University of California
Santa Barbara

Understanding the Real World through the Analysis of User Behavior and Topics in Online Social Media

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Computer Science

by

Theodore Georgiou

Committee in charge:

Professor Amr El Abbadi, Co-Chair
Professor Xifeng Yan, Co-Chair
Professor Divyakant Agrawal

March 2017

The Dissertation of Theodore Georgiou is approved.

Professor Divyakant Agrawal

Professor Amr El Abbadi, Committee Co-Chair

Professor Xifeng Yan, Committee Co-Chair

March 2017

Understanding the Real World through the Analysis of User Behavior and Topics in Online Social Media

Theodore Georgiou

To my parents, who raised me to love and appreciate science.

Acknowledgements

This work would not have been possible without the continuous support of many people.

I offer my deepest gratitude to my advisors Professor Amr El Abbadi and Professor Xifeng Yan. Their advice and feedback was invaluable and greatly helped in shaping the content of the current dissertation (and getting our papers published!).

I extend my appreciation to my lab-mates, Ceren, Cetin, Faisal, Vaibhav, Victor, Xiaofei and Aaron. Apart from the occasional chance to brainstorm with them, they also provided a pleasant environment to socialize and have fun! I also want to thank the professors at UCSB that I had the pleasure to collaborate with, Divy Agrawal, the only person in the US that calls me Thodoris, Jianwen Su, who was the first to introduce me to research at UCSB, and the Communication professors Miriam Metzer and Scott Reid, who offered their valuable knowledge for our privacy projects and grant proposal.

Finally, I would like to thank Meni and my family for their ever-soothing presence and support in my life, and my dear friend and roommate Stratos who made living in Santa Barbara feel a bit more like home.

Curriculum Vitæ
Theodore Georgiou

Education

2017	Ph.D. in Computer Science (Expected), UC Santa Barbara.
2016	M.Sc. in Computer Science, UC Santa Barbara.
2010	B.Sc. in Informatics and Telecommunications, University of Athens.

Professional Experience

(2017-) Software Engineer, Google (Expected)
(2016) Research Intern, IBM Research (T.J. Watson)
(2012-2013) Software Engineer Intern, Twitter Inc.

Awards

Outstanding Teaching Assistant Awards: Department of Computer Science (2011, 2012, 2016), Graduate Student Association (2016), and College of Engineering (2016) at UC Santa Barbara

Publications

- Theodore Georgiou, Amr El Abbadi, Xifeng Yan, "Extracting Topics with Focused Communities for Social Content Recommendation", Conference on Computer-Supported Cooperative Work and Social Computing (CSCW 2017)
- Theodore Georgiou, Amr El Abbadi, Xifeng Yan, "Privacy Cyborg: Towards Protecting the Privacy of Social Media Users", International Conference on Data Engineering (ICDE 2017)
- Theodore Georgiou, Amr El Abbadi, Xifeng Yan, "Mining Complaints for Traffic-Jam Estimation: A Social Sensor Application", International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2015)
- Ceren Budak, Theodore Georgiou, Divyakant Agrawal, Amr El Abbadi, "GeoScope: Online Detection of Geo-Correlated Information Trends in Social Networks", International Conference on Very Large Data Bases (VLDB 2014)
- Divyakant Agrawal, Amr El Abbadi, Vaibhav Arora, Ceren Budak, Theodore Georgiou, Hatem A Mahmoud, Faisal Nawab, Cetin Sahin, Shiyuan Wang ,"Mind your Ps and Vs: A perspective on the

challenges of big data management and privacy concerns", International Conference on Big Data and Smart Computing (BigComp 2015)

- Divyakant Agrawal, Ceren Budak, Amr El Abbadi, Theodore Georgiou, Xifeng Yan, "Big data in online social networks: user interaction analysis to model user behavior in social networks", International Workshop on Databases in Networked Information Systems (DNIS 2014)

Abstract

Understanding the Real World through the Analysis of User Behavior and Topics in Online Social Media

by

Theodore Georgiou

Physical events happening in the real world usually trigger reactions and discussions in the digital world; a world most often represented by Online Social Media such as Twitter or Facebook. Mining these reactions through social sensors offers a fast and low cost way to explain what is happening in the physical world. A thorough understanding of these discussions and the context behind them has become critical for many applications like business or political analysis. This context includes the characteristics of the population participating in a discussion, or when it is being discussed, or why. As an example, we demonstrate how the time of the day affects the prediction of traffic on highways through the analysis of social media content. Obtaining an understanding of what is happening online and the ramifications on the real world can be enabled through the automatic summarization of Social Media. Trending topics are offered as a high level content recommendation system where users are suggested to view related content if they deem the displayed topics interesting. However, identifying the characteristics of the users focused on each topic can boost the importance even for topics that might not be popular or bursty. We define a way to characterize groups of users that are focused in such topics and propose an efficient and accurate algorithm to extract such communities. Through qualitative and quantitative experimentation we observe that topics with a strong community focus are interesting and more likely to catch the attention of users.

Consequently, as trending topic extraction algorithms become more sophisticated and

report additional information like the characteristics of the users that participate in a trend, significant and novel privacy issues arise. We introduce a statistical attack to infer sensitive attribute values of Online Social Networks users that utilizes such reported community-aware trending topics. Additionally, we provide an algorithmic methodology that alters an existing community-aware trending topic algorithm so that it can preserve the privacy of the involved users while still reporting trending topics with a satisfactory level of utility. From the users perspective, we explore the idea of a cyborg that can constantly monitor its owners privacy and alert them when necessary. However, apart from individuals, the notion of privacy can also extend to a group of people (or community). We study how non-private behavior of individuals can lead to exposure of the identity of a larger group. This exposure poses certain dangers, like online harassment targeted to the members of a group, potential physical attacks, group identity shift, etc. We discuss how this new privacy notion can be modeled and identify a set of core challenges and potential solutions.

Contents

Chapter 1

Introduction

Since the establishment of online social media, real life events frequently trigger a social reaction on the web. This has led to an era where Big Data and social media content are strongly tied together [1]. Utilizing this vast, but publicly available, amount of information to mine the correlation between physical events and postings on Twitter or Facebook has proven to unveil hidden behavioral patterns or validate social and psychological theories that once required extensive and expensive surveys [2]. Additionally, the discovery of what is happening in the real world is now feasible through purely automated and algorithmic tools that only require access to the Internet. The study of social patterns in Online Social Media like Twitter or Facebook can be very helpful in identifying collective user behavior among specific segments of society. Towards this goal, *Trending Topics* have been popularly used in the detection of breaking news, as well as in marketing and advertising mechanisms.

1.1 Going Beyond Trending Topics

Currently, users of popular social media services like Twitter and Facebook use the real-time list of trending topics provided by each service to get a glimpse of what users outside their social circle are talking about, discover major events happening around them or far away, monitor breaking news, or get a measure of how popular a social movement is. Both Twitter and Facebook are putting a significant effort in delivering topics that are relevant and could lead to high engagement between their users and the posted content.

Trend analysis has its foundations in the problem of identifying *heavy hitters* or *top-k* in one-pass algorithms on data streams [3, 4]. Existing algorithms to extract trends from websites like Twitter or Facebook are quite simplistic and hence do not pose any privacy dangers. The earliest approaches to analyze trends in social media introduced the so-called *trending topics*. As the term hints, these topics are keywords, phrases, or hashtags with bursty and popular behavior. As an example, during the Senate Elections on November 4, 2014 the topics "ivoted" and "#senate" were trending on Twitter and were terms that hundreds of thousands of users mentioned in their tweets. We can consider trending topics as a single-dimensional trend analysis where Topic is the only dimension.

Usually, the origin of a trending topic is a popular real life event that is being discussed on social media or a meme that is spreading. Trending topics are used to understand and explain how information and memes diffuse through vast social networks with hundreds of millions of nodes. And due to their nature, trending topics are useful when reported in real time to reflect current events. Because of this requirement, methods that identify and extract trending topics need to be scalable and process data in a streaming fashion as efficiently as possible. In Figure 1.1 3 examples of trending topic reports are shown taken

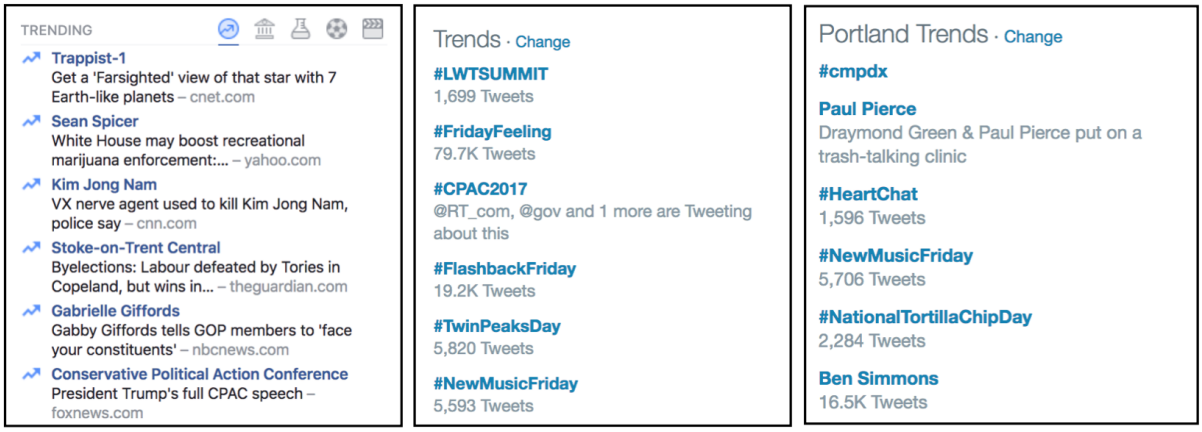


Figure 1.1: 3 examples of different trending topic lists. From left to right: Facebook, Twitter (Tailored), Twitter (by location)

from real Social Media websites at the same time. By observing that these reports have no overlap, it is obvious that different algorithmic approaches can result to completely different results.

The relevance of a topic to the user's interests, plays an important role in the success of such engagement. It has been observed that the user population involved in a trend offers high potential in understanding the trend and how other users might react to it. In a previous study on Twitter topics even simple social relations between the participants could greatly enhance the understanding of trending topics [5] or spammer detection [6]. Alternatively, we proposed a space-efficient framework [7], that extracts topics which are highly focused in specific geographical locations. Human evaluations showed that topics with a high geographical correlation tend to be more interesting than topics with a dispersed population.

In this Dissertation we propose a novel community detection algorithm that utilizes a spectrum of social characteristics rather than just geographic locations. The detection of community characteristics that are meaningfully correlated with a topic, like gender, age, location, race, ethnicity, political affiliation, etc., can yield powerful results which

are useful in a variety of domains. Marketers can understand their customers better by identifying the communities interested in their products. Advertisements, which usually are linked to a trending topic or event, can become more personalized. And of course, content recommendation can be improved through the extraction of target groups interested in specific topics. The framework scales linearly with the number of attributes, and reports communities that share a set of attribute combinations or sub-dimensions.

However, due to the open-access nature of Online Social Networks like Twitter, where everyone can see who says what, and depending on how much information a trending topic contains, novel notions of privacy emerge. As a concrete example, Twitter reports trending topics by location, even at the city resolution. Their service also offers a search functionality which enables the discovery of all social postings (tweets) that contain certain keywords, and those tweets are always associated with a user of the social media service. When Twitter reports that a topic is trending in Athens, Greece, anyone can find the users that mentioned this topic through Search and may, therefore, assume that they live in Athens, Greece. The location of a user could be considered a sensitive attribute, if for example they post provocative political opinions and are afraid of physical repercussions. As we will show later, an attacker can easily infer the location of hundred of thousands of Twitter users through a simple crawling of Location-based trending topics using the official Twitter API. These users do not geocode their tweets neither publicly display their location on their profile. Thus, the correlation between trending topics and attributes like location can lead to privacy leaks. Building smarter trending topic extraction algorithms, which contain richer demographic information of the involved users can further increase the privacy risk of any reported topic. It is important that any algorithm that extracts multiple correlated user attributes takes privacy seriously into account.

The public nature of Online Social Networks, like Twitter and Facebook, has intro-

duced a different privacy danger from the more traditional linkage attack (identifying the real identity of an online user). Attribute inference, the process of inferring an OSN user's attributes like age, gender, location, race, political preference, etc., can be extremely useful for the purposes of personalization in content recommendation, advertising, and/or social media analytics. For example, large Social Media websites like Facebook and Twitter already have proprietary methods for inferring social attributes of their users that are not explicitly provided by them. Recently, it was discovered that Facebook is able to learn a user's political preference between values like "Liberal", "Moderate", or "Conservative". However, if a third-party attacker is capable of inferring attributes that are sensitive or private then it is important to build techniques that can protect OSN users. For *example*, if it is reported that people that mentioned topic $\#BlackLivesMatter$ are 79% teenagers, 86% African Americans, and 67% live in Chicago, then an attacker can infer the age, race, and location of any user that mentions this hashtag with some statistical confidence. Thus, In the presence of even more sophisticated trending algorithms that capture several attributes apart from location, reports of trending topics further enable attribute inference attacks. On the other hand, in the presence of such a reporting system, users must be mindful of which topics they discuss in order to protect themselves from such inference attacks. This can be particularly tedious and time consuming given the nature of social media which promotes public and frequent posting, something that usually seems harmless when considered at the level of a single post. Towards this end, we built a privacy cyborg, that can undertake the task of monitoring its owner's posts in social media and automatically warn them if necessary.

Finally, we study how the context of specific topics, like who is posting on Social Media or when they are posting can affect the quality of a data mining or machine learning product that summarizes or predicts real life events through online social media (social sensors). Specifically, we study the problem of traffic-jam estimation and show

that knowing if a post is coming from a driver or not and which time of the day a post was made can significantly improve the accuracy of the traffic estimation.

1.2 Research Contributions

Through the studies and experiments performed throughout the duration of this Dissertation, we have made the following contributions:

- The introduction and definition of focused communities in Social Media. [8]
- Provide a scalable algorithm for the discovery of maximally focused communities with amortized linear time complexity. [8]
- Demonstrate the effectiveness of recommending topics with focused communities through human evaluation. [8]
- The introduction of a novel privacy attack model using sensitive attribute inference in the context of community-aware trending topic reporting.
- Provide an algorithmic methodology that identifies when a user's privacy is in danger of compromise, and preserves it by anonymizing the community characteristics of the reported trending topics. There are many ways to anonymize these characteristics, with different levels of utility loss, but our methodology aims to minimize this loss in an efficient manner.
- Build a system (cyborg) that can monitor an individual's privacy in real time and provide warnings when a sensitive attribute can be successfully inferred by an attacker that has access to community-aware trending topic reports. [9]
- A novel regression model for traffic-severity estimation based solely on the generated social volume. The proposed model exploits the fact that people complain in

different levels throughout the day and can be used to estimate traffic congestion in areas that lack proper traffic monitoring resources. The analysis is applied on a major Californian freeway (I-405) and spans across 6 months of data. [10]

- A better understanding of human behavior when it comes to drivers and their social media actions while behind the wheel. [10]

- Offer some initial vision on the concept of Group Privacy which would generalize the concept of an individual's privacy in Social Media to a whole community. This project is funded by NSF grant CNS 1649469.

1.3 Dissertation Organization

In Chapter 2 the definition of a focused community is given. Most of the work presented in the current Dissertation will be referring to the notion of focused communities. Specifically, in Chapter 3 the description of a scalable algorithm that can extract focused communities for topics discussed in Social Media is given. Then, in Chapter 4 we delve into the privacy concerns that can be raised in the presence of an algorithm that reports focused communities. This new privacy challenge can be approached by both sides: (a) The algorithmic perspective where the algorithm itself takes care of privacy issues by obfuscating results before publishing. (b) The user's perspective, where each individual personally undertakes the task of protecting themselves, but still with the assistance of technology. In Chapter 5 we demonstrate how important the context (who and when) is important in data mining applications that aim to understand what is happening in the real world through social media content. Finally, the Dissertation concludes in Chapter 6 with future plans on extending the privacy challenges discussed in Chapter 4 from the individual's level to the community (or group) level.

Chapter 2

Focused Communities

We start by defining the concept of a *focused community*. This definition will let us exploit specific properties in Chapter 3 to propose a novel framework that receives a social stream as its input and efficiently extracts and reports topics with the corresponding focused communities. Furthermore, in Chapters 4 and 6 we will refer back to this definition to identify privacy challenges and solutions.

Communities focused on topics, can sometimes be expected and sometimes unexpected. It is easy to anticipate that young boys will be interested in the PlayStation 4 gaming console even without monitoring the widely popular topic #PS4. But we might not expect that women in the area of Boston, MA, that also support the Democratic party, showed their solidarity to an arrested female teen named Justina with the not so popular topic #FreeJustina. It is even more unexpected to observe the hijacking of the hashtag campaign #ReasonsToVisitEgypt that was originally created to promote tourism in Egypt, but local citizens used it negatively to raise awareness for the country's political situation. The important take away is that using only the popularity or bursty behavior of a topic is usually not enough; a better understanding of the underlying community can yield a better ranking for interesting topics that might not be globally popular.

2.1 Definition

Focused communities are groups of social media users that have a focus on a specific topic and might not be related otherwise. The set of users belonging in a focused community share two properties: they all mentioned the same topic and they all share *some* characteristics. In order to extract and understand the underlying communities interested in a particular topic, T, two pieces of information are necessary. 1) The topic population P which includes every social posting that mentions topic T. We will refer to these social postings using the general term *datapoints* but in the specific case of Twitter they are called *tweets*. 2) The corresponding social characteristics (attribute values) for every datapoint. These attributes can include user demographics like Location, Age, Gender, Race, or characteristics like political affiliation, supporting soccer team, hobbies, etc. Each user that mentions a topic can be represented by an attribute vector. For example, a hypothetical 5-dimensional attribute vector could be: [Location: Los Angeles, Age: 18, Gender: Male, Citizenship: USA, Political Affiliation: Republican]. Certain attributes can be hierarchical, like Location or Age. If a user lives in Los Angeles, then she also lives in California, or USA, or the World. If a user is 15 years old then she also belongs in the "teenager" age bracket. Ultimately, given the population of a topic T, we want to extract a combination of attribute values in order to discover the "maximally focused community" interested in topic T. Note that the process of identifying a (maximally) focused community has to be applied individually on each topic's user population and not the whole stream of social postings or the whole user base. We will now formally define focused and maximally focused communities.

Suppose a domain with N total attributes where each attribute a_i has a finite set of values V_{a_i}. Categorical attribute values may follow a tree-like hierarchical pattern. As the most notable example, the Location attribute can be described using a tree hierarchy

of 4 levels: city, region/state/province, country, and "Worldwide". Values in each level of the hierarchy are connected to a single ancestor from the previous level and to an arbitrary number of successors in the next level (which can be zero for the values of the bottom level). We symbolize the root of the hierarchy with the value "*". Note that any attribute can be described at the very least by the trivial hierarchy of 2 levels where the bottom level contains all the values and the top level contains the root. Numerical attribute values can be viewed as hierarchical attributes as well. Using a radius r the hierarchical ancestor of a numerical value v can be dynamically estimated as the range $[v - r, v + r]$. Alternatively, the values of a numerical attribute can be discretized so it becomes categorical. In the current work we focused on categorical attributes but the proposed algorithm works also with numerical attributes.

Let P be a set of datapoints where each datapoint is represented by a vector of N attribute values $v_i \in V_{a_i}$. For simplicity, we will refer to these attribute vectors as *tuples*; therefore, any datapoint is considered a tuple which is practically a combination of attribute values. The *support* of a single attribute value is equal to the number of datapoints in P that contain this value. The *support* of a tuple is equal to the number of datapoints with values that match the values of this tuple.

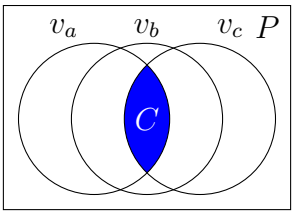
A combination of attribute values (tuple) describes all the users that match these values and can be visualized as the intersection of the N groups of users that match each individual attribute value (Figure 2.1a). These users are not necessarily connected in the social graph but instead connected through the fact that they all mentioned the same topic T. We refer to such groups of users as topic-based *communities*, or simply just communities, and represent them through the described notion of tuples. However, in any given topic population there is a vast amount or arbitrary attribute intersections that are mostly meaningless. In order to capture important communities we explore the notion of *focus*. The presence of focus dictates that there is *at least one attribute*

of the community (possibly more) that is not present to anyone else outside the topic community. This leads to communities that are not random intersections and is captured by the following definition of *focused communities*.

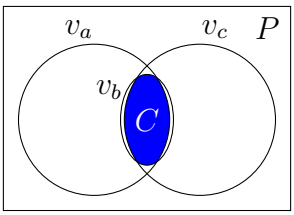
Let C be a group of users that all share a combination of common attributes represented by the tuple C_t. This group C is a *focused community* if there is at least one attribute value v in the tuple C_t that represents the community C which no other user in the complement $P - C$ matches. This attribute value v is practically an exclusive feature of the community. Again, while there is at least one attribute necessary to form a focused community there can be multiple exclusive attributes. To capture this difference, we will further introduce the notion of *maximally focused communities*.

