Line 18: $h = log(-\frac{|L|}{log(p)})$)), adding l_i to $StreamSummary_{t_x}$ if it became ψ-frequent (constant time) and deleting locations that became infrequent for t_x (amortized constant time). Adding all those operations together, amortized processing time for an insert is $O(log(-\frac{|T|}{log(p)}) + log(-\frac{|L|}{log(p)}))$ □

4.3.4 GeoWatch Accuracy Guarantees

Although *GeoWatch* monitors the traffic of locations and topics approximately, its accuracy is very high. As we prove in Theorem 4.4, *GeoWatch* has *guaranteed* perfect recall in detecting *trending* correlated pairs, where trending is defined based on non-decreasing relative frequency. We show in Section 5 that *GeoWatch* in practice succeeds in detecting all correlated pairs rather than only those that are *trending*. It also has a near perfect precision.

THEOREM 4.4. *At any given time ts, all trending correlated pairs of the time window ending at ts are reported by GeoWatch.*

PROOF. Consider a particular time window that spans over the period $[ts-w, ts]$, where ts is the end of the window and w is the time window size and includes N tuples. We now show that for any trending correlated pair (l_i, t_x), in this time period, is guaranteed to be captured. We loosely define a pair (l_i, t_x) with increasing frequency in a given time window as *trending*. In that perspective, $G_{[ts', ts]}(l_i, t_x) \leq G_{[ts-w, ts]}(l_i, t_x)$ where $ts - w \leq ts' \leq ts$ and $G_{[ts_1, ts_2]}(l_i, t_x) = \frac{F_{[ts_1, ts_2]}(l_i, t_x)}{F_{[ts_1, ts_2]}(l_i)}$ and $F_{[ts_1, ts_2]}(l_i, t_x)$ denotes the number of occurrences of the tuple between the time frames ts_1 and ts_2 and $F_{[ts_1, ts_2]}(l_i)$ denotes the number of occurrences of location l_i. Similarly, capturing the *trending* characteristics of the (l_i, t_x) pair, $H_{[ts', ts]}(l_i, t_x) \leq H_{[ts-w, ts]}(l_i, t_x)$, where $H_{[ts_1, ts_2]}(l_i, t_x) = \frac{F_{[ts_1, ts_2]}(l_i, t_x)}{F_{[ts_1, ts_2]}(t_x)}$. Since (l_i, t_x) is a trending correlated pair, by definition $F(l_i) \geq \theta N$ and therefore l_i is guaranteed to be tracked, let the time l_i starts being tracked be denoted by ts_{l_i} s.t. $0 \leq ts_{l_i} \leq ts$. Given the *trending* property, $G_{[ts_{l_i}, ts]}(l_i, t_x) \geq G_{[ts-w, ts]}(l_i, t_x) \geq \phi * F_{[ts-w, ts]}(l_i)$. Therefore, topic t_x will also be tracked at a time $ts_{lt} \leq ts$ which means t_x is guaranteed to be captured in the topics table. Since $H_{[ts_{lt}, ts]}(l_i, t_x) \geq H_{[ts-w, ts]}(l_i, t_x) \geq \psi * F_{[ts-w, ts]}(t_x)$, location l_i will also be tracked for topic t_x. Given the trending property, such frequencies will only increase in time guaranteeing that by ts, the pair will sustain its correlated property. □

With the sublinear memory and running time requirements as well as the accuracy guarantees, *GeoWatch* is a practical tool to detect geo-trends in social networks.