Figure 2.1 illustrates the difference between an arbitrary community (non focused) and a *focused* community with three attributes. As an example, we can assume that attribute a is Location with value v_a equal to Los Angeles, attribute b is Age with value v_b equal to 18 years old, and attribute c is Gender with value v_c equal to Male. In the first case, the population corresponding to the intersection of the three attributes defines a non focused community, ie., 18 year old males who live in Los Angeles. In the second case, the population corresponding to the attribute $v_c \equiv 18$ years old is almost identical to the intersection of all three attributes. Therefore, the support of the Los Angeles male community is almost equal to the number of users in P that are 18 years old, since almost nobody else in the complement $P - C$ matches this age.

We also establish the mathematical formulation of the focus requirement which will be used in the proposed algorithm that identifies focused communities within all the users that mention a particular topic T: Let $P_v \subseteq P$ be the set of all users in the topic population P that match a single attribute value v. The following must hold for a community C to be *focused*: $\exists v \in C_t$ so that $P_v \equiv C$. In order to discover focused communities in the presence of data noise or missing values this formula needs to be



(a) Non focused community



(b) Focused community ($\epsilon > 0$)

Figure 2.1: Illustration of a non focused community (a), which is the simple intersection of three attribute values v_a, v_b, and v_c. A focused community (b) has at least one attribute (v_b) that is as close to the intersection of v_a, v_b, and v_c.

relaxed by introducing a relaxation threshold ϵ so that we can measure how close a community is to being perfectly focused:

$$\left| \frac{|C|}{|P_v|} - 1 \right| \leq \epsilon \tag{2.1}$$

When the attribute value v is absolutely exclusive to the community C the left-hand side of the equation will be exactly equal to 0. When the exclusive attribute "leaks" outside the community C then the value will become greater than 0. We will refer to this value as the *focus metric* of the community. A value of 0 indicates that the community is perfectly focused. A value above ϵ indicates that it is not focused.

Because a focused community can have multiple exclusive attribute values, we now introduce the notion of maximality. A *maximally focused community* is a focused community that cannot become larger by introducing a new or different attribute value without losing its focus property (Equation (2.1)). Note that a topic population might contain multiple maximally focused communities which are guaranteed to not overlap, based on

the focus property (or might overlap slightly depending on the relaxation value of ϵ).

2.2 Attribute Generalization

Since the attributes values are hierarchical, as described above, a value v can be generalized to a direct ancestor of v in the hierarchy. Though generalization we can reach focused communities that were not possible as a combination of base values. The generalization of any value except "*" is possible; the root value "*" cannot be generalized since it has no ancestors. We denote the case of a missing attribute value using the "$\perp$" operator (bottom). A "$\perp$" value can be directly generalized to "*" through a single generalization step no matter how high the attribute hierarchy is. In the general case, an attribute a can be generalized from value v_a to value v_b if v_b *precedes* or is equal to v_a in attribute a's hierarchy. We denote this relation between v_a and v_b using the operators $\succeq$ (succeeds) and $\preceq$ (precedes): $v_b \preceq v_a$ or $v_a \succeq v_b$. As an example, for the Location attribute the following relations are true: Los Angeles $\succeq$ Los Angeles, Los Angeles $\succeq$ California, Los Angeles $\succeq$ USA, California $\succeq$ *, etc.

The support of a generalized attribute value in P is equal to the number of datapoints that contain any successor of the value. For example, in a two-dimensional space, the tuple [Location:California, Gender:*] matches datapoints like [Los Angeles, Male] or [San Francisco, Female]. The tuple that contains all the hierarchy roots is called *HEAD*: $HEAD \equiv [*, *, ..., *, ..., *]$. The *HEAD* tuple matches every datapoint in P: $\left|HEAD\right| = \left|P\right|$. Figure 2.2 shows an example of the formed lattice given a specific starting tuple with three attributes: Location, Gender, and Age. Connected nodes are reachable through a series of attribute value generalizations (*climbing*).

Since every single tuple with unique attribute values is a potentially self-contained focused community, we further require a focused community to meet a minimum support

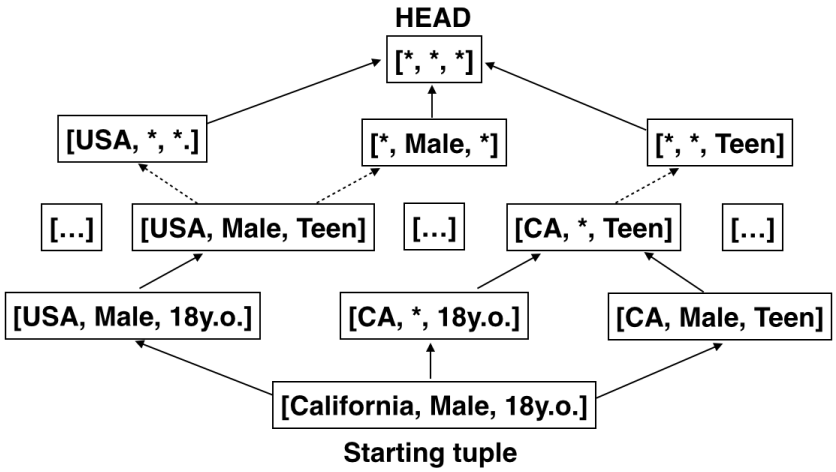


Figure 2.2: Partial view of the attribute lattice. Two connected nodes (solid arrow) in the lattice indicate that a tuple can be reached from the other through a single attribute generalization. A dashed arrow indicates that two nodes have other nodes between which are omitted due to space restrictions.

requirement, relative to the population P. More specifically, we introduce a support threshold $\xi \leq 1$ so that every *maximally focused community* has support of at least $\xi|P|$.

Focused communities are groups of people that share common characteristics without being necessarily connected through the social graph. The way such communities form is through the mention of topics on social media and are data-driven and individual members might not be aware of their membership to the community. Through the notion of focused communities we are able to identify groups of people that have a focused interest on potentially unexpected topics. We utilize this focused interest to extract trending topics that can be interesting to an even larger and more general population (Chapter 3). Additionally, we explore how this extraction introduces privacy concerns since it involves the knowledge and exposure of private user attributes (Chapter 4).

Chapter 3

Community-Aware Trending Topics

In this chapter, we describe a novel algorithm for the extraction of Focused Communities in real-time from a stream of Social Media posts.

3.1 Extracting Focused Communities

The proposed algorithm aims to extract the (maximally) focused communities for any topic: Given a topic T, extract all the maximally focused communities with a focus metric less or equal to ϵ and support greater or equal to ξ. The output of the algorithm is one or more tuples that define maximally focused community through a combination of attribute values. We first provide a basic overview of the algorithm, then discuss its two phases (sampling and climbing), show its efficiency and accuracy based on synthetic data, and finally, offer a way to deal with missing values in real datasets.

3.1.1 Overview of the Sample&Climb Algorithm

The algorithm can be applied on the set of datapoints that mention a topic T (for example, all the tweets that mention the hashtag #ObamaInThreeWords). This set of

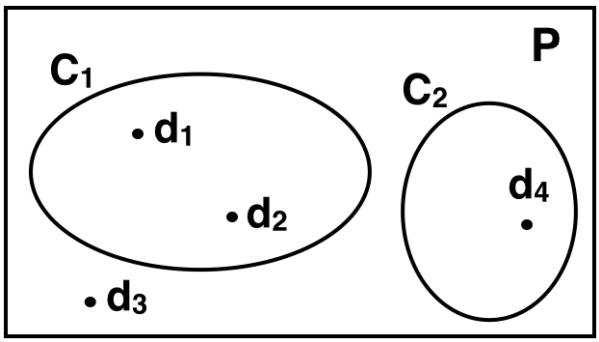


Figure 3.1: Sampling phase example.

datapoints is referred to as topic population P. To extract focused communities for other topics the algorithm needs to be applied separately to the corresponding sets of datapoints. Grouping the whole stream of datapoints into separate topic populations is a simple pre-processing step which will be discussed later. In this section we will assume and describe a single instance of the algorithm for a single topic. The extraction of a maximally focused community is an optimization problem: find a combination of attribute values (tuple C_t) that maximizes the size of the community defined by C_t, while minimizing the focus metric (Equation (2.1)). The Sample&Climb algorithm, named by its two phases, initially selects a random sample of datapoints from P (sampling phase) and uses each datapoint as a starting point to reach the attribute values of a focused community through a series of value generalizations (climbing phase). As a real example, the Twitter hashtag "#ObamaInThreeWords" was found to have a single maximally focused community that includes supporters of the Republican Party (Political affiliation), that are Male (Gender), between the ages 19-22 (Age), and that live in the United States (Location). In the following subsections we describe each phase of the algorithm in detail.

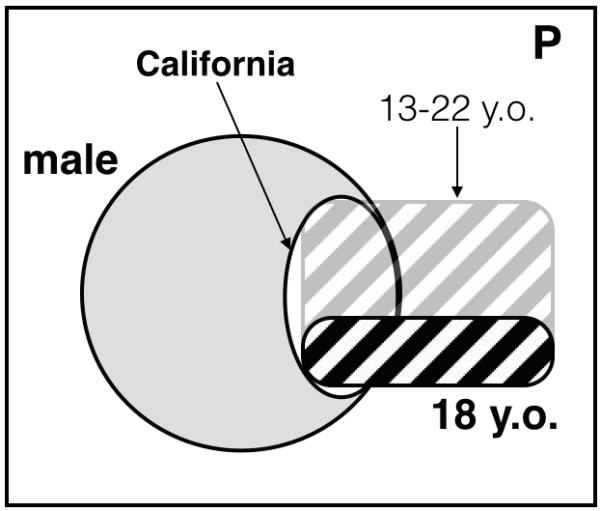


Figure 3.2: An example case where the greedy attribute selection policy can fail to select the best attribute.

3.1.2 Sampling Phase

The sampling phase must efficiently bootstrap the optimization problem of extracting a tuple that defines a topic's maximally focused community. The main goal is to avoid enumerating all possible attribute combination which would be exponentially expensive and instead seed the process with base combinations that are already observed in single datapoints. To that end, we *uniformly* sample k tuples from P (datapoints) and create a new set S; every tuple $t \in S$ is then fed to the climbing phase which will reach a potential maximally focused community. If the sampled tuple is actually a member of a maximally focused community (checked by Equation 2.1), the climbing phase should extract the community. If the sampled tuple is not a member of any focused community the climbing phase will not extract a community. The intuition behind this approach is to probabilistically select datapoints that might belong to a maximally focused community. This intuition is visualized in Figure 3.1 where we assume that in a population P two *focused* communities C_1 and C_2 exist. The sampling of datapoints d_1 or d_2 can enable the extraction of community C_1. The sampling of datapoint d_4 can enable the extraction

of community C_2. The sampling of datapoint d_3 does not enable the extraction of any community and a different datapoint needs to be sampled. If the datapoint is indeed a member of a community, then a series of attribute generalizations and focus metric computations can lead us to the actual attribute values of the community. For example, if the following focused community exists: [Location: USA, Gender: *, Age: 13-18] and the datapoint: [Santa Barbara, Male, 18] is randomly selected then the location value can be generalized twice (Santa Barbara $\rightarrow$ California $\rightarrow$ USA), the gender value once (Male $\rightarrow$ *), and the age value once (18 $\rightarrow$ 18-23) to reach the community.

When a sampled tuple successfully leads to the extraction of a maximally focused community, the result is saved. If the next sampled tuple succeeds an already extracted community, by a previous iteration in the sampling phase, then the tuple is skipped since it can only lead to a known community and would be a waste of resources to process it. Pseudocode for the *sampling* phase is provided in Algorithm 1. Line 5 tests if the new sampled tuple succeeds an already extracted community c. If the tuple is already a successor of an extracted community, the climbing phase is skipped since it will yield the same result given that the climbing process is deterministic. The returned result of the climbing phase is a maximally focused community if climbing was successful, or *NULL* if a focused community could not be extracted (line 8).

Based on the desired success probability of the sampling phase p_b, the appropriate minimum size of the sample S can be determined. Let k be the number of sampled datapoints and C a unique maximally focused community in P.

The sampling of datapoints can be simulated through a series of Bernoulli trials where success is defined as the selection of a datapoint tuple t so that $t \succeq C_t$. The number of trials is equal to the size of the sample: $\big|S\big| = k$. The probability of success in a single trial is equal to $p = \big|C\big|/\big|P\big|$. The probability of *at least one success out of k trials* (we can assume that P is large enough for the trials to be independent even

Algorithm 1: Sampling phase

```
Data: Tuples P, attribute hierarchies H[N]
Result: Set of maximally focused communities C
1 begin
2     C ← {};
3     S ← sample(P);
4     for t ∈ S do
5         if ∃c ∈ C  t ⪰ c then
6             continue;
7         c ← climb(t, P, H);
8         if c ≠ NULL then
9             C ← C ∪ {c};
```

without replacement) is equal to 1 minus the probability of getting 0 successes. This probability is defined by the geometric equation that describes the CDF of k Bernoulli trials: $1-(1-p)^k$. Therefore, we have:

$$p_b = 1 - (1 - \big|C\big|/\big|P\big|)^k = 1 - (1-p)^k \tag{3.1}$$

We want to find the *minimum* value of k so that the right hand of Equation (3.1) is greater or equal to p_b. Let $q = 1 - p$ be the probability of failure in a single trial.

$$p_b \leq 1 - (1-p)^k \implies q^k \leq 1 - p_b \underset{q<1}{\implies} k \geq \log_q(1-p_b) \implies$$
$$\implies k \geq \frac{\log(1-p_b)}{\log(q)} \underset{argmin}{\implies} k = \left\lceil \frac{\log(1-p_b)}{\log(q)} \right\rceil$$

Note that k is not directly dependent to the size of the population P, only on the probability of success p_b. As an example, to find focused communities with at least 30% the size of population P and with success probability $p_b = .99$ we need at least 13 samples. For communities with size 70% or more and the same probability p_b we need only 4 samples.

3.1.3 Climbing Phase

The climbing phase follows the sampling phase by consuming the sampled datapoint and producing a maximally focused community. More specifically, a tuple t is received from the sampling phase and the focus metric from Equation (2.1) is utilized to *climb* the *lattice* (see Figure 2.2) from t to a new tuple $t' \preceq t$, so that the support of t' in P is maximized and is at least ξ, and t's focus metric remains below the relaxation threshold ϵ. Similar to hill-climbing techniques, in every new iteration a new neighbor of the current solution is generated until an acceptable solution is reached. A tuple t has N possible neighbors: each one can be reached by generalizing a different attribute value of t.

Basic Climbing Approach

The pseudocode in Algorithm 2 describes this process. Starting from a tuple t, a new neighbor is produced in every iteration till a maximally focused community or a HEAD tuple is reached. *HEAD* represents the unique tuple that has all of its attribute values fully generalized: $HEAD \equiv [*, *, ..., *, ..., *]$. An accepted solution (focused community) is reached when both conditions in line 5 in Algorithm 2 are satisfied (focus metric and support). These two conditions alone do not guarantee maximality therefore the algorithm will not return at this point but will continue until the *HEAD* is reached and at this point will return the most recent accepted value for t'. In line 7 the next attribute for generalization is selected: a_g. Different selection policies will yield different results and offer different guarantees. Using the selected attribute, a new tuple t_{temp} is generated, identical to the previous t_{temp} on all attributes except a_g, which gets generalized (line 8).

Since the climbing process always follows an upward path – a neighbor is created only by *generalizing* a single attribute – there is a well defined maximum number of iterations, equal to: $\sum_{i=1}^{N}(H[i].numLevels - 1)$, where $H[i].numLevels$ is the number

of hierarchical levels for the i^{th} attribute. This sum can be approximated by $O(N)$. However, the selection policy for the next attribute to generalize has a significant impact on the performance of extracting a tuple t' that eventually corresponds to a maximally focused community. We will first discuss the *exact selection policy* that guarantees the discovery of a maximally focused community and then propose a *greedy policy* for a more efficient selection.

Algorithm 2: Climbing phase

Data: Attribute tuple t, all tuples P, hierarchies $H[N]$
Result: Maximally generalized tuple t'

```
1 begin
2     t_temp ← t;
3     t' ← NULL;
4     while t_temp ≠ HEAD do
5         if focus(t_temp, P) ≤ ϵ and support(t_temp, P) ≥ ξ|P| then
6             t' ← t_temp;
7         a_g ← getNextAttributeToGeneralize(t_temp, P, H);
8         t_temp ← {a ∈ t_temp | a_g ← H.parentValue(a_g)};
```

We start with a policy for selecting the next attribute of a tuple t to generalize (a_g) which guarantees reaching the correct attribute values of a maximally focused community C, if one exists *and* $t \succeq C_t$. This policy involves choosing the attribute with a value that when generalized to the next hierarchical level results in the largest support for the new tuple:

$$\underset{a_g \in t}{\operatorname{argmax}}\, support(\{a \in t | a_g.value \leftarrow H.parent(a_g.value)\}, P)$$

where $a_g.value$ is the current value of the attribute a_g (e.g. if the attribute is Location, it could be Los Angeles or California). The argmax function returns the attribute value for which the tuple support attains its maximum value. The main drawback of this approach

is the need to calculate the support of N different tuples in each iteration. Since a total of $O(N)$ iterations is required to reach a maximally focused community, the total time complexity becomes quadratic ($O(N^2)$).

Theorem 1 *The generalization policy will lead to a maximally focused community C if the starting tuple $t \succeq C_t$.*

Proof: Let C be a *maximally focused* community with size $\left|C\right| \geq \xi P$ and with a focus metric less than ϵ. Let t be a starting tuple with n attribute values so that $t \succeq C_t$ (C_t can be reached by generalizing attribute values in t). C_t can be correctly reached from t if after $O(n)$ iterations t' becomes C_t. The only way that a selection policy can fail to reach C_t, during the climb from t to *HEAD*, is if one attribute value of t gets generalized beyond the corresponding attribute value of C_t. To prove the theorem we need to show that the selection policy will never select to generalize an attribute of t that has the same value with the corresponding attribute of C_t.

Let t_i and t_j be the i^{th} and j^{th} attribute values of t, and c_i and c_j the i^{th} and j^{th} attribute values of C_t. Assume that t_i has reached the same value with c_i, and that t_j has not: $t_j \succ c_j$. C is a maximally focused community so given the maximality property any further generalization of an attribute in C_t cannot lead to a new focused community. Therefore, the generalization of t_i will not increase the support of t while the selection of attribute t_j (or any other attribute not generalized to the same level with C_t) will result in a new tuple t' with an increased support. Thus, as long as there are attribute values in t that are not generalized to the same level of C_t, their selection will always be prioritized over attribute values that have reached the correct level of generalization, till all of them are correctly generalized. ■

Greedy Attribute Selection Approach

To improve the efficiency of the focused community extraction algorithm and render it scalable, we propose a *greedy* policy to select the attribute a_g: choose the attribute value of the tuple that has the smallest support in P (argmin). The intuition behind this approach is that in a focused community defined by N characteristics, the characteristic with the smallest support is the one that likely constrains the size of the community the most. More specifically, the support of a tuple t is equal to the size of the intersection of the N attribute values in t and the size of this intersection is bounded by the support of the attribute value with the smallest support. The only way to increase this bound is by generalizing the smallest attribute in order to match more datapoints. This observation is illustrated in Figure 2.1b: if either of v_a or v_c is generalized, the intersection of the three attributes will still be limited by value v_b and remain almost the same size. Instead, the generalization of v_b has the greatest potential to increase the intersection. The mathematical form of this policy is:

$$\underset{a_g \in t}{\operatorname{argmin}} \, support(a_g.value, P) \tag{3.2}$$

The main benefit of the greedy policy over the exact approach, is the improvement of time complexity. While we need to compute the support of N attribute values in each iteration, we do not need to actually perform the operation for every attribute value in every iteration, since only one of the support values changes: the support of attribute a_g which gets generalized. All other attribute values of the tuple remain the same therefore their support does not change in the next iteration. Storing in memory the support of the $N-1$ attribute values only a single support calculation needs to be performed per iteration. With an $O(1)$ time complexity per iteration the total climbing time complexity becomes $O(N)$.

The downside of the greedy policy is that it does not offer specific guarantees for reaching a maximally focused community. In fact, there is a specific case where the greedy approach might choose to generalize an attribute value that is not the correct one. Figure 3.2 visualizes this scenario where all of the necessary requirements to fail are met: Assuming that a correct community exists and is [male, California, 13-22], if the climbing process seeded by the tuple [male,San Francisco,18] has currently reached tuple [male, California, 18] then the greedy policy will select attribute value *California* for generalization since it has the smallest support. However, the correct choice would be to generalize the value 18 to 13-22 in order to reach the focused community. If California is generalized, the focused community will not be reached.

3.1.4 Accuracy and Efficiency

To measure the *accuracy* and *efficiency* of the proposed algorithm we created a synthetic dataset of artificial topic populations that contain random focused communities. Using a pseudo-random attribute generation process we were able to inject communities into populations and then test the algorithm for the expected result, something that is not realistically feasible in this scale on real data. The synthetic dataset was specifically constructed to examine the accuracy and recall of the approach and includes a complete spectrum of scenarios — some that might be rare in a real dataset. The generation process for each topic population includes three phases: (1) Choosing a random attribute space with number of attributes n (between 5 and 20), possible values for each attribute a_i (between 2 and 50000), and the number of levels in each attribute's hierarchy h_i (between 2 and 5). (2) Choosing the attributes of the focused community C by randomly selecting a value c_i for each attribute a_i, given equal selection probability to each level of the hierarchy h_i. The result is a tuple that defines the expected focused community.

This community is also assigned a randomly selected size ratio p_C between 30% and 90% of the total size of the topic population. (3) The creation of the topic population so that it includes datapoints for the focused community but also other *noisy* datapoints that might or might not be part of the community. The population size was randomly selected between $10,000$ and $1,000,000$ datapoints to simulate numbers close to ones observed in Twitter's trending topics. A total of $10,000$ population groups were created, each with a single maximally focused community. The algorithm settings that we used are: selection policy: greedy, sampling size: 20 datapoints, $\epsilon : 0.15$, $\xi : 0.3$

The algorithm was able to find the correct communities in each synthetic population with an accuracy of 93.1%. A community extraction was labeled as successful when the exact correct community (combination of attributes) could be identified. In the rest of the cases that failed, most of the time there would be a community attribute value or two that were more generalized than they should. Measuring the accuracy on a per-attribute value basis, instead of the whole tuple, the average accuracy is 97.2%. The running time for all 10000 cases was a little less than 10 minutes on a 2.6GHz CPU.

3.1.5 Handling Missing Values

As opposed to synthetic data, one of the challenges when dealing with real social datasets is the sparsity of attribute values. This observed sparsity (missing values) is due to the low recall of specific inference tasks which usually originates in the general lack of sufficient information to infer attributes with high confidence (e.g., not enough textual information to infer the age of a user). In the presence of missing values (symbolized with $\perp$), an attribute tuple will not match every datapoint that it should. For example, the tuple [California, Male, *] does not match the datapoint [Los Angeles, $\perp$, 18] because $\perp$ does not succeed Male. Therefore, if there are missing values in each attribute, the

observed size of the community and the size of the exclusive feature(s) will differ and the focus metric will not result to a focused community.

To overcome this problem, we allow a tuple to match missing values during counting. Referring back to the previous example, we allow the tuple [California, Male, *] to match the datapoint [Los Angeles, $\perp$, 18]. This alteration fixes the issue of under-counting a tuple, but introduces over-counting: additional datapoints are now counted as part of a community. However, the community size over-estimation is statistically bounded. Let v_f be the attribute value that plays the role of the exclusive feature in the focused community C and let m_f be the ratio of missing values for the attribute a_f. The focused community can be divided in two parts: the datapoints that belong in the community and have a value v_f for the attribute a_f and the datapoints that belong in the community and have a value $\perp$ for the attribute a_f (missing value). Similarly, the datapoints outside the focused community can be divided in two parts: the datapoints that have a value $v'_f \neq v_f$ for the attribute a_f and the datapoints that have a value $\perp$ for the attribute a_f (missing value). Note that there are no datapoints outside the community with value v_f for the attribute a_f based on the definition of the focused community. The datapoints that could be mistakenly counted are the ones outside the community, with a missing value. The expected size of this subset is bounded by: $m_f(1-\xi)\big|P\big|$. In the presence of many missing values it is recommended to use a higher support threshold ξ for the correct detection of focused communities since the above value gets closer to 0 when $\xi \to 1$.

3.2 Experiments with Twitter Data

To understand the effectiveness of the proposed algorithm we performed experiments on a real dataset from Twitter. We first present the available data and the inference process of the user attributes like location and gender. We then discuss some interesting

findings from the extracted topics and the corresponding communities in the results.

3.2.1 The Twitter Dataset

The used Twitter dataset contains a uniform 10% sample of all the tweets and Twitter users from the following two periods: September 12 to October 26 of 2013 (45 days) and April 16 to May 24 of 2014 (39 days). The pool of topics contains every mentioned hashtag or capitalized entity from the tweets' raw text. The extracted tweet features include location, the list of external user mentions (@-replies), the device the tweet was posted from (e.g. iPhone, Android, web browser), and the general sentiment. Location extraction was done on (1) the tweet level using Twitter's geo-tagging mechanism, and to further improve the recall, on (2) the user level using a user-provided raw text field (similarly to [11, 12]). To infer location based on the user's field we applied a simple but precise pattern matching process that could identify location patterns like: "City, Region, Country", or "Region, Country", or just "Country". To validate the patterns we used a Location hierarchy provided by the MaxMind database [13]. The user device was extracted from the available information provided by the Twitter API. To infer the sentiment of a tweet we used the SentiStrength tool [14]. Note that not all features were available in every tweet; for example, less than 2% of the tweets had an explicit location tag or non-neutral sentiment.

Meaningful and interesting community extraction requires a diverse set of user characteristics/demographics. To expand the number of extracted attributes from the Twitter dataset we additionally infer the users' age, gender, political affiliation, and sports team preference. To extract gender and age we applied existing language models extracted from Schwartz et al. [15] on social media data. To apply the models we gathered all the tweets of every user for each of the two analyzed periods of data. While this is an

expensive process, especially space-wise, it can be done offline and does not affect the complexity of our Sample&Climb algorithm. For political affiliation we gathered the official Twitter accounts associated with the three most popular US political parties: Democratics, Republicans, and Libertarians. Then, a user's political affiliation was determined based on the simple majority of interactions (@-replies) with these accounts (e.g. if a user mostly interacts with Democrats, their party preference was labeled as Democrat). Similarly for sports, we collected the Twitter accounts of teams, players, and coaches for the following four US professional sports: Baseball, Basketball, Football, and Hockey. For every sport, a user's team preference was inferred based on their interactions with each team's accounts. For both party and sports team preference we aimed for high accuracy even if it sacrificed recall. The average accuracy across all the attribute inference processes is 92.1% without including sentiment analysis which has a lower accuracy of 68.7%. Accuracy was manually calculated from random samples of 100 users and their tweets for each process. Table 3.1 shows the **accuracy** of each inference task. Given that the language models are in English, age and gender inference only works for English speaking users. Similarly, political affiliation and sports teams are focused on users within the United States and Canada. For the age and gender inference we list the calculated precision from Schwartz et al. Note that their models were tested on Facebook data, so accuracy might differ slightly.

In total, the experimental setup contained **10 attributes**: 1) Location (either from the tweet or the user), 2) Age, 3) Gender, 4) Political affiliation, 5) Baseball team, 6) Basketball team, 7) Football team, 8) Hockey team, 9) Tweeting device (e.g. iPhone), and 10) Sentiment. While sentiment is not strictly a user characteristic, it helps with the interpretation of the results by hinting at the attitude of the community towards the topic. Apart from Location and Device all hierarchies have only 2 levels (trivial). The Location hierarchy has 4 levels: city, region, country, and *. The Device hierarchy has 3

Inferred Attribute	Source	Accuracy
Location	Geo-tagged tweets	100%
Location	User specified location	96.1%
Device	Twitter API	100%
Gender	Schwartz et al. [15]	91.9%
Age	Schwartz et al. [15]	.84 (R value)
Political Affiliation	Interaction with parties	83.4%
Baseball Team	Interaction with teams	91.5%
Basketball Team	Interaction with teams	93.7%
Football Team	Interaction with teams	87.8%
Hockey Team	Interaction with teams	95.0%
Sentiment	SentiStrength [14]	68.7%

Table 3.1: Inference accuracy of Twitter attributes

levels: specific device, mobile/desktop, and *.

Setup and settings. The execution of the community extraction algorithm was applied on the stream of tweets using a *sliding window* of size 500,000. On a typical day this amount of tweets can be produced within two minutes of real time. For every new window new topics get introduced, existing topics receive additional mentions, and old topics get evicted. To reduce noise, candidate topics are required to have at least 50 mentions during the window. The rest of the algorithm settings are: selection policy: greedy, sampling size (k): 20 datapoints, ϵ : 0.15, ξ : 0.3. The choice of ϵ is based on the fact that Twitter data is noisy and the community extraction should be relaxed enough to accommodate this noise. The value of the support threshold ξ is based on the average population of a Trending Topic on Twitter, which is usually between 1K and 200K tweets, therefore we can expect communities of size between 300 and 60K users (smaller communities would not be interesting).

Table 3.2: Examples of general Trending Topics.

Topic	Size	Sentiment	Location	Age	Gender	Politics	Size
#PS4	114	*	*	13-18	Male	⊥	111
#Bring1DtoGreece	117	*	Athens:AT:GR	13-18	Female	⊥	110
#NavyYardShooting	5427	Negative	US	19-22	*	*	5218
#OscarTrial	1242	Negative	Johannesburg:ZA	*	Female	⊥	1133
#ReasonsToVisitEgypt	50	Negative	AL:EG, CA:EG	*	*	⊥	49
#DisneySide (day 1)	54	Positive	Anaheim:CA:US, Orlando:FL:US	*	Female	⊥	50
#DisneySide (day 2)	53	*	CA:US, FL:US	*	Female	⊥	51
Penn State	64	Negative	Bloomington:IN:US, Indianapolis:IN:US	19-22	Male	*	56
#auspol	55	*	Melbourne:VIC:AU, Sydney:NSW:AU	*	Male	⊥	51
#auspol	461	Negative	AU	*	*	⊥	457
#FreeJustina	54	Negative	Boston:MA:US	*	Female	Democrats	51
#cdnpoli	151	Negative	ON:CA	23-29	Male	Republicans	139
White House	2989	*	US	*	Male	Republicans	2868
#ObamaCare	5090	Negative	US	*	Male	Republicans	4818
#ObamaInThreeWords	246	Negative	US	19-22	Male	Republicans	224

3.2.2 Qualitative Evaluation of Twitter Results

For each window of 500k tweets, tweets were grouped by topics to form the topic populations and the focused community extraction algorithm was applied on each topic. The final outcome of this experiment, is a list of topics and the corresponding maximally focused communities that were extracted, in each window. The extracted community of a topic might differ between different windows as additional users mention the topic and the population changes. We highlight some topics to showcase interesting behaviors and qualitatively argue that the results actually make sense. These topics are listed in Table 3.2 (general interest trends) and Table 3.3 (trends with a sports related focus). A "*" value indicates that the attribute got generalized to its top level of the hierarchy. A "⊥" value indicates that there was not enough information to extract a specific attribute value (due to missing values). Attribute values for Device and Basketball team are omitted due to lack of space. Topics that appear twice are taken from different days, and are listed to show the dynamic nature of focused communities as the topic population grows or just changes.

An interesting topic worth discussing is the hashtag *#DisneySide* which was a social

Table 3.3: Examples of Trending Topics in sports.

Topic	Size	Location	Age	Gender	Baseball	Football	Hockey	Size
#TMLtalk	3437	Toronto:CA	19-22	*	⊥	⊥	Toronto Maple Leafs	3096
#AZvsNO	50	⊥	19-22	*	⊥	Arizona Cardinals, New Orleans Saints	⊥	50
#RedSox	528	Boston:US	19-22	Male	Boston Red Sox	⊥	⊥	411
#Boston	51	⊥	⊥	⊥	Boston Red Sox	New England Patriots	Boston Bruins	51

media campaign by US Disney Parks. Disney asked fans to tweet photos of their 'Disney Side' from their visit to a Disney theme park. During the first day, most of the tweets occurred in the two cities where a Disney park is located: Anaheim, California and Orlando, Florida. The next day, the campaign audience expanded to include the whole states of California and Florida.

Other interesting topics and communities identified by our algorithm include: The hashtag *#NavyYardShooting* is about the mass shooting that occurred on September 16, 2013 on a US military base at Washington, D.C. and at its early stages it was mostly discussed by young adults in the United States. The topic *#OscarTrial* refers to the trial of the South African Olympian Oscar Pistorius and our algorithm correctly captured the location of the focused community (South Africa). Of particular interest, is topic *#ReasonsToVisitEgypt* which originally started as a touristic campaign for Egypt but got highjacked with citizens' complains, hence the extracted negative sentiment. Topic *Penn State* is related to a college football match where college Penn State played in Bloomington, Indiana. Indianapolis is also in the results since it is the capital of the Indiana state and it is very likely that fans/students might have specified it as their location. *#auspol* is a hashtag about police brutality in Australia. In the early stages of the trend it was mostly mentioned in the two largest cities of Australia but as it became popular, the whole country became the focused community. The topic #FreeJustina is about an arrested female teen named Justina from Boston. We observe that women in

the area of Boston, MA, that also support the Democratic party, showed their solidarity to Justina through this hashtag. *#cdnpoli* stands for 'generic canadian political issues' and this is why the topic's location is in Canada. *#AZvsNO* stands for 'Arizona vs New Orleans' and is describes an American Football match. *#Boston* is an interesting case with a focused community of users that were fans of local teams in all three sports. Finally, topics like *#PS4*, which stands for 'Play Station 4', and *#BringOneDtoGreece*, which stands for 'Bring 1Direction (the boy band) to Greece', further show how our algorithm identified the correct characteristics of the interested populations in each case.

There are also cases of topics and communities that we could not explain by associating the topic to a real event or expected behavior. For example, the topic *#SundayFunday* was found to have a maximally focused community of young-adult female residents of Houston, Texas. Or, the topic *#DefyExpectations* was found to be discussed by a focused community of teenagers. It is hard to explain why these specific communities were interested in these generic topics at a particular point in time. There are several cases like these in our results which proves that the topic-mentioning behavior of users in Social Media can be unpredictable and will be further studied in future work. However, uncovering the underlying characteristics of the topic population is a significant step towards this direction. Finally, an interesting general observation is that for topics related to activism or politics, usually the male demographic was prevalent (with exceptions like #FreeJustina). For topics related to memes or pop culture, mostly the female demographic was prevalent.

3.3 Application: Community-based Topic Ranking

One potential application for the extracted focused communities is to re-rank trending topics in order to increase their engagement potential as a social content recommendation

system. In this section we discuss a ranking formula and then show through experimental evaluation that with very basic calculations, ranking by focused communities leads to more engaging topics as compared to two standard baselines. Ideally, the community attributes can be exploited to deliver a more personalized recommendation experience to users by showing them topics with similar characteristics. We plan to further explore increasing the relevance of trending topics through this approach in future work.

3.3.1 Ranking Formula

To obtain an interesting ranking of topics we use a combination of two measures: Inverse Community Frequency and Relative Community Popularity. Both measures aim to normalize the raw frequency of a topic in order to boost those topics with interesting focused communities. Inverse Community Frequency (icf) is inspired by Inverse Document Frequency from text document ranking in Information Retrieval. Here we use it in a similar context: to tune down community characteristics that get associated with many topics. A community characteristic that appears in few topics only should be more interesting. Inverse Community Frequency, measures how many topics in the whole window W of datapoints also share a community characteristic. For example, the icf of location Santa Barbara will depend on how many topics in W have a focused community that contains Santa Barbara. The icf score of a community C is the product of icf scores for each attribute value in C. The **icf score** for a single attribute value a is equal to:

$$icf(a) = log\frac{N_t}{|\{T \in W | a \in C\}|}$$

where N_t is the total number of topics in W and the fraction denominator is equal to the number of topics T in W with a community C that contains the attribute value a. Relative Popularity takes values between 0 and 1 and practically compares the size of a

topic's focused community with the size of the community with the same characteristics in the window W of datapoints. The **relative popularity score** is calculated as the fraction of the support of a community in P over the support of the community in W:

$$rp(C) = \frac{support(C, P)}{support(C, W)}$$

For example, if a topic is being discussed by 100 women and the number of women in W is also 100, then this community has a relative popularity of 1. The overall scoring function is based on each topic's extracted focused community C and uses both notions of relative popularity and exclusive focus:

$$score(T) = |P| \times rp(C) \times icf(C) \tag{3.3}$$

where C is a focused community of the topic T, P is the population of the topic. The overall score of a topic is proportional to the topic's raw frequency (size of P), the relative popularity score of the topic's community, and the icf score of its community. Using this score metric we rank the candidate topics and obtain a final list of top-k topics which we will refer to as *community-based topics* or c-topics.

3.3.2 Experiments

To evaluate the ranking of *community-based topics* we used two baselines: (1) the *raw-frequency baseline* where topics are ordered by the number of mentions (also referred to as f-topics) and (2) the *burstiness baseline* where topics are ordered based on their temporal trendiness, which is calculated through chi-squared (expected vs. observed frequency of the topic). The latter baseline is time sensitive and requires the monitoring of each topic's historic frequency to capture its average and seasonal changes in frequency. The

average historic frequency is the expected value and is used in the calculation of the chi-squared formula to measure how bursty a topic might be, given a new observed frequency: $\chi^2 = (Expected - Observed)^2/Expected$. We will also refer to the burstiness-based topics as b-topics.

Based on the experimental results, we found that raw frequency leads to popular but not necessarily informative or disparate topics (e.g. #ipad). Burstiness leads to better topical diversity by eliminating those high frequency topics that are consistently popular. On the other hand, topics ranked based on their focused-community characteristics appear to generally be more interesting and are further enhanced with the information of *who* is interested in each topic. The average similarity between the community-based topics and each baseline was measured with the *Set Based Measure* described in [16]. In general, the goal is to determine the fraction of content overlapping (set intersection) at different depths of the ranking lists. Between the raw frequency ranking and the community-based ranking the average set based measure with a depth of 20 is equal to 0.089 while with a depth of 10 is 0. Between the burstiness based ranking and the community-based ranking the average set based measure with a depth of 20 is equal to 0.122 while with a depth of 10 is 0.098. These values indicate that the three rankings produce mostly heterogeneous top-k lists and signifies that highly popular or bursty topics usually do not contain focused communities.

As with many unsupervised learning tasks, evaluating the produced results is a challenging task. In content recommendation systems used by real users, one can run A/B tests to compare the success of the algorithm with a baseline. To evaluate the community-based topics in terms of potential usefulness and interestingness we (a) measure the entropy of the results as an objective quantitative measure, and (b) asked human evaluators to choose their favorite topics from a pool.

Using the notions of Self-information and Entropy from Information Theory we pro-

vide a measure of the information content for community-based trending topics. Self-information captures how surprising an event is based on the probability of the event. The entropy of the experiment (extracting community-based trending topics) is the expected value of every trending topic's self-information. The self-information of the community C_T for a single topic T is $I(C_T) = -log_2(Prob(C_T))$. Intuitively, the less likely a community is to observed the higher its self-information. The prior probability of C_T can be measured in the sliding window as the percentage of datapoints that contain C_T. The entropy of the results is equal to the expected value of all topic communities: $E[I(C_T)]$ (measured in bits). We also measured in the same way the entropy of communities associated with trends ranked by raw frequency and burstiness. In the majority of those cases, topics did not have a focused community but rather were mentioned by users with dispersed attribute values. However, we can still calculate the probability of the observed population characteristics for each topic based on the prior probabilities from the sliding window. The average entropy for the community-based topics was found to be **1.87** bits, for frequency-based topics it was much lower: **0.27** bits, and for burstiness-based topics it was similarly lower: **0.35** bits. This indicates that the extracted topics using our method contain surprising and potentially useful communities that cannot be trivially anticipated or that are not observed in topics ranked by frequency/burstiness.

Since we aim to use the new ranking to improve the recommended social content, we need to observe that real humans would be interested in viewing more content related to an extracted community-based topic. To quantify this property we use the two baselines described above, raw frequency and burstiness. We offer to each evaluator an unlabeled selection of 10 topics (pool) and ask them to pick the top 5 (in no particular order) based on which they find the most interesting. In the experiment description a topic is defined as *interesting* to a user if they would like to read more about it: get tweets about it, read news articles, see related images, etc. In the first experiment each pool of 10 topics

included 5 frequency-based and 5 community-based topics. In the second experiment, each pool contained 5 burstiness-based and 5 community-based topics. In both cases we evaluated how community-based topics compare to each baseline. To reduce any bias on the reported evaluations results, we performed each experiment with 5 different topic pools (so a total of 10 pools was created). Each pool was evaluated by an average of 61 Amazon Turk workers located in the United States.

The results are shown in Table 3.4 for the first experiment (f-topics baseline) and Table 3.5 for the second experiment (b-topics baseline). We counted for each pool how many times each topic was selected as interesting and sorted them by this number. The first three rows of each table display the percentage of community-based topics (c-topics) in the top-1, top-3, and top-5 of the evaluators' selections respectively. On average, the 73.3% of the top-3 selected topics was comprised of community-based topics when compared with raw-frequency topics and 79.96% when compared with burstiness-based topics. For the top-1 in the majority of the pools the evaluators selected a community-based topic most of the times. These values indicate that for both baselines, the *majority* of selected topics was community-based. The final two rows of each table show the percentages of c-topic and baseline-based topic (f-topic and b-topic) selections — how many times an evaluator clicked a topic of each category as interesting. This value can also be viewed as the probability of each category/method to produce an interesting topic. On average, community-based topics have 26.86% better chance to be more interesting than raw-frequency ranked topics and 49.43% better chance than burstiness ranked topics, which shows that in most cases users found our algorithm's results more appealing. Some topics ranked by raw frequency or burtiness are still interesting to users due to their popularity, but overall our method delivers more appealing results to the average person as represented by Amazon Turkers.

The histogram in Figure 3.3 shows the results of the first experiment (popularity

Table 3.4: Evaluation results from Amazon Turk on 5 different pools of topics. Comparison with raw-frequency baseline.