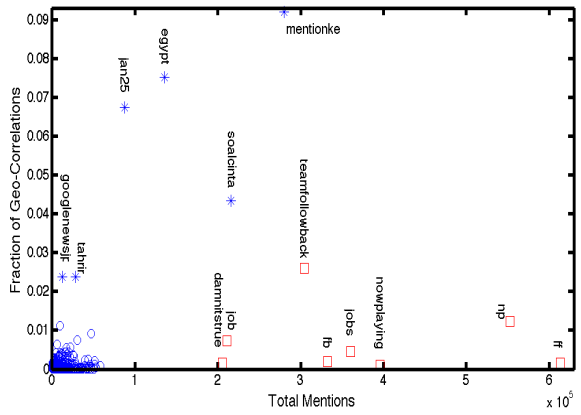
5. EXPERIMENTS

In this section, we provide a detailed experimental analysis of *GeoWatch*. First, we demonstrate the value of such data analysis by focusing on the types of topics and locations that are detected by *GeoWatch*. Next we evaluate the effect of parameters θ, ϕ, ψ as well as the window size on the accuracy and efficiency of *GeoWatch* (sensitivity analysis). Throughout those experiments we chose $\varepsilon = 0.0004$ and $p = 0.99$ to allow for small error.

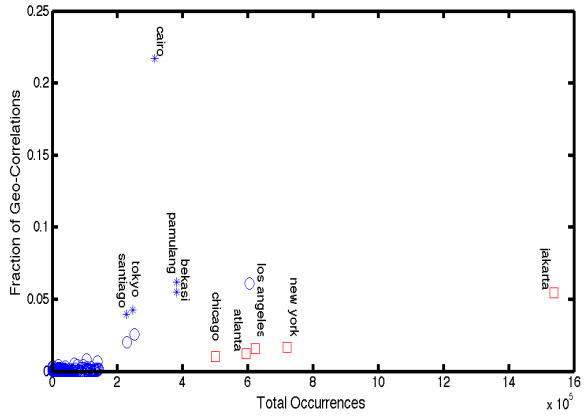
5.1 Geo-correlation and Twitter Analysis

In this section, in addition to evaluating the correlations detected by *GeoWatch*, we address the following two questions: *Are there topics that carry a higher geo-significance?* and *Are there locations that cause or exhibit local topical interests?* These two questions can be answered through the analysis enabled by *GeoWatch*. To address the first question, in Figure 3(a) we show the relation between the geo-significance of topics and the total number of times they are mentioned in the data set measuring their global importance. The geo-significance of a topic is measured in terms of the fraction of all the correlated pairs it appears in the entire stream. We chose a time window of 24 hours in this experiment and set $\theta = 0.005, \phi = \psi = 0.05$. *GeoWatch* provides means for reporting correlated pairs at any given time. For this experiment we chose 10 minutes as the reporting window, i.e. every ten minutes the correlated pairs at that particular time are reported. Note that the reporting window and the time window are two distinct values. The time window refers to the length of the sliding window while the reporting window reflects how frequently the report operation is called to determine the current list of correlations.

For ease of viewing, we eliminated all hashtags that had no correlations reported, which reduced the number of data points drastically from over 2 million to approximately 250. This indicates that even though there is a large number of topics discussed in social networks, there is only a small number of topics that carry significance in different localities. There are various hashtags that have high global significance while being much less important as a geographical trend such as #ff, #np, #jobs (represented by squares in



(a) Hashtags



(b) Cities

Figure 3: Geo-significance vs. trendiness of hashtags and cities

Figure 3(a)). For instance, #ff refers to "follow friday" and is a popular hashtag used in Twitter. Similarly, #jobs, referring to issues related to jobs, is a common hashtag that is of interest to Twitter users in the global scale. Unlike these topics that are of interest to the entire network, hashtags such as #jan25, #egypt, #googlenewsjp (represented by stars in Figure 3(a)) are a lot more significant as a geographical trend. The first two of these hashtags relate to recent uprisings in Egypt while #googlenewsjp is mostly used to discuss issues about the Fukushima earthquake in April 2011.

We performed a similar analysis to capture which cities carry geo significance, i.e. cities whose residents are interested in local topics. For this purpose we plot the number of correlations a given city appears in versus the number of tweets originating from that particular city. As can be seen from Figure 3(b), the static representation of a city measured by the number of tweets originating from it, is not representative of the geo-significance of that place. For instance, there is a relatively small number of tweets originating from Cairo but due to those tweets being mostly about local events (the recent political uprising) they have a high geo-significance. Another city with a large number of geo-correlations is Santiago. Examples of detected correlations for this city include sports related hashtags (e.g. #bielsa) and cultural events and TV programs (e.g. #wewantsupershowinlatinoamerica). On the contrary, we see that Jakarta, a city where a large number of identified users reside, does not appear in a large number of correlations, meaning that users from this area are in general less concerned about local events.