	Pool 1	**Pool 2**	**Pool 3**	**Pool 4**	**Pool 5**	**Average**
% of c-topics in top-1	0%	100%	100%	100%	100%	80%
% of c-topics in top-3	33.3%	100%	66.6%	100%	66.6%	73.3%
% of c-topics in top-5	60%	60%	40%	80%	60%	60%
% of clicks on c-topics	49.75%	54.86%	52.28%	64%	58.75%	55.92%
% of clicks on f-topics	50.25%	45.14%	47.71%	36%	41.25%	44.08%

Table 3.5: Evaluation results from Amazon Turk on 5 different pools of topics. Comparison with burstiness baseline.

	Pool 1	**Pool 2**	**Pool 3**	**Pool 4**	**Pool 5**	**Average**
% of c-topics in top-1	100%	100%	100%	100%	100%	100%
% of c-topics in top-3	66.6%	66.6%	66.6%	100%	100%	79.96%
% of c-topics in top-5	60%	40%	80%	100%	100%	76%
% of clicks on c-topics	54%	52.4%	62.2%	63.6%	66.8%	59.8%
% of clicks on b-topics	46%	47.6%	37.8%	36.4%	33.2%	40.02%

baseline) performed the same way as with Amazon Turk on 12 Computer Science graduate students from the University of California, Santa Barbara. Topics denoted by (C) are community-based and topics denoted by (F) are frequency-based. All of the top selections ended up being community-based topics even though some evaluators were interested in simply popular topics like #ThrowbackThursday or #NowPlaying.

This difference between Turkers and Computer Science graduate students indicates that a group of people with a biased interest in news (like graduate students) might find content based on topics with a community focus more interesting than a random diverse population (like US based Amazon turkers). In future work we plan to explore personalized topic scoring that produces personalized rankings to increase the recommended content's relevance.

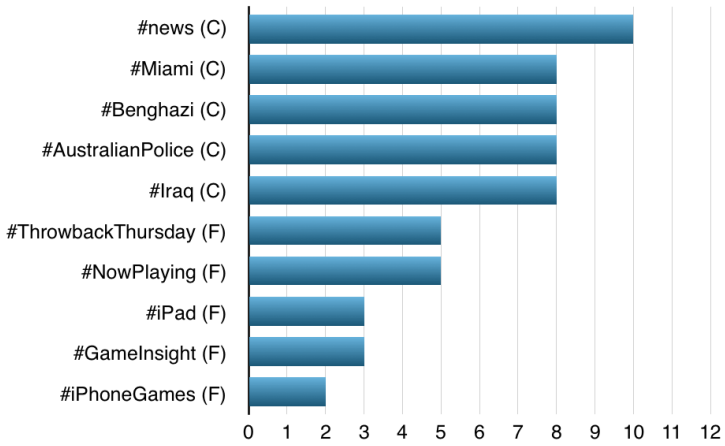


Figure 3.3: People's topic preference from a pool of 10 topics.

3.4 Related Work

Existing social content recommendation systems have mainly relied on the similarity of users in the social network. Walter et al. [17] have proposed a model to use the users social connections to reach contents and filter the contents by their trust relationship. Golbeck et al. [18] have considered online social networks as recommendation networks by exploiting the easiness of information cascades on such platforms. DuBois et al. [19] have proposed to improve the collaborative filtering recommendations by using the trust information as the weights between users. Finally, a hybrid approach was introduced by Wang et al. [20]. In the current work we utilize the notion of trending topics as a platform for content recommendation and identify communities based on common attributes (demographics) between the users to further boost this notion. Many algorithms have been proposed for discovering interesting trending topics utilizing techniques from the areas of Anomaly Detection, Data Streams, and Clustering. In existing studies, trending topics are mined for specific interest areas like Sport [21], Earthquakes [22], News reporting [23, 24], general event detection [25, 26], or search support on trending events [27]. In this paper we research the novel idea of identifying the underlying user commu-

nities that are interested in social media topics and then utilize this knowledge with the overall goal of providing more interesting, insightful, and relevant content to the users of the social network. Such a task can be challenging in terms of complexity when dealing with a non trivial number of community characteristics. The official Twitter Trending Topics are personalized to the user by displaying the top topics from categories the user is interested in. This is a simple approach to serve relevant trends but focuses only on interests (e.g. Technology, Politics) or location, and does not identify topics where the underlying population has specific properties, thus, can miss less popular topics with highly interesting community characteristics.

Our algorithmic work builds on many techniques in areas that share common properties with this problem, most notably from Subspace Clustering and Frequent Itemset Extraction/Association Rule Mining. Association Rule Mining using Frequent Itemset Extraction [28] is a well studied area and poses similarities to the attribute-based community extraction. Techniques that sample the data to perform fast itemset extraction are the closest to our proposed approach since probabilistic algorithms are used to reduce complexity. Such techniques include Toivonen [29] and Chakaravarthy et al. [30]. Clustering algorithms for data in multiple dimensions, known as subspace clustering algorithms, are usually divided in two categories: density-based methods and k-means-based methods. A detailed survey on both categories can be found in [31]. Similar algorithmic principles are used to solve the frequent itemset and association rule mining problems as well (e.g. a-priori pruning is used in [28] and [32]). Our approach mainly differs from existing sub-space clustering and association rule mining techniques by combining a sample phase and then a greedy climbing of the lattice to efficiently (linear time) identify the combination of user characteristics that form a community for a particular trending topic. Efficiency is key since vast amounts of data are processed in real time.

Finally, similar to our approach, probabilistic or Monte Carlo based methods for com-

munity extraction have also been explored in Perozzi et al. [33]. They study extracting community attributes that form highly connected subgraphs within the social network. To detect the correct values for each attribute they utilize Monte Carlo sampling to randomly select values until a connected subgraph is formed. Our approach seeds the process by sampling k datapoints from a trending topic's population. This leads to a much more agile and efficient attribute value selection process.

3.5 Remarks

We study the problem of extracting multi-dimensional communities focused on individual topics by introducing the notion of a maximally focused community with properties that enable the efficient discovery of interested communities defined by a subset of social attributes. These properties led to the development of an algorithmic framework for the extraction of maximally focused communities of any topic with proved linear time complexity. Finally, we provide a robust ranking that boosts topics with *relatively popular* or *exclusively focused* communities through metrics adapted from IR.

Extensive experimentation was conducted on two different datasets: one real from Twitter with data from large periods in 2013/14 and one synthetic. The results highlight the efficiency, correctness, and stability of our proposed algorithm. As an application, we demonstrate the power of our approach to identify interesting communities for trending topics, sometimes expected and sometimes unexpected. It is interesting to observe that females in Boston, which also support the Democratic party, show their solidarity to an arrested teen (#FreeJustina). It is unexpected to discover the hijacking of a touristic hashtag in Egypt from local citizens that try to raise awareness for the country's political situation (#ReasonsToVisitEgypt). Such data can be used to better understand a topic's population and, essentially, recommend more relevant and interesting social content.

Chapter 4

Privacy in the Context of Community-Aware Trending Topics

In this chapter we formally introduce a novel privacy model that captures the notion of sensitive attribute inference in the presence of community-aware trending topic reports where an attacker can increase their inference confidence by consuming these reports and the corresponding community characteristics of the involved users. We discuss a basic attack and provide an efficient algorithm that preserves the privacy of each individual user so that sensitive attributes can not be successfully inferred. To the best of our knowledge we are the first to address this notion of privacy and introduce an algorithm that uses the idea of attribute generalization in combination with Artificial Intelligence techniques to efficiently defend against this type of attack.

4.1 Motivation

Due to the public nature of Online Social Networks like Twitter, apart from identifying the real identity of a user, an attacker will usually try to *infer* sensitive attribute values

of certain users utilizing knowledge of the social network (who is a friend with whom, or who follows who). Furthermore, a sensitive attribute inference attack is also a significant risk in the context of community-aware trending topic reporting and to the best of our knowledge has not been studied before. At the same time, large Social Media websites like Facebook and Twitter already have proprietary methods for inferring social attributes of their users that are not explicitly provided by them. Recently, it was revealed that Facebook is able to learn a user's political preference between values like "Liberal", "Very Liberal", "Moderate", or "Conservative". This is a particularly interesting case since user content on Facebook is usually not accessible to anyone except the user's immediate social network. However, if sensitive attribute information, like political preference, is used in the context of enriching other features which are publicly known, like Facebook's Trending section, then this feature could start leaking sensitive information to virtually anyone.

To demonstrate how sensitive attribute inference could be applied as an attack in the context of trending topics, we provide a hypothetical example in Figure 4.1 where users mention certain topics that were reported as trending from a community-aware algorithm (listed in the table at the top of the figure). The information in the table is public to everyone, similarly to the lists of Trending Topics that Facebook and Twitter already publish to their users in general, or even for specific geographic locations. The main difference is that each topic is also linked with values for specific attributes like gender, age, location, political preference, etc. The association of an attribute value with a topic indicates that this specific attribute value is a characteristic for the majority of the users that mentioned the topic (but not necessarily all of them). For an attacker, this means that they cannot be 100% confident that every user mentioning topic T_1 lives in Boston. However, when users discuss *several* topics, the attacker's confidence may increase. As shown in Figure 4.1 Alice and Bob each mention some of the topics that

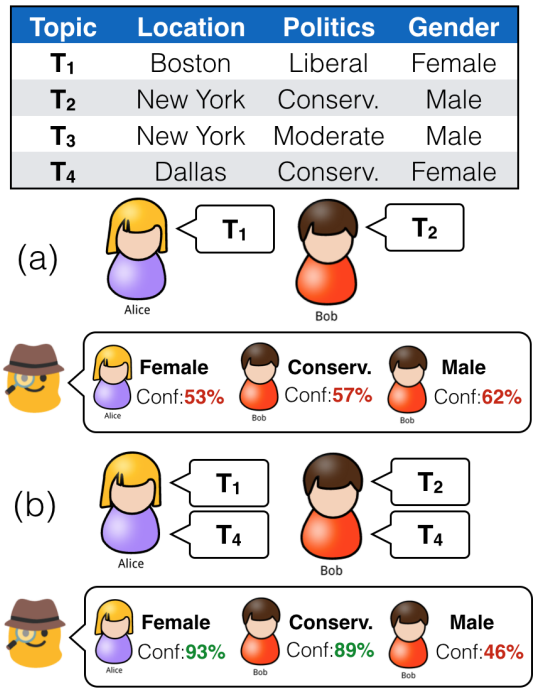


Figure 4.1: Alice and Bob are two users who have discussed some topics. These topics were reported as trending and additionally, for each topic certain demographic information was extracted for 3 attributes: Location, Political Preference, and Gender. These values indicate that a significant *portion* of all the users that mentioned each topic, belong to the community defined by those values. An attacker can observe these values and can also find which topics Alice and Bob have discussed. Based on this knowledge, the attacker can infer certain attribute values of Alice and Bob with certain confidence. In case (a), where Bob and Alice have only discussed a single topic, the attacker has low inference confidence. In case (b), Bob and Alice have also discussed topic T_4 which increases the confidence of the attacker for Alice's gender and Bob's political preference but at the same time decreases the confidence for Bob's gender because T_2 and T_4 have mostly male and female communities correspondingly.

happen to be listed in the table of trending topics. Since the attacker can obtain a list of the users that mentioned each topic (e.g., Twitter provides such search functionality), they can also *increase* their confidence (note the difference between cases (a) and (b)) in inferring Alice and Bob's sensitive attributes like political preference or gender without even accessing their posted content or network.

In Table 4.1 we list some real examples of topics and their corresponding community characteristics (attribute values) that we extracted from Twitter data. The communities are characterized by values for several attributes including Location, Gender, Age, Political party (US only), or even Sports teams. Note that these attribute values are temporal and might change over time, even for the same topics. Each topic has a frequency (how many unique users mentioned it) and a community defined by the attributes that describe a significant part of the users that mentioned the topic. In practice, it is impossible to observe topics where the entirety of their population forms a homogeneous community on some attribute values, therefore, the reporting algorithm will only guarantee that at least some percentage of this user population shares the reported attribute values. Note, that a community is not necessary to have a value for every attribute, as it happens for "#NFL" where the user population is homogeneous only on Gender and Location and not in Age or Politics. In the last column of the table we provide the number of privacy violations for each topic, i.e. the number of social media users that will have *at least one attribute* exposed to an attacker if the corresponding trending topic is publicly reported.

An attacker similar to the one in Figure 4.1 can peruse the rows of Table 4.1 and attempt to infer sensitive attribute values for the involved users. If there is a user that mentioned both topics #ObamaCare and #ObamaInThreeWords then the attacker can be very confident that the user supports the Republican party, that they are located in the United States, and moderately confident that they are male and a young adult. In the presence of even more sensitive attributes like sexual orientation, religion, or race, such

Table 4.1: Real examples of community-aware trending topics

Topic	Frequency	Community characteristics	Size	Violations
#NavyYardShooting	5427	Location: USA, Age: 19-22	5218	2561
#NFL	1534	Gender: Male, Location: USA	1212	389
GOP Debate	3278	Gender: Male	3004	36
#FreeJustina	54	Location: Boston, Gender: Female, Political party: Democrats	51	13
#OscarTrial	1242	Location: Johannesburg:ZA, Gender: Female	1133	345
#ObamaCare	5090	Location: USA, Politics: Republicans	4818	1002
#ObamaIn3Words	246	Location: USA, Age: 19-22, Gender: Male, Politics: Republicans	224	76
#RedSox	528	Location: Boston, Age: 19-22, Gender: Male, Team: Boston Red Sox	411	256

inference attacks need to be understood and prevented. Note that this kind of attack is different from existing privacy scenarios where the attacker infers sensitive attributes through the user's local social graph (e.g., [34]). In the case of community-aware trending topics, membership to a community is implicit and happens just by mentioning certain topics. Therefore, even if a user is careful with which groups they subscribe to or become members of, or with whom they socially connect with, sensitive information can still be exposed simply through the mention of a topic.

4.2 Related Work in Privacy

Data privacy is a thoroughly studied area and several families of algorithms have been proposed to deal with different kinds of attacks, mostly on published anonymized datasets. Most notably, the concepts of k-anonymity [35], l-diversity [36], t-closeness [37], and Differential Privacy [38] include methodologies to preserve data privacy and

information anonimity. However, privacy in Online Social Networks follows a different data model where most of the information is publicly available: the Twitter social graph, the set of online postings by every user in Twitter, user membership in Facebook pages, etc. What is not accessible though, is information about sensitive characteristics that users might want to keep hidden from the general public. An attack to discover these characteristics is known as sensitive or private attribute inference.

There are studies and published algorithms for inferring user demographics based on the content posted by social media users or their social network. [15] developed language models to identify the gender and age of Facebook users. [39] describe a method to infer user demographics by utilizing external knowledge of website user demographics and correlating it with a social media service. Their approach mainly differs from Schwartz et al.'s in its ability to infer the user characteristics without analyzing the content of postings. While these are considered valid sensitive attribute inference methods, they do not study the privacy and utility implications.

[34] were the first to study the privacy of sensitive attributes in the context of Online Social Networks. They describe a variety of attack models to infer sensitive user attributes but the model most related to the current work, is the model that utilizes the membership of users in Facebook pages. This model is similar to the "membership" of a user to a trending topic's community. However, they do not provide any algorithmic solution since it is the choice of the user to subscribe to a page. In Privometer, [40] measure how much privacy leaks from certain user actions (or from their friends' actions) and create a set of suggestions that could reduce the risk of a sensitive attribute being successfully inferred, like "tell your friend X to hide their political affiliation". Similar to Privometer, [41], and then [42], propose a method for preventing information leakage that introduces noise, by removing edges or adding fake edges, to the social graph. This idea was then extended to a finer-grained perturbation in [43] where edges are only added *partially*. [44] built

a system called "curso" that identifies when a user's privacy is violated through the analysis of their local network. There are also studies that focused on the anonymization of network data where the attacker tries to statistically infer the relationship between members of the social network. Most prominent works in this area include [45] and [46]. [47] also studied the same problem but specifically consider *distributed* social networks.

Dealing with privacy on a virtually infinite stream of data poses its own challenges and most of the aforementioned techniques and analyses focus on static datasets/databases. Dwork et al. have studied the problem of creating privacy-preserving algorithms in a streaming environment and proposed a family of algorithms called Pan-Private Streaming Algorithms [48]. However, the main focus of these algorithms is to deal with specific kind of attacks where the attacker might be in control of the machine where the algorithm is running but does not have access to the stream. This does not apply to our problem where an attacker has access to every social posting.

4.3 Data and Attack Models

4.3.1 Data Model

The users of a Social Media service are represented as a set $U = \{u_1, u_2, ..., u_n\}$. Each user u is associated with a vector v of k *sensitive* attributes (e.g., location, age, etc.). The attribute a_i of a user u ($u.v.a_i$) can take on one of a set of possible values $\{a_{i1}, a_{i2}, ..., a_{im_i}\}$, where m_i is the corresponding attribute's total number of unique values. The values of an attribute form a *hierarchy* which for some attributes can have a significant depth (e.g., for location: cities, to regions, to countries, to continents, to wordwide) or be trivial (e.g. for gender: from male and female to any gender). An attribute value can be *generalized* by being replaced with an ancestor value from the hierarchy. A user can mark a set of

attributes as sensitive and keep them private. Or depending on the nature of an attribute, e.g., race, which the social media service might infer using its own proprietary inference algorithm, it could be considered as sensitive for everyone.

The content of the Social Media service is represented as an infinite stream P of posts. Every post $p \in P$ has a unique author (user) $p.u$ and contains an arbitrary number of topic keywords $p.T = \{t_1, t_2, ...\}$. We define a publicly available search function $SEARCH$ that returns all the users mentioning a given topic keyword t: $SEARCH(t) = \{p.u | t \in p.T\}$. The number of users mentioning t is referred to as *topic population* and its size is equal to $|SEARCH(t)|$ and referred to as *topic frequency* (second column in Table 4.1). We can assume that each user that mentions topic t is counted only *once* to avoid bias from spamming. The search function $SEARCH$ is defined for multiple topics as well, and returns the *intersection* of the users that mention all the given topics.

We define a *homogeneous community* as a group of users with identical values in some of their attributes, but not necessarily connected in the social graph. More formally, a *homogeneous community* contains users that share the same values for a combination of attributes $C \in \wp\{a_1, a_2, ..., a_k\}$ where $\wp$ is the powerset symbol and a_i is a user attribute (e.g., location, age, etc.). Users that live in San Francisco, are 25 years old, and are male, form a homogeneous community that contains all the users identified by these values for the attribute combination {location, age, gender}. Users in New York form another homogeneous community defined by the singleton attribute combination {location}.

A *community-aware* trending topic algorithm (referred to as $CATT$ – citation removed for blind review) identifies topic keywords mentioned by a *homogeneous community* that has at least size ξ of the total topic population ($0 < \xi \leq 1$). For example, if $\xi = .7$, a topic with frequency 1000 will have *at least* 700 users forming a homogeneous community. The CATT algorithm reports records in the form of a stream of tuples: (t_i, C_i), where C_i is the set of attribute values that define the homogeneous commu-

nity CATT identified for topic t_i. If a topic t has no homogeneous community of size $\xi|SEARCH(t)|$ or larger associated with it then it isl not reported by CATT. We will refer to homogeneous communities simply as *communities* for the rest of the paper and to topics extracted via a community-aware algorithm as *community-aware topics*.

CATT extracts trending topics using a *batch-based* sliding window on the stream of social postings of the service. At the end of each window, CATT reports a set of pairs (t_i, C_i) which includes all the extracted topics from the current window. We refer to the output of CATT for each window of social postings as a *batch*. Table 4.1 shows an example of such a batch that contains 8 pairs. Through the definition of community-aware trending topics, the users of the social media service inherit an implicit membership to communities just by mentioning certain topics. Using a single reported pair (t_i, C_i) one can infer that at least $\xi\%$ of the users in $SEARCH(t_i)$ are characterized by the values of C_i. This constantly increasing knowledge enables an attacker to gradually improve their inference confidence for a given user's sensitive attribute(s).

Note that execution of CATT requires the knowledge of community attributes for the involved users. Realistically, CATT is executed by the Social Media service itself which has access to private user information or even its own proprietary method to extract attributes. Attackers lack access to the necessary information to execute CATT themselves.

4.3.2 Attack Model

A CATT algorithm reports a stream of batches of pairs (t_i, C_i). The attacker knows CATT's threshold ξ, as it is public knowledge, has access to the output stream, and to the search function $SEARCH$ which returns the set of users that have mentioned the provided topic(s). It is also safe to assume that the attacker has general knowledge of each attribute's prior distribution. For example, such knowledge might include the location

distribution based on a Census, the age distribution based on published statistics from the social media service, the gender distribution based on users that have this information public, etc. We can safely assume that the attacker is omnipotent and can indefinitely store the pairs (t_i, C_i) and the corresponding sets of users $SEARCH(t_i)$. The goal of the attacker is to infer a user's sensitive attribute by exploiting the knowledge of each topic's community C_i and the users associated with it. In the presence of an omnipotent attacker a privacy preserving algorithm must maintain all previous trending topics and communities to accurately calculate the probability distribution of the sensitive attribute values, of each user.

In related literature on sensitive attribute inference [34, 40, 42], an attacker would train a Naive Bayes Classifier to choose the value of a sensitive attribute L that maximizes the probability distribution $P(L|u.T)$. However, though Naive Bayes is known to be a decent classifier, it is also known to be a bad estimator [49]. For the inference process to be accurate, a high probability bound is necessary, so we consider that attack to be successful only when the inference probability of an attribute value is greater than a set threshold θ (e.g., $\theta = .75$ or $.85$) and not just by simply being the maximum over any other value. We will be using a global value for θ across all attributes and users, but the proposed model and algorithm support different values for each combination of attribute and user, if this is desired.

4.4 Privacy Model

4.4.1 Sensitive Attribute Inference

Having established the models for the data (social stream) and the attacker (inference of sensitive attributes) we can now formally define the privacy model. For every user in

the social network that discusses several topics in a streaming fashion, we want to protect against having their sensitive attribute values leaked through the continuous reporting of community-aware trending topics. Specifically, any attacker that has access to current and historical reports of community-aware trending topics should not be able to infer any user's sensitive attribute with confidence that is higher that a set value θ. At not point should an attacker be able to infer a *lower bound* for the distribution $P(L|u.T)$ (probability distribution of sensitive attribute L of a user u given the topics T that u has mentioned), that is higher than θ.

Definition: If there is even a single case where a user's sensitive attribute can be inferred with confidence larger than θ, this comprises a *privacy violation*. A community-aware trending topic algorithm that is capable of maintaining a record of zero privacy violations while it continuously reports new batches of topics is called *θ-private*.

Referring back to the example of Figure 4.1, if θ is set to .75 then an algorithm that reports the topics in the table of the figure is *not* θ-private in case (b), since the attacker can infer the gender of Alice and the political preference of Bob with confidence that is higher than θ. To make the algorithm θ-private we would need to obfuscate the gender and political preference associated with topics T_1, T_2, and T_4. If Alice and Bob had only discussed topics T_1 and T_2, as in case (a), then the algorithm would be θ-private for this specific instance.

The inference of a sensitive attribute involves estimating the probability of a specific value given some background knowledge. As already discussed, the attacker has access to prior attribute probabilities and the output and settings of CATT. The Naive Bayes classifier is a powerful and simple technique to calculate the probability of a sensitive attribute value. Arguably, if the attacker has additional information of other sensitive attributes (e.g., already knows that Alice is a woman because she has her own photo in her profile) then they can get a better estimation of the probability of another sensitive

attribute, like her location, than they would from Naive Bayes. In the following subsection we focus on the calculations necessary to get a lower bound of the probability $P(L|u.T)$ using Naive Bayes. The end goal is to anticipate what values the attacker can successfully infer so that they can be kept private. This is typically easy since the attacker's knowledge is generally based on publicly available information and the privacy model can incorporate it if necessary. To keep things simple, for the rest of the paper we assume that the attacker has no existing knowledge of sensitive attribute values and therefore the Naive Bayes Classifier can set a precise upper bound. The introduced privacy model is independent of how $P(L|u.T)$ is calculated by an attacker and the privacy preserving algorithm proposed later can be easily adjusted to calculate these distributions differently.

4.4.2 Naive Bayes Inference

Given a collection of topic and community tuples (t_i, C_i) (the output of CATT) and a search function $SEARCH$, an attacker may attempt to infer the sensitive attributes of users that mention at least one of the topics t_i. Let u be a user that has *mentioned* k topics $t_1, t_2, ..., t_k$ and let L be one of the user's sensitive attributes (e.g., location). The probability distribution of L, given that the user mentioned some topics $t_1, t_2, ..., t_k$ is:

$$P(L|t_1, t_2, ..., t_k) = \frac{P(t_1, t_2, ..., t_k|L)P(L)}{P(t_1, t_2, ..., t_k)} \tag{4.1}$$

by applying the Bayes Rule. $P(L)$ is the prior multinomial distribution of the attribute L and can be assumed to be known to an attacker based on their general knowledge on such information. The probability distribution of a user mentioning topics $t_1, t_2, ..., t_k$ given L, $P(t_1, t_2, ..., t_k|L)$, is equal to the number of users u that mention all the k topics and have a specific value for L, over the total number of users with that value of L. For

example, for $L = a$:

$$P(t_1, ..., t_k | L = a) = \frac{|\{u | u.v.L = a, t_1 \in u.T, ..., t_k \in u.T\}|}{|\{u | u.v.L = a\}|} \tag{4.2}$$

where $u.v.L$ is the attribute L in the user's vector of attributes v. Similarly, the prior probability of topics $P(t_1, t_2, ..., t_k)$ is equal to the number of users that mentioned these topics over the total number of users n: $|SEARCH(t_1, t_2, ..., t_k)|/n$

While an attacker might have knowledge of the attribute's multinomial distribution and the ability to calculate the prior probability of any topic combination (using the search function $SEARCH$), they cannot compute the set of users that have a specific attribute value $L = a$: $\{u | u.v.L = a\}$. Instead, they can obtain an approximate value of the probability distribution $P(t_1, t_2, ..., t_k | L)$ based on the reported tuples from CATT. The attacker can exploit the guarantees provided by CATT that a reported trending topic t_i has a population of size $|SEARCH(t_i)|$ with a homogeneous community C_i with size at least $\xi |SEARCH(t_i)|$.

More specifically, if the attribute L is not part of C_i, then the topic population of t_i follows the prior distribution of L: $P(t_i | L) = P(L)$. If $L \in C_i$ and has a value $L = a$, then applying the Bayes Rule we get:

$$P_{approx}(t_i | L = a) = \frac{P(L = a | t_i) P(t_i)}{P(L = a)} = \frac{\xi}{P(L = a)} P(t_i) \tag{4.3}$$

Similarly, the probability that a user with attribute value $L = b$ mentions topic t_i is equal to:

$$P_{approx}(t_i | L = b) = \frac{P(L = b | t_i) P(t_i)}{P(L = b)} = \\ = \frac{(1 - \xi) P(L = b) |SEARCH(t_i)|}{P(L = b) n} = (1 - \xi) P(t_i) \tag{4.4}$$

The attacker can now approximate the probability distribution (4.2) by assuming topic independence given L:

$$P_{approx}(t_1, t_2, ..., t_k|L) = \prod_{i=1}^{k} P(t_i|L) \tag{4.5}$$

where each factor of the product can be computed using the probability formulas from (4.3) and (4.4). Note that topic independence given L is an assumption that can be true when the number of topics k is large. It practically means that only one user mentions all the specific topics. For example, if a user u mentions 2 topics t_1, t_2 and $SEARCH(t_1, t_2) = \{u\}$ then u is the only common user mentioning both topics so, assuming that $|SEARCH(t_1)|$ and $|SEARCH(t_2)|$ are large numbers, t_1 and t_2 are statistically independent.

An attacker can use the following formula to approximate the distribution $P(L|u.T)$:

$$P_{approx}(L|u.T) = \frac{nP(L)\prod^{t_i \in u.T} P(t_i|L)}{|SEARCH(u.T)|} \tag{4.6}$$

If for any value of $L = l$, the probability $P(L = l|u.T)$ becomes larger than the threshold θ then we assume that the privacy of this user for L is violated.

4.5 Privacy Preservation Methodology

A *community-aware* trending topic algorithm is also *θ-privacy-preserving* if its output does not enable the inference of sensitive user attributes with a confidence greater than a threshold θ, for any of the users involved. We will refer to this modification of the CATT algorithm as *θ-CATT*. At the same time, the goal is to keep reporting trending topics with maximum *utility*. Maximizing the utility of the results is a competing goal with preserving privacy since the algorithm could report an empty result set and the privacy

leakage would be zero. Issues arise when the algorithm reports at least one trending topic t_i and its community C_i and for all users in $SEARCH(t_i)$ some statistical information is leaked. Especially challenging is the fact that users continuously discuss new topics which results in a constant stream of information that an attacker can use to increase their inference confidence of sensitive attribute values (as demonstrated in Figure 4.1 between cases (a) and (b)).

We now introduce a novel approach that utilizes the concept of generalization in combination with Artificial Intelligence to efficiently solve the exponentially expensive anonymization problem while preserving significant utility.

4.5.1 Utility of Trending Topics

The goal behind extracting trending topics that certain communities focus on is to provide additional insight into why certain topics end up trending, understand which user demographics are interested in an event, product, etc., and generally provide more interesting, surprising and personalized trending topics to the users of the social media service. Using the notion of Self-information from Information Theory [50] we provide a measure of the information content for community-aware trending topics. Self-information can capture how surprising an event is based on the probability of the event. The total utility of θ-CATT's results is equal to the self-information sum of every reported topic's community. The *self-information of a community* C_i is $I(C_i) = -log_2(Pr(C_i))$. Intuitively, the less likely a community is to be observed, the higher its self-information. Since we are using the logarithm with base 2, self-information is measured in bits. The prior probability of C_i can be empirically measured in the sliding window as the percentage of users that contain attribute values C_i. This metric provides a systematic way to measure the utility of the reported trending topics and can be used to calculate the informa-

tion/utility loss when anonymization is applied. We define a *utility function util*() which returns the utility over a set of tuples (t_i, C_i). Note that other metrics can be used as well without requiring alterations to θ-CATT.

4.5.2 Community Attribute Anonymization

θ-CATT needs to constantly monitor the maximum confidence of a hypothetical attacker to infer every sensitive attribute of every user in the service. When θ-CATT identifies a trending topic t_i with a homogeneous community that involves $|SEARCH(t_i)|$ users, it has to make sure that none of the users $u \in SEARCH(t_i)$ will have their sensitive attributes leaked by publishing (t_i, C_i). To ensure that, it calculates the probability of each sensitive attribute for every user u: $P(L|u.T)$ and checks if the value becomes greater than θ. If it does not, then the pair (t_i, C_i) is published. If it does, θ-CATT will anonymize the sensitive attribute of the topic's community before publishing, while preserving as much utility as possible. At the same time, the algorithm needs to ensure that its anonymization policy will not lead to a state where everything needs to be anonymized completely. which would result in a total loss of utility.

We utilize the method of *attribute generalization* to achieve anonymization similarly to k-anonymity [51]: if the city of a user can be inferred, θ-CATT reports location at the state level instead, which will alter the inference probability since a much larger population is described by this value. Generalization of categorical attributes is achieved by moving up a level in the attribute hierarchy (as described in earlier section). Depending on the depth of an attribute's hierarchy, a single generalization (moving up a single level in the attribute's value hierarchy) might lead to complete anonymization which also means zero utility for this attribute. For example, generalizing the value "male" will result to "any gender" (or "*") which does not provide any gender information.

The θ-CATT algorithm practically encapsulates the privacy-agnostic CATT algorithm which just extracts the community-aware trending topics by consuming the social stream. θ-CATT receives the batch of topics and attributes pairs (t_i, C_i) (as described in earlier section), and combined with the knowledge of every user's sensitive attributes and the topics they have previously mentioned ($u.T$), calculates if any user's privacy would leak with the publication of the specific batch. If at least one user's sensitive attribute would be exposed, then some topic communities need to be anonymized before publishing. Note that the confidence level for inferring an attribute value of a specific user can fluctuate depending on the topics mentioned by this user. For example, the attacker might believe that a user u lives in San Francisco with probability .749 (assume $\theta = .75$), but then the user mentions a topic t that is associated with a community which lives in New York. This will decrease the probability that user u lives in San Francisco and *enable* the algorithm to report a topic t' in the future, even if the community of t' is located in San Francisco. This will push again $p(L|u.T)$ closer to .75.

4.5.3 Finding the Best Anonymization Strategy

In order to output a list of trending topics that contains no privacy violations, a decision must be made that involves choosing which topic communities should be anonymized without sacrificing too much utility. There are many solutions to this problem, each with a different level of utility loss. To avoid solving this problem in exponential time by trying all possible combinations and choosing the one that minimizes the utility loss, we propose an algorithm that efficiently finds the best strategy for identifying a near optimal combination to anonymize. The θ-CATT algorithm is able to identify the privacy risk each new topic-community pair poses before publishing it, ideally in real time. To achieve this computation, θ-CATT needs to store: (1) the history of trending topics previously

reported by the algorithm, that each user u has mentioned, and (2) the communities that were reported to be correlated with those topics. This information can be stored and efficiently accessed through a hash-table data structure. Combined with the knowledge of the prior probability distributions of the sensitive attributes all the necessary information is available to calculate the inference probabilities. This way θ-CATT can simulate the behavior of an attacker and identify privacy violations early.

Batch-based Anonymization

When a batch of pairs (t_i, C_i) is reported by CATT, θ-CATT will iterate through all pairs, apply necessary anonymizations and publish the altered set of pairs. A naive approach to identify which pairs require anonymization, is to iterate through them one by one, and if a pair violates the privacy of at least one user, appropriately anonymize the community's sensitive attribute(s) before moving to the next topic. However, the iteration order might lead to non-optimal results where more communities get anonymized than necessary to preserve privacy and utility loss is not minimal. For example, it might be better to anonymize a single community C_3 instead of anonymizing two communities C_1 and C_2 and achieve the same privacy gain. Occasionally, the combination of two topic communities can enable their publication without anonymization while if we each pair is individually considered, then neither of them would get reported. For this reason, θ-CATT considers the privacy and utility of the whole batch to identify the best anonymization strategy which minimizes the required attribute generalization and utility loss.

Assume for simplicity that there is a single sensitive attribute L and let S be a batch of k pairs (t_i, C_i) with communities that have a value for attribute L. Since the generalization of an attribute in a community C_i lowers the total utility of the batch, we want to generalize L in the least possible number of communities. An *anonymized*

batch S' is a *modified* version of S with an arbitrary number of the communities in S anonymized (a community is anonymized when its attribute L is generalized at least once as described earlier). If a community does not contain a value for attribute L, it is ignored since it will not alter any user's inference probability for L. Therefore, there is a total of 2^k different anonymized batches S' ranging from the case where nothing is anonymized to the case where all k communities are anonymized and every possible combination in between. Whether a community in the batch gets anonymized or not, is encoded in a batch S' as a series of 0s and 1s (state).

The goal for θ-CATT is to find the batch S' that has greater utility than any other S'': $util(S') \geq util(S'')$ while at the same time S' preserves the privacy of every user's sensitive attribute. For example, in Table 4.1, $k = 8$ and S contains the eight topic-community pairs listed in the table. If reporting these 8 pairs violates the privacy of any of the involved users, then θ-CATT will identify an anonymized version of the batch that does not leak sensitive attributes.

A* State Encoding

To find the best anonymized batch S', a naive approach would be to enumerate all 2^k possible batches and keep the batch with the maximum utility, which at the same time does not leak any sensitive user attributes. However, this approach has exponential complexity $O(2^k)$. Instead, we propose a customized version of the A* algorithm, which is an Informed Search method [52], to identify a good batch S' *efficiently*. A* is a search algorithm, hence, it requires a search tree with a starting node and a goal node to reach. Each node of the tree is called a *state* and corresponds to a batch S'. The starting state would be the non-anynomized batch S while the goal state would be the anonymized batch S' that preserves the privacy of all involved users. There are many acceptable goal states, so additionally a cost function is needed to indicate the amount of sacrificed

utility to reach a specific state. Finally, A* requires a way to retrieve all the *neighboring* states of a given state in order to construct and traverse the search tree.

Each anonymized batch S' corresponds to a state and all possible states form the search tree. We encode S' as a k-digit binary number where the i-th digit corresponds to the pair $(t_i, C_i) \in S'$. A value of 0 as the i-th digit indicates that the sensitive community attribute L in (t_i, C_i) is generalized, while a value of 1 indicates that it is not. Ideally, we would like to report the batch S' that corresponds to the value 111...1 (no anonymization). A batch S' is an *ancestor* of batch S'' in the search tree if their encoding differs in exactly one digit, where this digit is 0 in S' and 1 in S''. Using this notion of ancestors a search tree can be defined where the encoding 111...1 is the root node and a node's children contain all descendant encodings. For example, for $k = 4$, the children of root node 1111 are: 1110, 1101, 1011, and 0111. The children of 1110 are: 1100, 1010, and 0110, etc. A visual example for $k = 3$ is shown in Figure 4.2. All search tree branches will have 00...0 as the leaf node which corresponds to a fully anonymized batch and is the least desirable result since its utility is minimal.

As the *starting state* of A* θ-CATT selects the batch S (original, non-anonymized output of the CATT algorithm) which has encoding 111...1. The *goal state* will be the first state that has no privacy leaks (all sensitive attribute inference probabilities are below θ). Given a random state S', the neighbors are generated by flipping a single digit with value 1. If there are no such digits left, the search tree has reached its end. Given that the algorithm is stable across batches (all probabilities are below θ before a new batch), there should always exist a node in the search tree that will be acceptable as a goal state. In the worst case this will be the state with encoding 00...0 at the bottom of the search tree (Figure 4.2).

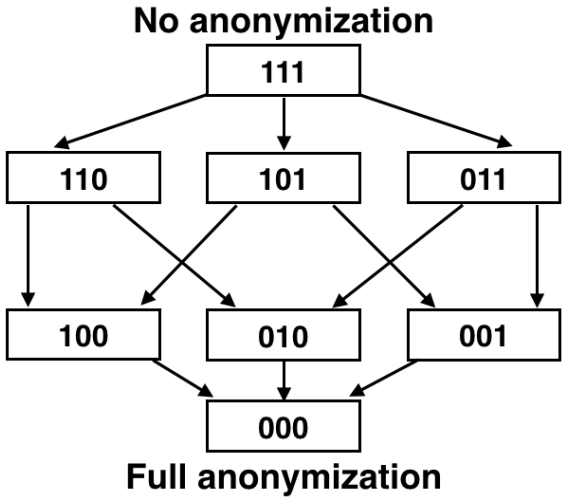


Figure 4.2: Full search tree (with $k = 3$). The "no anonymization" state is the starting state of A*.

A* Cost Function

A* requires a cost function that returns the cost of visiting each state. θ-CATT utilizes the following cost function $f(.)$: $f(S') = g(S') + h(S')$. Function $g(S')$ returns the total utility loss: $g(S') = util(S) - util(S')$, where S is the original non-anonymized set of topics and communities. Function $h(S')$ is the *heuristic* that estimates how close the current state is to the goal state and we use the following measure: $h(S') = \#$ users with a privacy violation. The number of users with a privacy violation is obtained by iterating through all the involved users in the batch and calculating the probability of inferring their sensitive attribute(s) with confidence higher than θ (equation 4.6). The function g measures the cumulative cost to reach a node in the search tree (how much utility has been sacrificed) and function h *estimates* the remaining distance of the goal state, where there is no privacy violation for any user. Note that this specific heuristic is not *admissible* (it might overestimate the cost to reach the goal state), which means that A* might not find the optimal path. Not finding the optimal path means that some additional utility might be sacrificed in order to greedily reach a goal state in less steps.

Since the two functions g and h measure different units we normalize them with two weights α and β: $f(S') = \alpha g(S') + \beta h(S')$ where $\alpha + \beta = 1$. The exact values of α and β depend on the total number of users (for g) and the specific utility function used (for h).

The heuristic $h(S')$ for an attribute L is calculated using the following formula: $h(S') = \sum_u isLeak(u, u.T \cup \{(t_i, C_i)\}, L, \theta)$, where *isLeak* is a binary function that returns 1 if the user u has an inference probability (equation 4.6) more than θ and 0 otherwise. For this probability calculation the previously mentioned topics of the user ($u.T$) are required, in addition to the new topics of the current batch (t_i, C_i). Accumulating over every user involved in the *current batch* (mentioned at least one topic t_i), $h(S')$ becomes equal to the number of violations.

Algorithmic Complexity

A* checks recursively if the current node is an acceptable goal state — number of privacy violations is equal to zero — and if it is not, it expands its children nodes and adds them in a priority queue to visit them next. Priority is calculated using the $f(.)$ function. This strategy enables θ-CATT to find a path to a batch S' that does not violate the privacy of any user, while reducing the number of necessary steps. The only trade-off is that the utility of the reached S' might not be optimal. For multiple sensitive attributes, the same process can be executed in parallel.

Let V be the set of sensitive attributes, k the size of the batch with pairs of topics and communities, T the set of all topics in the batch, and n the total number of users in the social network. The *time complexity* of the algorithm is:

$$O(|V| \cdot k \cdot |SEARCH(T)| + |SEARCH(T)| \cdot |u.T|)$$

The main bottleneck of the algorithm is the calculation of the inference probability

(Equation 4.6) for a specific attribute and every involved user. First, the whole process must be repeated for every sensitive attribute. This entails linear complexity to the number of sensitive attributes. Second, probability calculations must be repeated every time the cost of a state in the search tree is valuated. While there are 2^k states to explore, the customized A* with the proposed greedy heuristic can reach a local optimum in logarithmic complexity. $log_2(2^k) = k$, thus, the algorithm scales linearly (amortized) with the number of topics in the batch. Finally, we need to calculate probabilities for every involved user, so the time complexity will also be proportional to $|SEARCH(T)|$. The inference probability formula (Equation 4.6) contains the product of the empirical probabilities $P(t_i|L)$ where t_i is an old topic the user has mentioned and L is a sensitive attribute. To avoid calculating this product every time the inference probability is measured, we can instead store in memory the products for all topics the user has mentioned so far. The prior probability of $P(L)$ needs to be calculated only once per batch and n is a fixed number (at least in the context of a batch). The only "problematic" term is the denominator of the fraction, $|SEARCH(u.T)|$, which requires the calculation of the intersection of every set of users that mentioned the same topics with user u. However, this value needs to be calculated only once per user, per batch. Therefore, the time complexity of the inference probability calculation is constant.