The analysis provided so far focused on the cumulative geo-significance of topics and locations but *GeoWatch* provides a more useful tool that can capture geo-significance of topics or locations along a temporal dimension as well by detecting correlations along a sliding window. There is a large number of interesting topics detected at particular points in time but do not appear in Figures 3(a) or 3(b) due to their short lived activity. A few examples include: Iwaki aftershock on April 11, as well as the main Japan earthquake on March 11. On these days the hashtag #earthquake is detected to be correlated with Tokyo due to a large number of Twitter users from Tokyo mentioning this topic. Such behavior signals that *GeoWatch* can be used in crisis management as it detects the emergency event in a fast and automated manner. However, local interests detected by *GeoWatch* are not only limited to emergency events. *GeoWatch* also provides a good depiction of the population pulse at a given location. Not only can the political interests of a population be captured as in the case of correlated pairs, such as *(Cairo, #Jan25)*, but it can also capture other, more casual interests. For instance, a large number of correlated pairs involving Soccer teams appear in British cities, especially compared to other cities of the world, indicating a high British interest in this sport. Examples of this type of correlation include *(Notttingham, #NewCastle)* or *(Liverpool, #lfc)*.

Local and short lived events, such as political demonstrations and cultural events, are also among the topics captured by *GeoWatch*. As an example, the correlated pair *(Madrid, #11m)* is captured due to the demonstrations in Madrid on the anniversary of bombings that happened on March 11 2004, killing 191 people. Examples of detected cultural events include the correlated pair *(Austin, #sxsw)* that is due to the SXSW festival on March 16 2011 in Austin. Other correlation pairs appear in the general form of *(city, #city)*. This is due to the fact that Twitter users use hashtags to geo-tag and organize important information, especially in the case of emergencies [33]. Note that the correlations detected are currently restricted by the use of hashtags as topics. As future research direction, we aim to investigate determining significant keywords in tweets and using them as topics as well.

The Value of Detecting Correlations: So far we have focused on correlations between locations and topics as a measure of trend significance. One simple way of studying geographical trends, however, is to employ *per-city analysis* to capture top topics in each location irrespective of the importance of these topics in other localities. *Per-city analysis* is easier to implement yet can contain noisy information, i.e. topics that are trendy in general and carry no geographical significance. In order to analyze how much such noise exists in Twitter, we performed various experiments in which we compared trends detected by *GeoWatch* to those detected by *per-city analysis*. The experiments were performed for various θ, ϕ and ψ settings which consistently gave similar results. Here we provide an overview of the results obtained in a particular setting where $\theta = 0.005$ and $\phi = 0.05$. In this case, the trends detected through the simple scheme are simply topics that were at least ϕ-frequent for any of the θ-frequent locations. This list of topics clearly contains *at least* as many location-topic pairs as *GeoWatch* which further filters this list using the parameter ψ to find correlations.

The trends detected through *per-city analysis* can potentially contain topics that are globally important and does not carry geo-intent.

In order to test the degree to which this happens, we compare the number of locations for a given topic that appear trendy in *GeoWatch* and in *location based top-k*, for $\psi = 0.05, 0.1$ and 0.2. The results show that the average of this value in the entire data set is between 1.4 to 1.7 times larger in *location based top-k* compared to *GeoWatch*. This indicates that by ignoring the value of correlations, *location based top-k* is not able to disentangle the connection between geographies and topics and therefore reports topics that are global trends (e.g. #ff) as local ones.