The necessary *space complexity* to store the probability products for each user and sensitive attribute is: $O(n|V|)$.

4.6 Experimental Results

For our experiments we used a real Twitter dataset that contains a uniform 10% sample of the complete Twitter Firehose stream from a 39 day period between April 16 and May 24, 2014. Each tweet also contains the information of its author (user).

The extracted topics include unigrams, hashtags or capitalized entities from the tweets' raw text. The four extracted user demographics include location, gender, age, and US political party preference. Location extraction was done on (1) the tweet level using Twitter's geo-tagging mechanism, and to further improve the recall, on (2) the user level using a user-provided raw text field (similarly to [11] and [12]). To infer location based on the user's field we applied a simple but precise pattern matching process that could identify location patterns like: "City, Region, Country", or "Region, Country", or just "Country". To validate the patterns we used a Location hierarchy provided by the MaxMind database. To extract gender and age we applied existing language models extracted from [15] on social media data. The hierarchy for gender includes the leaf nodes "male"/"female" and the top level of "all genders" or "*". Similarly, the hierarchy for age includes the leaf nodes "13-18"/"19-22"/"23-29"/"30+" and the top level "*". Finally, for political party affiliation we gathered the official Twitter accounts associated with the three most popular US political parties: Democratics, Republicans, and Libertarians. Then, a user's political affiliation was determined based on the simple majority of interactions (@-replies) with these accounts (e.g. if a user mostly interacts with Democrats, their party preference was labeled as Democrat). To apply the models we gathered all the tweets of every user in the analyzed data period. While this is an expensive process it can be executed offline.

We consider all four attributes to be *sensitive* for every user. Then we ran two versions of our algorithms (simple CATT and θ-CATT) and compared the results. The algorithm settings are: $\theta = .7$ (attacker's inference confidence), $\xi = .5$ (community size as a ratio of the topic population), utility $util(\{(t_i, C_i)\}) = \sum_{i=1}^{k} I(C_i)$ (self-information sum), $\alpha = .999$, and $\beta = .001$. The selected values were empirically chosen to reflect a realistic scenario that can generate a plethora of privacy violations. Note that normally, the batch size k is not set to a specific value but depends on the data.

The average number of extracted trending topics and community pairs in the dataset is 112 per window (a window of data corresponds to a single batch of trending topics as described in earlier section). We focus on the topics that have a specific city-level location, or age, or gender, or political party preference values, which on average is $k = 21.57$ topics per batch. The per-batch average number of unique location values is 15.2, number of unique gender and political party values is 2, and number of unique age values is 2.8. The average number of involved users is 8162. The average utility without any anonymization (simple CATT) is 43.1 bits but also contains an average of 213.2 privacy violations. Privacy violations were counted by identifying users that have inference probabilities (equation 4.6) for either location, age, gender, or political party preference, that is higher than θ. To preserve the privacy of the location attribute, θ-CATT anonymized on average 4.3 communities to bring the number of privacy violations to 0. The average utility of the anonymized results published by θ-CATT is 38.37 bits, so there is a total utility loss of 4.73 bits.

Examples that demonstrate cases where a community got anonymized to preserve the involved users' privacy are listed in Table 4.2. The 4th column lists how many privacy violations would occur if the original community was published. The 5th column shows how the proposed algorithm decided to anonymize the community by generalizing at least one attribute. *After anonymization, θ-CATT managed to bring all privacy violations to 0 so that the reported results are θ-private.* For the topic #OscarTrial the location attribute was generalized to hide the location of 345 users. For the topic #ObamaInThreeWords both age and party preference are generalized to preserve the privacy of 76 users.

In Figure 4.3 it can be seen how the utility loss scales for different values of θ. As expected, when $\theta = 1$, an attacker must be 100% confident when inferring a sensitive attribute which in reality is practically impossible and results in maintaining the *full utility* of the results (equal to the utility of CATT's output). On the other end, for

Table 4.2: Real examples of communities and the corresponding anonymized versions.

Topic	Original Community	Size	Violations	Anonymized Community
#OscarTrial	Location: Johannesburg,ZA, Gender: Female	1133	345	**Location: ZA**, Gender: Female
#FreeJustina	Location: Boston, Gender: Female, Politics: Democrat	51	13	Location: Boston, **Gender: ***, Politics: Democrat
Bruins	Location: Boston, Gender: Male, Age: 19-22	196	58	**Location: ***, Gender: Male, Age: 19-22
#ObamaIn3Words	Location: USA, Age: 19-22, Gender: Male, Politics: Republican	224	76	Location: USA, **Age: ***, Gender: Male, **Politics: ***

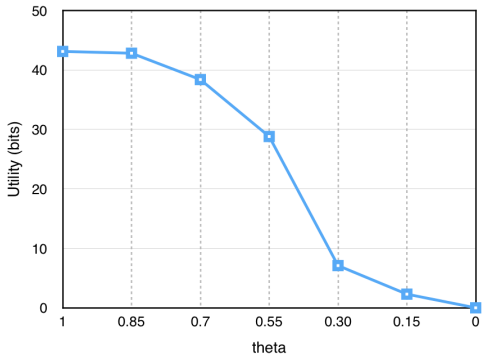


Figure 4.3: Utility loss for different values of theta.

$\theta = 0$, no information leakage is permitted at all, therefore, full anonymization of the communities is necessary and utility becomes equal to 0. These two extremes are equally not practical for a meaningful and realistic combination of trending topics with utility and preserved privacy. Based on the values in Figure 4.3 we observe that choosing a value of θ above .6 can maintain at least 73% of CATT's original utility of community-aware trending topics. This curve can be a useful guide to choose the best θ value for the desired privacy and utility trade-off.

Figure 4.4 shows the running time of our privacy preservation algorithm. All running times are recorded on a personal laptop with a 2.6GHz Intel Core i5 processor and 16Gb of RAM. There were 70 datapoints each corresponding to randomly sampled batches of topics. Since the complexity of the algorithm is mainly affected by the number of *involved users* (users mentioning one of the topics in the batch) the plots demonstrate how the running time is affected by this number. The plot in Figure 4.4 shows the execution time (y-axis) that corresponds to batches with a certain number of involved users (x-axis). Each data-point corresponds to a single batch. The number of topics with sensitive attributes (batch size) was quite stable throughout our experiments with a mean of $k = 21.57$ and a standard deviation of 3.35. The plot also contains the corresponding least-square linear trendline and its equation. All reported running times are within the range of 0 seconds (no anonymizations were necessary for these batches so A* immediately found the goal state to be the starting state) and 160 seconds. Note that the time necessary to stream-in the data of a single batch takes around 3-4 minutes based on the rate of new tweets being created on Twitter, therefore, an average running time of 39.56 seconds is more than sufficient to produce results before the new batch is even ready for processing. This means that the algorithm can be used in a real-time fashion, a strong requirement for any streaming algorithm.

To examine if the running time is affected by the size of a batch k we also performed an

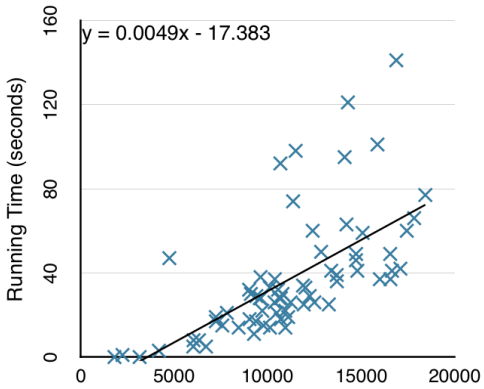


Figure 4.4: Running time as a function of the number users (x-axis), with average number of topics $k = 21.57$.

experiment where we forced the number of topics to be always equal to 15—an arbitrarily selected value that is less than 21.57—by randomly dropping some topics. We observed that the running time is also increasing *linearly* with the number of users, as expected. Altering k had no apparent effect on how the running time scales with the number of users, similar to the slope of the trendline in Figure 4.4, which proves that the greedy heuristic of A* has sublinear amortized complexity.

Generally, community-aware trending topics have less mentions than the topics that can be found on Twitter's page (curse of dimensionality). Still, based on the trendlines in Figure 4.4, we estimate that the running time for $100K$ users, which is a number that can be observed for trending topics on the Twitter web-page, would be approximately 490 seconds which is again acceptable based on the rate of generated tweets. Therefore, our algorithm satisfies the efficiency requirement of a practical real-world setting.

Finally, we tested how easily we can attack private attributes in existing Trending Topics reports. As mentioned earlier, Twitter provides Trending Topics by location (a total of 401 cities in the world). We crawled these topics through the Twitter API, and managed to infer the location of approximately 300k users that mentioned topics which were trending only in a single location just within a single day of crawling. 11.8% of

these users had their location public and sampling through them we estimated that this location inference attack was 82.33% successful. This proves how easy it would be for an attacker to exploit location-aware Trending Topics to infer the location of thousands of users. Therefore, altering trending topic algorithms to preserve the sensitive attributes of Social Media users is indeed important.

4.7 Privacy Cyborg

From the individual's perspective, a Social Media user must be mindful of which topics they discuss in order to protect themselves from such inference attacks described above. This can be particularly tedious and time consuming given the nature of social media which promotes public and frequent posting, something that usually seems harmless when considered at the level of a single post. Towards this end, we built a privacy cyborg, that can undertake the task of monitoring its owner's posts in social media and automatically warn them if necessary.

The idea of a cyborg fighting our social battles on our behalf was recently introduced by Anand Rajaraman [53]. The vision involves a local computational software resource that runs continuously (whether its owner is online or offline) and performs various tasks like protecting from online attacks, filtering out people that try to connect with malicious intend, following up on discussions, etc. Different tasks require different levels of sophistication and technology, but the cyborg in this demonstration mainly constitutes a proof of concept that targets online privacy. Similar to Privometer [40], the privacy cyborg can monitor a Twitter user's profile and what is being posted to identify potential risks of leaking sensitive information such as the user's location, race, or age. In contrast to Privometer which considers structural actions, i.e., connecting with a friend, joining a group, or liking a page, our focus is on the posted content and its role in revealing

sensitive attributes.

4.7.1 Privacy Model

The goal of the cyborg is to preserve the privacy of its owner, hence, we need to understand how sensitive attribute inference works, and implement it locally to simulate a hypothetical attack. Through this process the cyborg can identify what an attacker could infer, make a judgment call whether an attack can indeed be successful or not, and in case of the former warn the user in an appropriate way.

The Bayesian model is often used for inference attacks in the literature. We follow the same model here, and assume that the attacker can acquire knowledge that involves the correlation of topics and attributes. Based on this knowledge an attacker can identify **the most probable** value of a sensitive attribute by comparing each value's probability (e.g., 79% chance a user is a woman versus 21% chance the user is a man). More specifically, based on the individual prior probabilities of the topics T mentioned by a user, an attacker can calculate the probability $P(A|T)$ for each attribute A.

We assume that an attacker can acquire the following knowledge: The general prior probability distribution of a sensitive attribute A, $P(A)$. The observed conditional probability distribution of a topic t given the attribute A, $P(t|A)$, which can be derived from a rich trending topics report (similar to the one described in Chapter 3). With this information, the attacker can approximate the probability distribution of a user's sensitive attribute A, given the user's set of topics T: $P(A|T)$. We assume that the attacker can successfully infer the value of an attribute A if any value of $P(A|T)$ is greater than a desired threshold, e.g., 0.75.

4.7.2 The Cyborg: Supported Operations

The privacy cyborg is implemented as a daemon process that runs constantly on a local machine. While the cyborg can obviously interact with its owner when they are using their personal computer, it still needs to monitor the correlations between topics and attributes even when the owner is offline. Thus, the cyborg can perform the following two tasks: 1) Inform the user of how the public perceives them, and 2) warn the user if something they are about to post can put their sensitive attributes in danger.

More specifically, since the cyborg is practically simulating an inference attack in real time, it is able to derive a description of its owner, as perceived from their publicly posted content. This description is given as a report with a list of attributes and the corresponding probabilities for each value (e.g., male 34%, Los Angeles 56%, etc.). Since we focus on Twitter data for this demo, this task can be performed on any non private account without additional permissions. Note that these reports can be returned on demand or triggered when something changes. Due to the nature of the inference process, probabilities can change without the user posting anything—could be the result of a population shift for a topic of interest.

However, since the publication of new topics remains the most effective way for these probabilities ($P(A|T)$) to change the cyborg can proof-read a new post and inform them whether any sensitive attributes will be compromised, i.e., an attacker will be able to successfully infer them following the publication of the post.

4.7.3 The Cyborg: Technical Components

The cyborg needs access to the same knowledge as the attacker it tries to simulate. Namely, access to the historical tweets of its owner, access to reports of topics and the correlated attribute values with specific percentages, the prior general probability

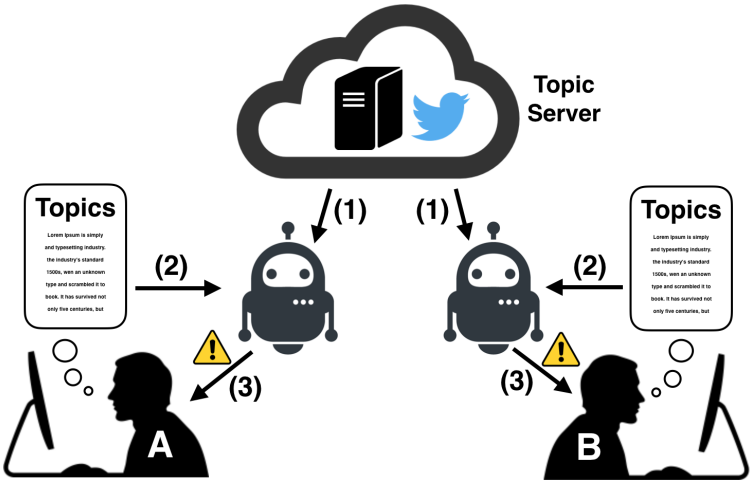


Figure 4.5: Visualization of the cyborg, related actions, and components. Note that each owner/user has their own cyborg.

distributions of the attributes, and the user-provided settings for the privacy threshold (the attribute inference probability is high enough to pose a privacy risk).

Figure 4.5 is a visualization of how the cyborg works. The cyborg owners and social media users (bottom left and right) want to post content online that contains a growing list of topics (action 2). This list goes through the cyborg for proof-checking. In the background, the cyborg consumes information from the social media service (action 1) and monitors the topics of the user (both old and new). With this acquired knowledge the cyborg can warn its owner when a sensitive attribute is at risk (action 3).

The cyborg itself comprises of a daemon running on a local computer with Internet access so it can acquire and maintain the necessary data for calculating the attribute probability distribution $P(A|T)$. Interactions with the owner are supported though a graphical UI where the proof checking and warning displaying can be performed. The prior probability distributions and correlated topics/attributes are provided by an external server (the cloud in Figure 4.5) and are assumed to be public knowledge, similar to how trending topics on Twitter are public. This server constantly collects tweets from the Twitter Streaming API, and computes in real time the correlation between topics with

specific attribute values. To identify the attribute values of other users, a periodic job is executed on the server that infers the attributes of any other user in the social stream (detailed in Chapter 3). This information is then used to calculate the percentages of association between attribute values and topics in the stream.

A demonstration of a working cyborg prototype can be found in this video: `https://youtu.be/PfzC39i9nbg`

Chapter 5

A Social Sensor Application: Mining Complains for Traffic-Jam Estimation

5.1 Motivation

In the current chapter's experiment we focus on the fact that the sentimental state or mood of the analyzed population (in the context of social sensors and event discovery) is seldom taken into consideration. Most algorithms measure the levels of a disaster or the magnitude of an event as a simple function of the corresponding social media discussion volume. This simple function can be anything from a linear model to an exponential distribution. But what is often ignored is the state of the people that participate in the online discussion. For example, an overly enthusiastic crowd might give a false idea of the size of a political demonstration. A shy demographic might lead to the perception that a specific music trend is not as popular as it really is. People complaining about their jobs during a very hot day might give the false sense they are generally unhappy

with their work environment. To avoid arriving to such false conclusions based on online social signals, a better understanding is needed of when people publish content on social media, what emotional state they are in, and which factors might have led them there.

For our experiments, a specific user behavioral pattern was examined: complaining in social media while stuck in traffic jams. We combined two large publicly available datasets, one for traffic in California and one for Twitter content, to study how car drivers react in social media while driving during increased traffic congestion. Driving a car is already known to be a stressful activity for many and things can be much worse during traffic jams; frustration and boredom may lead drivers to make irrational decisions or act irrationally due to anger. Unfortunately, such behavior can increase traffic congestion, be dangerous, and cause accidents. Social Media have already been utilized for some time now to help with traffic de-congestion. From specialized social media apps like Waze [54] - a crowd-sourced community that monitors traffic, accidents and other events in real time - to regular use of Twitter to automatically or manually publish reports and alerts of the street conditions [55]. The purpose of such information tools is for drivers to inform themselves about traffic conditions before getting in their car and make the necessary choices to optimize their commuting route and time. In reality, a non trivial amount of smartphone owners are observed to use their handheld devices while driving, despite laws that render the use of handheld devices by drivers for texting purposes illegal, for obvious safety reasons [56].

Apart from getting informed about traffic, users resort to social media to also complain or update their Twitter/Facebook status about being stuck in traffic. Most frequently, such status updates include humorous remarks, swearing, frustration, and the occasional warning about traffic congestion on specific freeways (for others to see). Some Twitter users state humorously that the best time to tweet is during rush hour traffic, or that the 405 freeway is the only freeway where there's enough traffic to stop and tweet

about traffic (I-405 is a freeway in Los Angeles, California). We use this signal as a social sensor to model the circumstances and traffic conditions, and how the drivers' frustration may have an impact on the observed social discussion volume.

Indeed, we discovered that social reaction fluctuates in a non trivial manner. Different circumstances lead to different volumes of complaining about the traffic severity instead of following a strictly linear correlation. And while in many cases correlation is not equal to causality, for this particular experiment, the observed correlation between the real world (traffic) and the social reaction (tweets) is actually a causal relationship. The measured social reaction - tweets made by drivers stuck in traffic - is strictly caused by traffic congestion and the two variables are strictly dependent.

In Section 5.2 we list the related literature, in Section 5.3 we describe the used datasets, in Section 5.4 we provide the regression analysis, in Section 5.5 we offer the correlation findings between sentiment and traffic complaining, and finally, in Section 5.6 we compare the new regression model with baselines.

5.2 Related Work on Social Sensors and Traffic Analysis

There are two research fields related to the subject of the current work: 1) Studying and modeling of Traffic Congestion and 2) Social sensors utilized on online social media to mine information about physical events.

Traffic Analysis: There has been a lot of work and many studies that focus on the general analysis of traffic. They deal with questions like: How does traffic correlate with urbanization and economic growth? What causes traffic when there is no apparent reason? How does human behavior contribute in traffic congestion?

Traffic is studied in a plethora of areas including: (a) Financial/Political: measuring urban growth [57], (b) Psychological: measuring human behavior, DUIs etc. [58], [59], (c) Transportation: improving roadway conditions [60], and (d) Mathematics/Statistics: modeling traffic using statistical and mathematical frameworks [61].

Online Social Sensors: Social sensors and the discovery of what is happening in the real world through social media is a well studied area. Kryvasheyeu et al. [62] examine how social sensors performed during hurricane Sandy (disaster control), García-Herranz et al. [63] utilized the social friendship network to quickly detect viral diseases, Zhaoet al. [64] use social media content to discover physical events in real time with a focus on sports events, and finally, Aggarwal et al. [65] wrote a book chapter that describes the current developments and challenges on social sensing in the context of data mining.

Studies that focus on social sensors specifically for the improvement of traffic reporting are closer to the problem tackled in our work: [66], [67], [68], [69], [70], [71]. To the best of our knowledge there are only a couple publications that utilize crowd-sourced data to improve traffic prediction or identify traffic anomalies. In an ongoing Microsoft Research project [71] researchers try to combine the vast amount of historical data (both social and traffic) to create a single model for traffic prediction. Both works from Daly et al. [69] and Ribeiro et al. [70] mine the social sphere to identify and explain traffic conditions and events. Jingrui He et al. [66] propose a way to improve traffic prediction, by combining social data from Twitter and historical traffic data. The authors use a raw, but localized, tweet stream to discover the users' future destinations and combine it with historical traffic data to produce a near-term (5 minutes to 1 hour) traffic prediction. The results show an improvement of the mean absolute percentage rate by almost 2% from the baseline model that only utilizes historical traffic data. Such approaches can be much improved with a more fine grained modeling that improves the correlation between social volume and traffic congestion. We propose such a model and show in Section 5.6 how this

kind of traffic prediction can be potentially improved. Our work is different in the sense that we model traffic jams purely through social sensors without any knowledge of the traffic's historical distribution. Also, we examine which factors, like mood or sentiment, may lead to specific patterns of social reaction.

Pan et al. [72] propose a system for monitoring traffic via mobile signals in order to identify anomalies in the usual traffic flow. This is a more precise approach to discovering traffic anomalies but is not directly applicable to our setup since the information is not always publicly available (crowd-sourced) as it is on Twitter or other social media. Finally, [67] appears to be the only work that studies the correlation between social volume and traffic, at different hours of the day, but does not offer a model that captures their observations. To the best of our knowledge, all models in the mentioned publications ignore latent social factors that could skew the social volume related to traffic.