While a list of trends containing non-local topics can result in information overload, the degree to which this information overload affects comprehension is also contingent on the ordering at which the results are presented. For instance, if in the list of *location based top-k* results, topics with *real* geo-intent (i.e. geo-correlated location-topic pairs) are presented on the top of the list, the effect of information overload can be negligible. Therefore, next we study the performance of *location based top-k* results based on the ordering of the results. For this purpose, we compute the average precision of the set of pairs reported by *location based top-k*, ordered by the frequencies of the pairs, in capturing the real correlations as defined in Section 4.1. Here we present the results obtained by setting $\theta = 0.005$ and $\phi = 0.05$ and varying ψ between 0.05 and 0.2. The results show that the average precision quickly degrades from 0.51 to 0.2 as ψ increases. The small average precision shows that the real geo-intents captured by the correlations would be buried under a large amount of noise created by simple techniques such as detecting frequent topics per location.

5.2 The Accuracy of GeoWatch

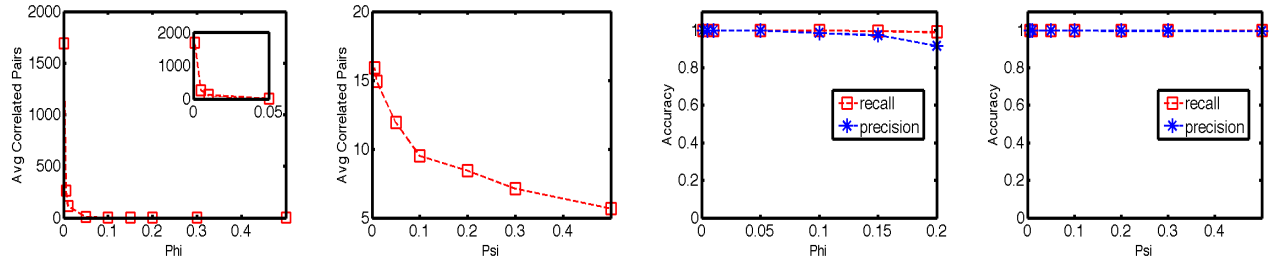
We first start by examining the number of correlated pairs detected with varying values of ϕ and ψ. As can be seen in Figure 4, increasing ϕ and ψ drastically decreases the number of correlated pairs. Evaluating the effect of changing ϕ, the other two parameters were set to $\theta = 0.005$ and $\psi = 0.05$, while varying ϕ between 0.005-to-1. Similarly, evaluating the effect of changing ψ, the other parameters were set as $\theta = 0.005$ and $\phi = 0.05$ while varying ψ between 0.005-to-1. The difference is more significant for small ϕ values, which indicates that it is less likely for the entire population to be interested in *only* one topic, while it is far more likely that there is *only* one (or few) location(s) that is interested in a given topic. Note that this artifact is somewhat created by design; the limitation of θ filters extremely inactive locations with few users whose interest can be extremely focused. These experiments provide a guide to the ***right choice of*** ϕ, ψ ***and*** θ ***values*** since one can make parameter choices based on the number of correlations that they aim to capture at a given time. However, we would like to point out that the proper settings for these values are dependent on the social network studied as well as the specific application. Therefore, our goal is to provide a general framework that can meet different needs rather than defining one set of parameter settings that is globally optimal.

Next we examine how varying the values of ϕ and ψ affects the recall and precision of *GeoWatch*. As stated in Theorem 4.4, *GeoWatch* is guaranteed to capture all the *trending* correlated location-topic pairs, where trendiness is defined based on a non-decreasing frequency function. In this section, we show two important findings: first, *GeoWatch* succeeds in capturing correlated location-topic pairs that do not necessarily follow this strict distribution and second, in addition to recall, *GeoWatch* 's precision is very high. As shown in Figures 4(c) and 4(d), *GeoWatch* has a perfect recall rate over various settings for ϕ and ψ values while the precision rate is slightly affected by increasing ϕ. The results provided in Figure 4(c) are obtained by setting the time window to 24 hours, $\theta = 0.005$ and $\psi = 0.05$ and varying ϕ. Similarly, the results provided in Figure 4(d) are obtained by setting the time window to 24 hours, $\theta = 0.005$ and $\phi = 0.05$ and varying ψ. Due to space limitations we omit the figures showing the behavior of *GeoWatch* with varying θ values. The analysis shows that the number of correlated pairs drops drastically with increasing θ.