In [73] we proposed a new regression model for the estimation of traffic congestion purely using social signals – more specifically complaints that drivers post on Twitter while stuck in traffic jam. This new model is based on the observation that drivers complain differently during different hours of the day. In the current work we further correlate the observed fluctuations of complaints with the general mood of Twitter users. This correlation explains why people behave differently (complain more or less) for similar levels of traffic congestion and increases our understanding and trust for the originally proposed model. Based on this correlation we reach a more concrete conclusion, that the sentiment of the studied population should always be attributed in social sensor applications, otherwise a simple approach might sacrifice quality and accuracy of the results.

5.3 Data Model

Towards building a regression model for traffic congestion that only utilizes social signals, two different datasets are required: The social dataset that contains signals of traffic congestion and the dataset with the ground truth of the traffic congestion volume. For the social dataset we used Twitter data since drivers like to complain about traffic there. For the ground truth (traffic jams) we use official sensor data from the California Department of Transportation.

5.3.1 California Traffic Data

The first step towards a combined traffic and social analysis is to obtain the necessary traffic congestion information and establish the ground truth. We focused on the area of California where the Department of Transportation (CALTRANS) collects a wide range of traffic statistics and publishes them online on the PEMS website [74]. CALTRANS maintains a plethora of physical stations known as Vehicle Detector Stations (VDS) on freeways across the state of California. Many sparsely inhabited areas have no stations but most metropolitan areas like Los Angeles, San Francisco and San Diego are very well monitored. Each VDS is located next to a freeway and reports data like lane occupancy (if there are more than one lanes), speed in each lane, and health status, with a frequency of 5 minutes. For the purposes of this analysis, we did not use the raw data from the VDS stations since the PEMS website does not provide a programmatic way to download data for many stations. Instead, PEMS computes and reports all traffic bottlenecks on a daily basis, so we used these reported stats. **Definition:** A **traffic bottleneck** occurs where the traffic demand exceeds the available capacity of the roadway facility.

More specifically, a bottleneck between two station detectors on the same freeway is observed under the following conditions:

- There is a speed drop of at least 20 mph (32 Km/h).
- The overall speed is less than 40 mph (64 Km/h).
- The distance between the two stations (minimum extent of a traffic jam) is at least 3 miles (4.8 km).
- The speed drop is observed for at least 70% of a 35 minute duration.

Note that these conditions have been chosen by CALTRANS. It is beyond the scope of this work to validate the above numbers, conditions, and semantics of traffic congestion. Since we are using the same definition across the whole analysis, there is no bias that could skew our observations.

For each analyzed day, the full list of all reported bottlenecks in California is obtained. Each bottleneck consists of a location (VDS latitude and longitude), extent, duration, and delay. Extent is the distance, in miles, of the reported traffic jam. Delay is the total duration, in minutes, of the congestion. Finally, delay is an artificial composite metric that describes the total loss of time due to the bottleneck and is measured in "vehicle-hours":

$$TotalDelay = N \times extent \times duration \times \left(\frac{1}{speed} - \frac{1}{35}\right) \quad (1)$$

where N is the total number of cars affected by the congestion and speed is the reported speed during a bottleneck. Note that this is a simplified version of the total delay formula [75]; PEMS is actually using the non publicly available knowledge of each lane's occupancy and corresponding speeds to increase the accuracy of the delay computation. In any case, due to the nature of this formula to combine all the other metrics (extent, speed, duration) as well as the total number of affected drivers, it is

Day of the Week	Number of Bottlenecks
Monday	1948.72
Tuesday	2321.04
Wednesday	2418.96
Thursday	2583.38
Friday	2481.80
Saturday	1069.1
Sunday	642.88

Table 5.1: Average number of traffic bottlenecks in California per day of the week throughout a period of 6 months in 2014 (May - October). The observed trend is that the reported number of bottlenecks increases towards the end of the week and is significantly lower during the weekends when most people do not commute to work. It is interesting to note that Monday's traffic appears to be quite lower than the rest of the week even after removing holidays that could cause a possible bias.

commonly used by traffic analysts [76], [77] as the indicator of how severe a traffic jam is. We will also be referring to it as "traffic volume" or "bottleneck severity".

One drawback of the PEMS-generated bottleneck report is that it does not provide an accurate time for each bottleneck (only the exact location). Instead, CALTRANS provides a low granularity time attribute named "shift" which takes the values AM, PM, and NOON. Therefore, bottlenecks can only be studied on a shift basis, which for the purposes of our paper is enough as shown later on. The AM shift includes the hours between 5am and 10am, the NOON shift between 10am and 3pm, and the PM shift between 3pm and 8pm. Bottlenecks that occur during the night or after hours are not reported and based on the raw traffic data, traffic-jams during those hours are extremely rare and would not be useful for a statistical analysis. As shown in Section 5.4, very low traffic periods may occur even during the day, especially during weekend mornings or national holidays.

Daily traffic data was collected for every day within the period from May 2014 to October 2014. Table 5.1 shows the average number of reported bottlenecks for each day of the week. The relatively high number of unique traffic jams can be explained by the

fact that some bottlenecks might occur in very close locations and due to CALTRAN's bottleneck definition get captured as individual traffic jams. For this reason the number of incidents is not used to measure the severity of traffic on a freeway but instead we use the total delay formula in Equation (1). In order to match traffic jams with a physical location, we use the corresponding VDS station that observed each bottleneck, to identify the county/city and more importantly the exact freeway the station is measuring. Through the freeway name and number (e.g. US-101) we can then process the social data and collect tweets that correspond to a specific freeway's traffic jam.

Figure 5.1: Greater Los Angeles Area and freeway I-405 highlighted red. Due to its strategic position and size, I-405 is heavily used from Los Angeles residents.

5.3.2 Social Data

We use Twitter as the social sensor platform to study traffic jams. To obtain the necessary data we used the Streaming API [78]. Another explored alternative was the use of the Search API [79] which however does not provide any guarantees on the distribution or the completeness of the search results and therefore introduced statistical bias.

Using the streaming API, however, while guaranteeing completeness, has two draw-

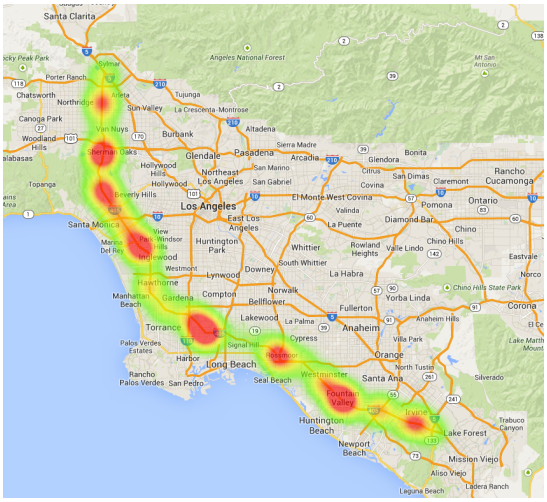


Figure 5.2: Traffic severity heatmap for I-405. The heatmap shows the major congestion points during the evening hours, averaged across all weekdays.

backs when compared to the search API. First, one can only collect new data and there is no historical data access. Second and most important, the streaming API does not support geo-enabled queries in a form that would be helpful to the current analysis. One may query for all tweets from California OR all tweets about traffic, but not their intersection. Alternatives exist, like collecting all tweets from Los Angeles and separately all tweets about traffic and then join them but due to the rate-limiting imposed by Twitter it would not be feasible to get all tweets from Los Angeles, given the large number of Twitter users living there. Not having the ability to filter tweets by location led us to collect any tweet that mentions the keyword "traffic" and then proceed to filter down the collected tweets using other heuristics. Specifically, only the tweets that mention the freeway name get under consideration, tweets from automated or traffic reporting accounts (like police departments and radio stations) are removed, and finally, human judges manually go through all remaining tweets and keep only those that were made by people stuck in traffic. The last step is consistently performed using basic rules like: tweet text contains phrases with temporal hints like "this traffic" or "on my way", tweet

contains a picture of other cars in traffic jam taken from inside a car, tweet contains a self-taken picture of the driver (selfie).

The last filtering step to keep only tweets from people that drive in traffic, is the only one that needs human assistance to complete. It is still the most error prone step, since Twitter users will not always be explicit about being behind the wheel while tweeting. It's important to note here that interacting with a (smart)phone for purposes like texting, checking social media, tweeting etc. while driving, even during stand-still traffic, is considered illegal in California [56]. However, this does not discourage people from posting selfies (self-portrait photographs) on Instagram, or tweeting about the annoying traffic. Still, the fact that such actions are deemed illegal makes it an interesting signal to study.

Due to privacy reasons, the social data used in this study was anonymized, especially since as stated above, there are legal issues involved when tweeting while driving. It should be noted here that the processed social postings (publicly available tweets) are made by Twitter users with non private accounts and are openly provided by Twitter through the streaming API. However, to satisfy privacy and ethical concerns, we are not publishing any names, usernames, or content that could lead to the identification of specific users.

The final product of the social data collection is a set of tweets (including all meta-data provided by Twitter), grouped by date and shift (AM, NOON, PM), made by people while stuck in traffic jams. In the rare cases where a user made more than one tweets during a specific time period we counted only one of them. We will be referring to the number of tweets as "social volume" in this analysis. Note that the number of tweets is fairly low (usually less than 20 per shift of the day), these numbers are very consistent and stable throughout the whole analyzed period which greatly reduces the likelihood of random bias.

5.3.3 I-405 Freeway

Given the mentioned limitations posed by the collection of social data, we focus on one major freeway, infamous for its devastating traffic jams: San Diego Freeway I-405. I-405 (Figure 5.1), founded on 1964, has a length of 72 miles, passes through the whole city of Los Angeles and is used by hundreds of thousands drivers daily and there is always stand-still traffic reported during rush hours. People even call it the "monster" [80] as a humoristic acknowledgment of its size and severe traffic. The heatmap in Figure 5.2 shows how traffic is distributed across the freeway; traffic congestion is not evenly distributed but instead there are some specific points where traffic jams mostly occur, which makes traffic at these points even more severe during rush hour. We chose I-405 over US-101 (another popular candidate) because it is limited in the area of Los Angeles while US-101 covers the whole west coast of the United States. However, we made sure that the traffic patterns observed in I-405 are not unique. The traffic volume between the two freeways was compared and we found that they follow the exact same patterns for all days of the week and all shifts of the day. Therefore, it is safe to say that the choice of I-405 does not introduce any freeway-specific traffic anomalies.

5.3.4 Tweets from Drivers

As explained in subsection 5.3.2 only tweets made by people driving during traffic jams are counted, instead of every tweet mentioning traffic and the freeway name. Utilizing the latter as the social volume, would introduce cases where the raw volume of noisy tweets is misleading for estimating the actual traffic. There are two categories of "noisy" tweets. First, there are tweets made by automated accounts (e.g. police dispatch, highway patrol) or news agencies that report traffic on Twitter [55]. Such tweets are published whenever traffic bottlenecks occur and are usually agnostic of the exact severity of the bottleneck

or how much it really annoys the drivers. The second category consists of tweets that are potentially about traffic, posted by normal users, but not during their commute. The problem posed by both categories is that those tweets are not part of a direct social reaction to a traffic jam. Any traffic jam estimation that utilizes those tweets would introduce excessive noise and predictive bias. As an example of a case where the raw volume of all tweets is misleading, on Friday the 23rd of March 2014 a new carpool lane opened for freeway I-405 which caused an abnormally high volume of discussion among Twitter users. Most of this discussion included chatter about the potential usefulness of this new lane or excitement about it. On another similar case, a celebrity Twitter user made a tweet about being stuck on traffic which triggered many replies from fans and followers. In both cases, any conclusions or traffic modeling based on the generated "social reaction" will be very biased unless the data is correctly processed and filtered.

In Section 5.6 we compare the traffic regression error between a model that uses tweets only from drivers and a model that uses tweets from every normal Twitter user that talks about traffic (automated accounts, news stations, and bots are still removed). We show that focusing purely on tweets from drivers actually increases the accuracy of the regression.

5.4 Analysis

The purpose of this analysis is to discover hidden features that would more accurately estimate the magnitude of traffic congestion through social media. Our basic assumption is that there are cases where the size of a social media event may be different from how humans perceive it. Perception is a complicated process and there are many factors that play a role (e.g. mood, enthusiasm, weather, family status, political beliefs, etc). We assume that complaining about traffic falls under the umbrella of such events and

study the correlation between traffic and complains to show that indeed there are other latent factors that contribute in non-trivial fluctuations of the social reaction volume. Traffic jams are measured with high accuracy by automated traffic monitoring stations but human perception of a bottleneck may vary under different circumstances. To the best of our knowledge this is the first work to study how fluctuations, potentially due to psychological factors like mood or sentiment, can improve the accuracy of a social sensor.

To describe this analysis, first some basic statistics are provided, then a baseline model for regression is offered that will be used for comparison to our proposed model, and finally, the analysis of traffic congestion by time of the day will be described, based on which we build the proposed regression model.

5.4.1 Basic Data Statistics

To begin the analysis, a better understanding of the two datasets (traffic volume and social volume) is necessary. As mentioned at the end of Subsection 5.3.2 our analysis is focused on the California freeway I-405. Table 5.2 and Figure 5.3 show the traffic volume on I-405, by day of the week and shift of the day. Only weekdays are shown since traffic congestion during weekends is extremely low. Note again that these numbers describe the total delay and not the amount of cars traveling. Close to zero traffic volume in our context means that there is no introduced delay since the cars in the freeway are running at a speed close to the limit and not that there is no traffic at all. Our proposed model works for weekends as well but they are omitted from the current analysis for simplicity. The reader can view the actual statistics for weekends in Tables 5.2 and 5.3. These tables also provide the standard deviation for each average.

The first observation based on the traffic volume data is a clear traffic increase towards the end of the day (PM). Also, for every weekday, the morning and noon traffic fluctuate

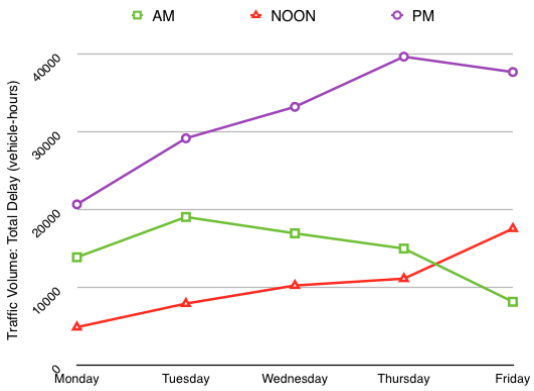


Figure 5.3: Traffic averages for each day of the week and shift of the day. The general trend for most of the weekdays (no weekend) is that PM traffic is always worse than AM and NOON and AM is worse than NOON except on Fridays.

Day	AM mean	AM stdev	NOON mean	NOON stdev	PM mean	PM stdev
Mon	16840.25	2927.09	3462.38	1271.59	21234.75	3234.97
Tue	18747.29	1907.23	5299.43	3212.63	27126.57	3705.78
Wed	19708.20	2741.86	5451.60	2473.53	34484.80	2725.18
Thu	19167.00	3225.76	7585.11	1764.80	40134.67	4830.19
Fri	11997.67	2857.14	13364.78	2270.96	41370.00	3769.05
Sat	200.25	50.77	7038.50	2332.95	9759.00	2767.73
Sun	54.50	91.71	2893.25	1231.31	3020.25	907.63

Table 5.2: Traffic volume (total delay) statistics for I-405 (Los Angeles) by day of the week. To measure traffic volume we sum up the Total Delay of each reported bottleneck across I-405 during each day's shift.

far less than the evening's. The second observation is that evening traffic gets worse towards the end of the week (Thursday and Friday) as can be seen in Figure 5.3. There are many potential explanations of why these patterns occur. Arguably, the reasons why most people drive during rush hours are work related. Therefore, most patterns could be explained by how people schedule their work hours. For example, it could be that during Fridays people tend to leave earlier from their work and cause a more concentrated traffic congestion around 4pm and 5pm. Regardless of the reason, the fact remains that traffic volume is higher during evenings and towards the end of the week, and lower at noon and in the mornings.

Day	AM mean	AM stdev	NOON mean	NOON stdev	PM mean	PM stdev
Mon	5.00	1.51	2.88	1.64	7.12	2.47
Tue	5.71	2.36	4.71	3.64	8.14	2.41
Wed	6.60	2.51	3.40	2.30	12.40	2.19
Thu	5.78	2.54	4.89	1.69	15.11	3.41
Fri	3.33	2.06	6.78	3.07	15.78	5.63
Sat	1.25	1.04	4.88	2.36	5.38	4.24
Sun	0.25	0.71	2.00	0.93	1.00	1.07

Table 5.3: Social volume (number of tweets) statistics. These only include tweets made by drivers stuck on I-405 traffic jams.

Similar to the traffic volume, Table 5.3 and Figure 5.4 show statistics about the social volume (number of tweets), again on a day-of-the-week and shift-of-the-day basis.

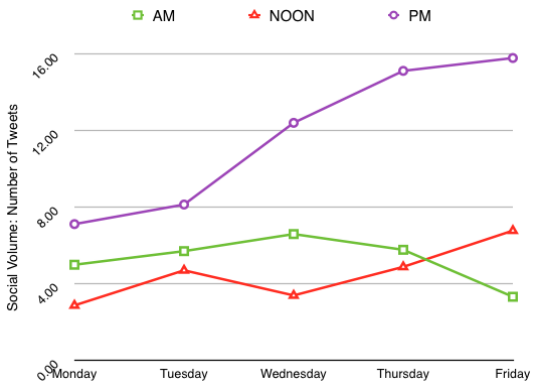


Figure 5.4: Social volume averages by day of the week and shift of the day. The general trend of social reaction appears to be in sync with the traffic volume (if compared with the plots in Figure 5.3.

The social volume statistics confirm our intuition that social reaction is proportional to the traffic volume. Same as with the traffic, during morning and noon hours social volume is generally low across all days of the week but peaks up during the evening hours. Also, the evening social volume becomes higher towards the end of the week (Thursday and Friday).

5.4.2 Naive Approach: Linear Model

From the basic statistics we listed in Subsection 5.4.1 it would be reasonable to expect a linear relation between traffic volume and social volume. It makes absolute sense that social reaction becomes stronger when traffic jam conditions worsen. Based on this hypothesis we can use linear (least squares) regression to compute a linear model that can estimate traffic based on the number of generated tweets (a typical example of social sensors). Figure 5.5 depicts the linear model as a straight line:

$$TrafficVolume = 1850.0 \times SocialVolume + 5299.1$$

Note that the model's coefficient of determination (R^2) is 0.6597 which can be considered high depending on the application and desired level of regression precision. We list in Section 5.6 the absolute and relative errors yielded by this model when trying to estimate (predict) traffic.

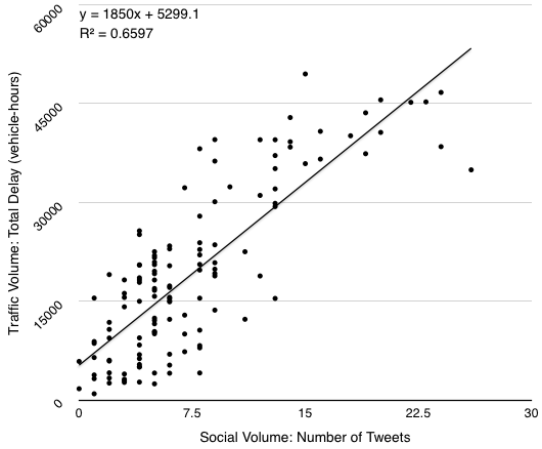


Figure 5.5: Plot of traffic volume vs social volume. Each point describes the data of a single day and shift. The x-axis measures the social volume (number of tweets) and the y-axis measures the traffic volume as total delay (vehicle-hours). We can fit a linear model with R^2 value of .6597.

We also tried to fit a second degree polynomial model to the data. The result was a minor improvement of the R^2 value but unfortunately, due to physical limits, greater traffic volume values that could validate a polynomial model do not exist. Without loss of generality or introducing any bias for further findings, we assume a linear fit for the purpose of this study.

While the linear model appears to be relatively accurate, certain underlying patterns exist, which are ignored. Plotting the same data from Figure 5.5 and grouping datapoints by shift of the day in Figure 5.6, makes it clear that each group has its own characteristics and behavior. The conclusion from this observation is that latent features might describe the connection between traffic and social reaction in a better way. In Section 5.5 we explore the general sentiment or mood as a possible connecting factor. In any case, this conclusion lead us to the hypothesis that a different model that exploits such patterns could fit better than the naive linear model.

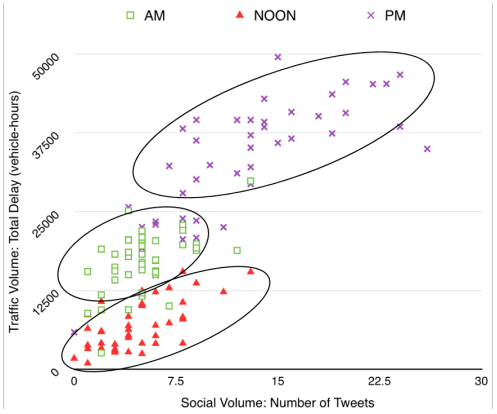


Figure 5.6: Same datapoints from Figure 5.5 but grouped by shift of the day. PM datapoints are mostly located on the upper-right, AM datapoints at the center-left, and NOON datapoints at the lower-left.