5.3 Space and Time Efficiency of GeoWatch

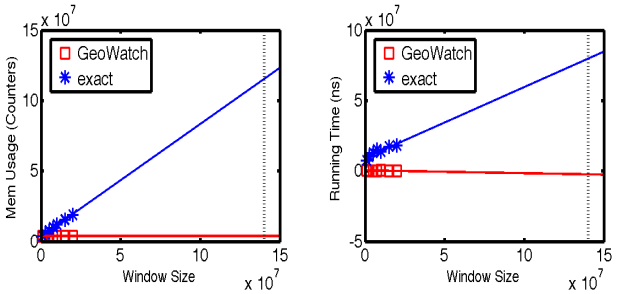
Space Efficiency of *GeoWatch*: In Figure 5(a), we provide a comparison between the exact solution and *GeoWatch*. The space comparison is based on the number of counters used by the two methods. For the exact solution this value would be equivalent to the number of unique elements while for *GeoWatch* it captures the number of elements maintained in the trending lists as well as the memory used for the sketches. Results provided in Figure 5(a) are based on the settings $\theta = 0.05$, $\phi = \psi = 0.1$ but we note that the general trend is similar for various other settings as well. *GeoWatch* provides means for defining the window size in terms of actual time or the number of elements to be maintained. For the purpose of this experiment, as our goal is to capture how well the algorithms scale, the window size is defined based on the number of elements. The recent numbers published by Twitter claim an average of 140 million tweets per day [36]. Therefore a geo-trend detection mechanism that is aimed to capture daily trends should process 140 million elements on average. We performed experiments setting the window size to 1, 2.5, 5, 7.5, 10, 15 and 20 million respectively and used linear regression to capture the memory usage when this number reaches 140 million. This point is marked by a dashed vertical line in Figure 5(a). Memory usage of the exact solution is comparable to *GeoWatch* for small window sizes. However, as the window size gets larger, memory requirements of the exact solution get larger while *GeoWatch* is unaffected.

Time Efficiency of *GeoWatch*: In satisfying Premises 3, 1 and 2, *GeoWatch* answers three types of queries at any particular time: reporting on frequencies of locations (Premise 3), frequency of topics (Premise 1) and reporting on correlated pairs (Premise 2). The efficiency of *GeoWatch* in answering queries relating to Premises 3 and 1 can be directly inferred from the results of heavy-hitters approaches and more specifiacally the sketch based method we use as a building block [20]. Due to space limitations, we omit such analysis and focus on the efficiency of *GeoWatch* in reporting correlated pairs. The three types of operations of interest are; *insert*, *remove* and *report*. In Figure 5(b), we present a similar analysis to the one presented for the space usage with identical settings for the parameters of the system ($\theta = 0.05, \phi = \psi = 0.1$), while we note that the results are similar for other settings. As the number of elements in the time window increases, the time required to report on the correlated pairs increases linearly for the exact solution while *GeoWatch* is not affected. Similar to Figure 5(a), we mark the 140 million point that corresponds to the average number of tweets per day. The results show that the exact solution does not scale. Also note that this linear fit is under the assumption of limitless memory. In reality as the number of elements increase in the given window, the memory required for the exact solution increases drastically. Implementing the exact solution in a real system with memory limits would result in thrashing which in turn increases run time drastically. Similar analysis was performed to test the efficiency of update methods. Unlike with the report method, these methods scale nicely with increasing window size for both exact solution and *GeoWatch*. As the window size increases, resulting in specific elements remaining in the lists for longer periods, update methods involve updating al-



(a) Effect of ϕ on # pairs (b) Effect of ψ on # pairs (c) Effect of ϕ on accuracy (d) Effect of ψ on accuracy

Figure 4: Effect of ϕ and ψ in the average number of correlated pairs detected by *GeoWatch* and accuracy measures



(a) Memory Usage (b) Report Running Time

Figure 5: Memory usage and Running Time comparison

ready existing structures more often than creating and destroying counters, resulting in such a performance. In general, our experiments show that update performance of *GeoWatch* is comparable to the exact solution, but for certain parameter settings the exact solution slightly outperforms *GeoWatch*. Note however, that this analysis is performed assuming unlimited memory. By increasing memory for the exact solution, update methods would also result in thrashing and consequently worse running time, while such an increase is not warranted for *GeoWatch*.

6. CONCLUSION

Geography plays an important role in our lives, shaping the friendships we form, and the interests we develop. The significance of geography in data analysis is clear since "...near things are more related than distant things" as the first law of geography states. Such significance incidentally also exists in the virtual extension of our daily lives; online social networks where users tend to befriend people and talk about events that are close-by. However, studying social networks through geo glasses goes well beyond a simple intellectual exercise. Recent events have shown that online social networks can be used in the case of a crisis to first *detect* the emergency event and later to deliver important information to interested users. Due to the large amount of noisy data shared on social networks, the *detection* of such significant *local* events becomes a non-trivial problem. Therefore, it is a critical task to provide large-scale data analysis tools that analyze social networks from a geographical perspective and detect such local events or interests in an online manner by also capturing the temporal aspects of information trends. This undertaking is the main focus of our study.

To this end, in this work we studied the online detection of geo-correlated information trends, i.e. identifying correlated location-topic pairs along a sliding window in a social data stream. We showed that the exact solution for such a problem requires keeping track of all possible pairs of location-topic pairs which is infeasible due to the large scale of data. Therefore, we introduced *GeoWatch*: an approximate solution that requires only sub-linear memory and running time while guaranteeing to capture all *trending* correlations. We experimentally studied the value, accuracy and efficiency of *GeoWatch* in Twitter and showed that this tool provides a manageable list of interesting location-topic pairs including crisis events such as earthquakes, or local events such as political demonstrations, concerts or sports events. The experiments show that *GeoWatch* scales well with increasing amount of data while the exact solution suffers from such an increase. In addition, the experiments show that, in addition to perfect recall measures, *GeoWatch* also has a high precision.

Even though in our experiments we apply *GeoWatch* to detect trends in Twitter, the tool is generic enough to be used in other social networks as well. Similarly, the topics, as defined based on hashtags in this study, or locations, defined based on cities, can be redefined. In fact, topic detection of information items shared in social networks is an important open problem which can reshape how a topic is to be defined in *GeoWatch*. Similarly, locations of interests can be regions, countries or simply arbitrary polygons on a map. *GeoWatch* can easily be used to detect geo-trends in all those resolutions. An important future work in this context is to detect hierarchical geo-trends by capturing the right resolution in which a topic is trending in an online manner. Although multiple *GeoWatch* structures can be used in parallel to address this problem, our future goal is to investigate if there are more compact ways in which hierarchical geo-trend detection can be performed.

7. REFERENCES

[1] B. Adams and K. Janowicz. On the geo-indicativeness of non-georeferenced text. *ICWSM-12*, 2012.

[2] J. Allan, editor. *Topic detection and tracking: event-based information organization*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[3] L. Backstrom, E. Sun, and C. Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 61–70, New York, NY, USA, 2010. ACM.

[4] C. Budak, D. Agrawal, and A. El Abbadi. Structural trend analysis for online social networks. *Proc. VLDB Endow.*, 4:646–656, July 2011.

[5] H. Cao, G. Hripcsak, and M. Markatou. A statistical methodology for analyzing co-occurrence data from a large

sample. *J. of Biomedical Informatics*, 40(3):343–352, June 2007.

[6] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent elements in data streams. In *ICALP'02*, pages 693–703, 2002.

[7] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *CIKM '10*, pages 759–768. ACM, 2010.

[8] G. Cormode and M. Hadjieleftheriou. Finding the frequent items in streams of data. *Commun. ACM*, 52:97–105, October 2009.

[9] G. Cormode and S. Muthukrishnan. What's Hot and What's Not: Tracking Most Frequent Items Dynamically. *TODS'05*, 30(1):249–278, 2005.

[10] A. Dalli. System for spatio-temporal analysis of online news and blogs. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 929–930, New York, NY, USA, 2006. ACM.

[11] E. D. Demaine, A. López-Ortiz, and J. I. Munro. Frequency estimation of internet packet streams with limited space. In *ESA'02*, volume 2461, pages 348–360, 2002.

[12] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing. A latent variable model for geographic lexical variation. In *EMNLP '10*, pages 1277–1287, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[13] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, 165(1):91–134, June 2005.

[14] M. Gardner and D. Altman. *Statistics with confidence: confidence intervals and statistical guidelines; Statistics with confidence: confidence intervals and statistical guidelines*. Brithis Medical Journal, 1995.

[15] S. Havre, B. Hetzler, and L. Nowell. ThemeRiver: visualizing theme changes over time. In *InfoVis 2000*, pages 115–123, 2000.

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

[17] Indonesia, brazil and venezuela lead global surge in twitter usage. `http://www.comscore.com/Press_Events/Press_Releases/2010/8/Indonesia_Brazil_and_Venezuela_Lead_Global_Surge_in_Twitter_Usage`.

[18] Tweetstats. `http://tweetstats.com/trends`.

[19] Trendsmap. `http://trendsmap.com/`.

[20] C. Jin, W. Qian, C. Sha, J. X. Yu, and A. Zhou. Dynamically maintaining frequent items over a data stream. In *CIKM '03*, pages 287–294. ACM, 2003.

[21] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *WWW '10*, pages 591–600, 2010.

[22] T. Lappas, M. R. Vieira, D. Gunopulos, and V. J. Tsotras. On the spatiotemporal burstiness of terms. *Proc. VLDB Endow.*, 5(9):836–847, May 2012.

[23] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD '09*, pages 497–506, 2009.

[24] A. M. MacEachren, A. C. Robinson, A. Jaiswal, S. Pezanov, A. Savelyev, J. Blanford, and P. Mitra. Geo-Twitter analytics: Application in crisis management. In *25th International Cartographic Conference*, July 2011.

[25] G. S. Manku and R. Motwani. Approximate frequency counts over data streams. In *VLDB'02*, pages 346–357, 2002.

[26] Maxmind world cities with population. `http://www.maxmind.com/app/worldcities`.

[27] Y. Meng and M. H. Dunham. Efficient mining of emerging events in a dynamic spatiotemporal environment. In *Proceedings of the 10th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, PAKDD'06, pages 750–754, Berlin, Heidelberg, 2006. Springer-Verlag.

[28] A. Metwally, D. Agrawal, and A. El Abbadi. An integrated efficient solution for computing frequent and top-k elements in data streams. *TODS'06*, 31(3):1095–1133, 2006.

[29] A. Metwally, F. Emekçi, D. Agrawal, and A. El Abbadi. Sleuth: Single-publisher attack detection using correlation hunting. *Proc. VLDB Endow.*, 1(2):1217–1228, Aug. 2008.

[30] B. Poblete, R. Garcia, M. Mendoza, and A. Jaimes. Do all birds tweet the same?: characterizing twitter around the world. In *CIKM '11*, pages 1025–1030. ACM, 2011.

[31] D. M. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *WWW '11*, pages 695–704. ACM, 2011.

[32] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling. Twitterstand: news in tweets. In *GIS '09*, pages 42–51, 2009.

[33] K. Starbird and L. Palen. "voluntweeters": self-organizing by digital volunteers in times of crisis. In *CHI*, pages 1071–1080, 2011.

[34] W. Tobler. A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46(2):234–240, 1970.

[35] I. Tsoukatos and D. Gunopulos. Efficient mining of spatiotemporal patterns. In *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases*, SSTD '01, pages 425–442, London, UK, UK, 2001. Springer-Verlag.

[36] Twitter blog: #numbers. `http://blog.twitter.com/2011/03/numbers.html`.

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

[38] B. Wing and J. Baldridge. Simple supervised document geolocation with geodesic grids. In *ACL*, 2011.

[39] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 1029–1038, New York, NY, USA, 2010. ACM.

[40] Y. Zheng. Tutorial on location-based social networks. In *Proceedings of the 21st international conference on World wide web*, WWW '12, 2012.