5.4.3 Analysis by Time of the Day

To evaluate whether traffic is perceived differently under different circumstances we computed the ratio of Traffic Volume over Social Volume for different times of the day (averaged across all weekdays). We also tried to explore correlations with the day of the week or the weather (temperature) but the time of the day proved to be by far the strongest feature. The ratio of traffic volume over social volume measures how much drivers complain per traffic delay and lower values indicate higher complaining. Note that due to the characteristics of the traffic jam dataset, the analysis is performed on a shift basis (AM shift: 5AM-10AM, NOON shift: 10AM-3PM, PM shift: 3PM-8PM). A plot of these ratios can be found in Figure 5.7. On the right-most column of the chart, the average ratios across all weekdays are shown. The lower a value of a ratio is, the more drivers complained on Twitter about traffic. So it is evident through these ratios that morning social reaction to traffic appears to be the lightest as if people do not care as much. On the other hand, noon reaction is the heaviest while NOON traffic is the lightest as seen in Figure 5.3. This indicates that humans react to traffic congestion differently based on the hour of the day and just measuring the raw volume of social complains regardless of what time it is will be misleading.

Through these ratios the conclusion is made that a different time of the day indeed results in different levels of traffic reaction. In Figure 5.8 the datapoints are plotted based on the shift (AM, NOON, and PM). We can then fit individual models on each subset of the data. In Figure 5.8 the linear models are plotted using least squares regression. As with the naive liner model (subsection 5.4.2), we also tried to fit other models (polynomial, exponential) but the linear yields the best results even if not all individual R^2 values are high enough. The 3 individual sub-models for each shift of the day and the corresponding R^2 values are listed in Table 5.4.

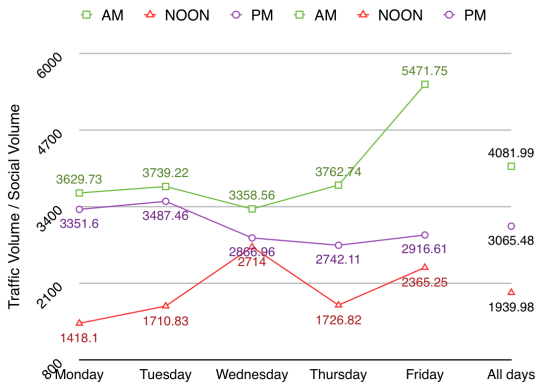


Figure 5.7: Traffic volume / Social volume ratios. Lower values indicate heavier social reaction. Morning social reaction to traffic appears to be the lightest while noon reaction is the heaviest. Humans react to traffic congestion differently based on the hour of the day. Even though NOON traffic is the lightest (Figure 5.3) it causes the most severe social reaction.

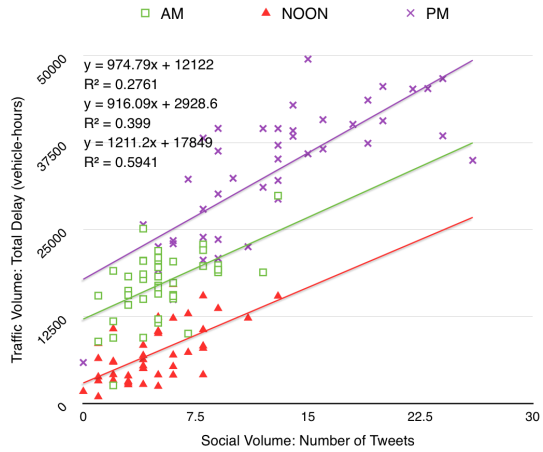


Figure 5.8: Shift-based linear model: A mixture of 3 different linear models, one for each shift of the day (AM, NOON, PM).

Note that the individual R^2 values for each shift are lower than the R^2 value of the linear model (which is 0.66). While this could be interpreted as a bad fit of the proposed model to the data, we will show in our experimental analysis in Section 5.6 how the proposed model compares to the naive linear model and other baselines, when used in the context of estimating traffic through social volume. Generally, R^2 values are not always the best indicator of well-fitness and in cases where residuals form specific

Shift of the Day	Social (SV) to Traffic (TV) Model	R^2
AM	$TV = 974.79 \cdot SV + 12122$	0.2761
NOON	$TV = 916.09 \cdot SV + 2928.6$	0.3990
PM	$TV = 1211.2 \cdot SV + 17849$	0.5941

Table 5.4: Social-to-Traffic Modeling, by shift of the day (shift-based model).

patterns, can be misleading. In any case, the actual superiority of our model will be shown through its regression accuracy.

5.5 Sentiment Correlation

There is strong evidence that human psychology plays a significant factor in traffic jams. It has been proven through experiments that in many cases human driving behavior can be the single cause for traffic congestion since people are unable to keep a steady speed which causes traffic waves [81]. This tight connection between human behavior and traffic jams triggered the idea that social reaction might also be affected by psychological factors. As described in Section 5.3.4 we measure the tweets from actual drivers which mostly involve complains about the traffic. According to psychologists, the act of complaining is actually beneficial for humans since it relieves stress and makes them feel better [82]. Also, people tend to complain more when they are less happy which led us to study the correlation of sentiment and social reaction for traffic.

Golder et al. [83] have published an extended study where they measured the sentiment of English speaking Twitter users in many countries with the USA among them. Their main observation is that positive and negative sentiment fluctuate throughout the day. The authors also show findings where positive sentiment reaches higher values during the summer (more daylight) and the opposite during the winter. Our main takeaway from Golder's paper is that sentiment fluctuates throughout the day which could also translate to different levels of complaints. Figure 5.9 shows how sentiment fluctuates

during each weekday and hour. The general trend appears to be that people are the happiest early in the morning, (around 7-8), then become temporarily less happy during the afternoon, and finally sentiment gets back to higher values during the evening. This trend correlates with the 8-hour working schedule of most people which would mean that they are less happy at the middle of their work schedule.

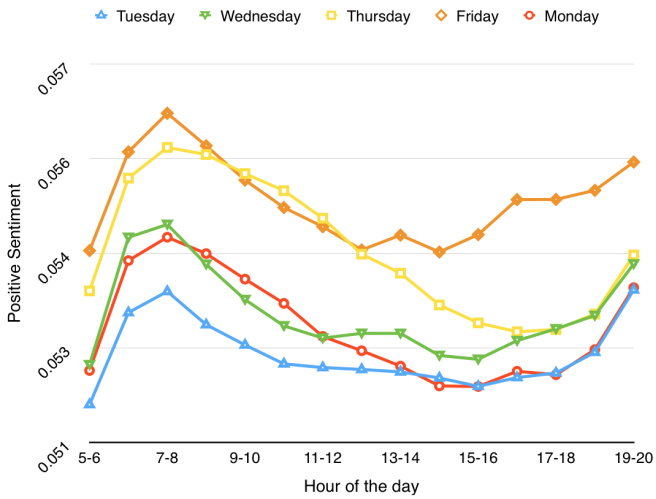


Figure 5.9: Plot of the positive sentiment values by time of the day, for each day of the week. Measurements describe the english speaking US population and are provided by Golder et al. [83].

To offer a potential explanation of the observed behavior of drivers (different levels of complaining during different times of the day) we further analyzed the positive sentiment index provided by Golder et al. (shown in Figure 5.9). In the bar-chart part of Figure 5.10 the average positive sentiment across all weekdays is plotted (from Figure 5.9). All timeslots of the same shift are then grouped together and average positive sentiment across each shift is computed (left chart in Figure 5.11). This grouping shows that NOON and PM shifts have the same average of positive sentiment. However, there is an important difference: the rate of change of sentiment. As can be seen from the derivative of the sentiment (line-chart part of Figure 5.10), during NOON the positive sentiment decreases while during the PM shift the positive sentiment increases (right

chart in Figure 5.11). This indicates a different rate of mood change: during NOON people are in a mood to become less happy while during the evening they are in the mode of gaining their original positive sentiment.

Therefore, people are in different moods during each shift (NOON: they grow less happy, PM: they grow happier, AM: they are the happiest) which correlates with our plotted ratios between traffic and social volume from Figure 5.7: stronger complains during NOON, weaker complains during PM, and the weakest complains during AM even though the actual traffic volume is lowest at NOON.

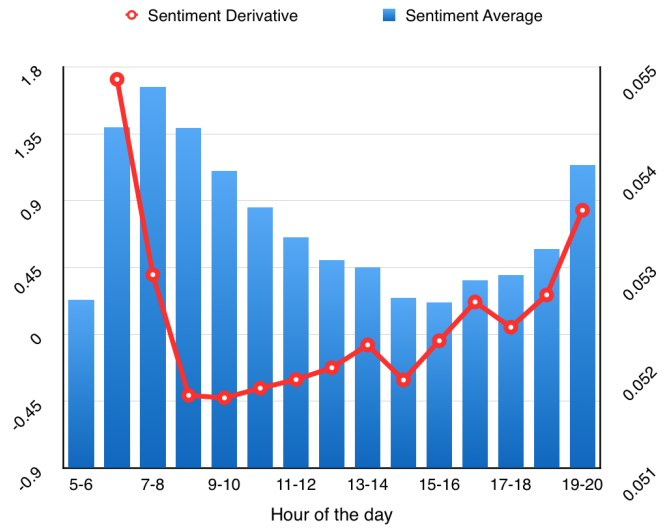


Figure 5.10: Average positive sentiment bar-chart for weekdays, by hour of the day. The red line shows the derivative of positive sentiment. Positive sentiment has the highest value in the morning, reaches a local minimum in the afternoon and regains high values in the evening.

The importance of this finding is two-fold. First, it offers a possible insight on the driver's cognitive state when stuck on traffic. While we already expected that people will get more frustrated with increased traffic congestion we show that less obvious factors like the time of the day or the day of the week can make things even worse. Second, we can see that any statistical models that utilize and associate social data with traffic prediction or analysis need to attribute more factors than just the volume of social

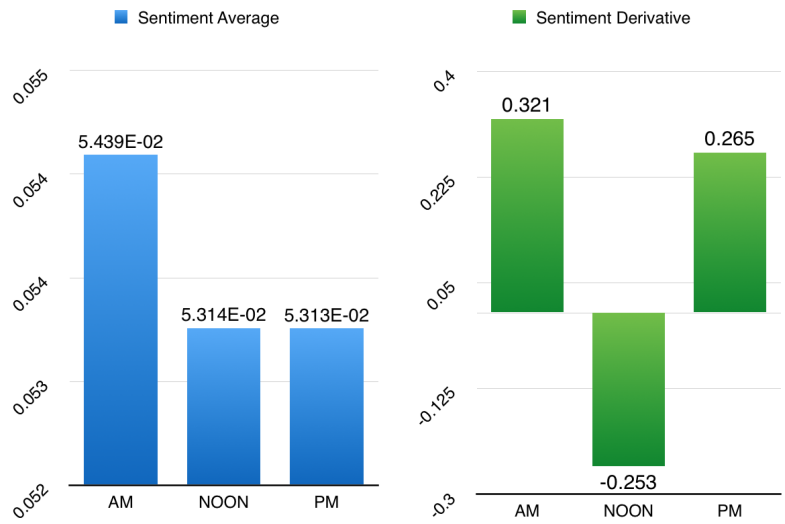


Figure 5.11: Left: Average positive sentiment bar-chart, grouped by shift (AM, NOON, PM). Right: Average value of the derivative, by shift. While the average positive sentiment is almost the same during NOON and PM, the sentiment derivative has different signs which indicates a different change rate: during NOON people are in a mood to become less happy while during the evening they are in the mode of gaining their original positive sentiment.

postings. However, correlation between sentiment and level of complaining does not necessarily equal causality. The above analysis is only offered as a potential explanation. Also note, that for each different shift, drivers have different destinations: At AM it is mostly work, at NOON it should be lunch, and at PM it is mostly getting back home. But regardless of which factor leads to social-reaction fluctuations, the shift-based model is able to capture this different behavior and, as will be shown in Section 5.6, improve the accuracy of traffic regression.

5.6 Traffic Prediction

In this section we describe the details of the shift-based model and provide comparisons between the proposed model, the naive linear approach and some additional baselines. Note that the term prediction is used in the context of statistical regression

and social sensors and not predicting future traffic.

5.6.1 Models

To measure the regression improvement of the proposed shift-based model we introduce some baseline models. The first baseline model is the naive linear model that was described in Subsection 5.4.2 (denoted as NAIVE). Since the shift-based model is practically splitting the datapoints in three categories, we have a second baseline model that just picks 3 random partitions and fits a linear model on each one (3-random-partitions model denoted as RAND3). Random partitioning makes sense as a baseline because if the proposed shift-based model had no statistical significance, then it should yield similar results with the random partitioning.

Similar to the shift-based model we also tried to fit the data on a daily basis – one linear fit for each day, from Monday to Friday (baseline denoted as DAY-BASED). Finally, two more models are introduced that use the naive linear model (NAIVE) to fit the datapoints of each day of the week (NAIVE-DAY) and the datapoints of each shift (NAIVE-SHIFT). Practically, for NAIVE-DAY we apply the simple linear model on the data by day of the week and for NAIVE-SHIFT we apply the simple linear model on the data by shift of the day. The last two models are not generated by training data (like the proposed SHIFT-BASED), and do not require cross validation since they do not change; we measure their fitness purely to demonstrate that there is no statistical bias in this SHIFT-BASED model.

The shift-based model (denoted as SHIFT-BASED) is a composite model with a different submodel for each shift of the day (AM, NOON, PM). As described in Section 5.4 the shift-based model is created by applying a least-squares linear model on each shift of the day (Table 5.4 and Figure 5.8). Since the number of datapoints for each shift is

Model	Mean Error			R^2
	Squared	Absolute	Relative	
Naive Linear	5.5856	**6062.0**	**0.6619**	0.6596
Random 3 Partitions	5.8604	6209.8	0.6783	0.6611
Day-based	5.9272	6374.8	0.6658	0.6064
Naive by Day	5.406	5983.5	0.6607	0.6064
Naive by Shift	5.3690	5958.2	0.6584	0.4230
Shift-based	2.4245	**3739.8**	**0.3598**	0.4230

Table 5.5: Error comparison for each regression model. Squared error values are $\times 10^7$.

equal (there is one datapoint for each shift for each day of data), the overall precision of the shift-based model can be defined as the average precision of each submodel. For example, when measuring the squared error of the model we need to compute the squared error for each submodel and then get their average.

5.6.2 Model Comparison

To compare the predictive power of each model the following cross validation setup is used: Repeated random sub-sampling validation. For each model, the data points are randomly ordered and then the first 80% of the datapoints is picked as training dataset and the rest 20% as validation dataset. Using least square regression we fit a linear model to the training data and then calculated the estimation error on the validation data. This process is repeated 1000 times and the average errors across all 1000 splittings are calculated. For the relative errors, all cases where the expected traffic volume is close to 0 were ignored, since it was introducing very large values.

The average squared, absolute, and relative errors for each model are listed in Table 5.5. For completeness of the analysis we also provide the coefficient of determination R^2 in each case. The Shift-based model significantly outperforms all the baseline models which proves that focusing on the different shifts of the day has a statistically significant effect while other approaches like day-based perform poorly. In terms of absolute error,

Model	Mean Error			R^2
	Squared	Absolute	Relative	
Naive Linear	8.5558	7378.3	0.7259	0.4948
Shift-based	3.5027	4527.4	0.3919	0.2571

Table 5.6: Error comparison for the linear and shift-based model with all tweets about traffic (no driver-based filtering). When tweets are not coming directly from drivers in traffic jam the error is significantly higher.

we observe a 38% improvement between the Naive Linear approach and the shift-based model. In terms of relative error we observe more than 45% improvement. Figure 5.12 visually shows how each model compares based on absolute error.

Finally, we calculated the regression error without applying the driver constraint; now tweets can originate by anyone and not only drivers that are stuck in traffic. Our hypothesis is that the raw data will contain excessive noise that will reduce the quality of the regression. In Table 5.6 we list the average errors for the linear and shift-based models. Basic filtering that removes tweets from spam accounts is still applied but all the rest of the tweets from normal Twitter users remain. Using this raw dataset for regression, results in an increased error for both linear and shift-based models. The conclusion from this comparison is that filtering of social posting based on users directly affected by traffic congestion results in a better model and accuracy. We show the absolute error of the shift-based model when all tweets are used (SHIFT-BASED-ALL) in Figure 5.12 and it is 21% higher.

Note again, that even though the Coefficient of Determination (R^2) is lower for the SHIFT-BASED model compared to most baselines, it achieves a very significant improvement in traffic estimation which shows that R^2 is not always a good measure of fitness when modeling this particular traffic/social dataset.

Therefore, using a regression model that attributes the fluctuation of sentiment throughout the day by modeling differently the social behavior for each shift of the day,

we can significantly reduce the traffic estimation error and gain better accuracy. Moreover, filtering the social signal by focusing only on active drivers when estimating traffic congestion further boosts the results since it eliminates the noise from traffic reporting tools and services.

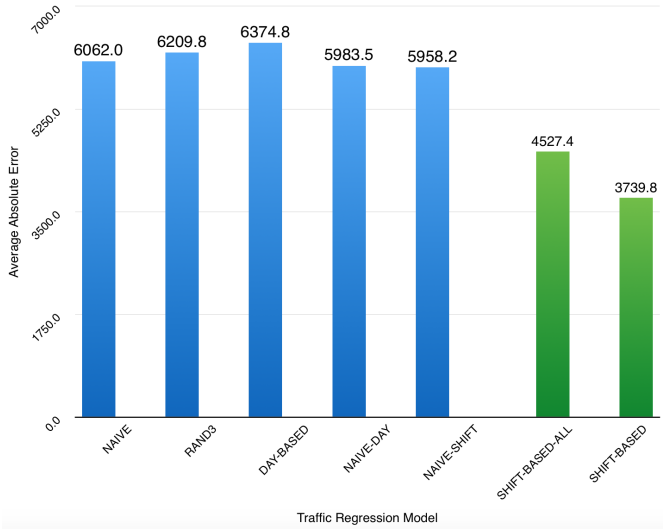


Figure 5.12: Model comparison: Absolute Error. The proposed shift-based model achieves 38% less error than the naive linear model.

5.7 Remarks

Social sensors offer a fast and low cost method to understand the physical world through online content of social media. Mining the correct correlation between the crowd's reaction and an event's magnitude can be very critical and improves our understanding of what is happening and how much it effects our lives. Using the correlation between traffic congestion and social reaction on Twitter as a showcase we show that exploring dimensions that have different psychological links, like the time of the day, can lead to a better grasp of the traffic severity. We propose a novel model to estimate traffic jams using social sensors, that utilizes three linear submodels, one for each shift

of the day (AM, NOON, PM) and social posting from car drivers. We show that the proposed model can be at least 38% better than the naive linear approach and performed several comparisons with different baselines to prove that these findings are statistically significant. We also show that without filtering the tweets by drivers only the regression error of the proposed model would increase by 21% due to the noise. We observe that humans tend to complain more about traffic when they are less happy and we offer the exact linear sub-models that describe these relations between complaints and traffic, for different times of the day. Finally, we offer a potential explanation as to why people complain differently throughout the day, for different levels of traffic congestion, by observing correlation with the general sentiment (mood).

Chapter 6

Future Work in Group Privacy

6.1 Motivation

Due to the public nature of online social media services such as Twitter or Instagram, users typically have a simple goal, to communicate with friend and followers. Many social media users tend to publish a large amount of information that is both structural (likes, favorites, retweets, shares) and non-structural (photos, opinions, conversations) in nature. According to social theory, users "imagine" a specific audience [84] whenever they post something new, and this audience consists of a set of users that they believe will receive and consume their posts. However, the imagined audience and the actual audience of a user's posts can significantly diverge from one another in two ways: First, it can be much smaller than anticipated, since not all followers/friends will necessarily read every post that a given user posts. Second, it can be much larger than expected because the public nature of social media encourages engagement and provides mechanisms to make information widely available and visible. In such cases, a Twitter post by a user with a few followers could grab the attention of a celebrity with millions of followers who might decide to share it, which suddenly exposes it to a vastly larger audience than anticipated.

This difference between the *perceived audience* and the *actual audience* can lead to interesting situations, especially when the audience reaction to what is posted is negative. But even for positive reactions, where the response to a post is widely positive but still massive and unexpected, some people find it difficult to deal with their sudden fame or might even face offline repercussions. A real example is when photos of teenagers become Internet memes and, even if the meme is positively perceived, those teens might face bullying at school purely because they are in the spotlight. In these ways, it makes sense to discuss privacy and private behavior in the context of public information because the networked structure of social media can push public information to audiences beyond those it was intended to reach, thereby potentially creating a privacy breach []. This is different from the more traditional notions of data privacy or privacy that argue any claim to privacy is forfeited as soon as information is posted publicly.

What may be even more interesting from a privacy perspective is when specific characteristics of the person posting on social media is targeted in combination with what they write. For example, the same opinion posted by a LGBT teenager and a non-LGBT adult can create widely different reactions. Moreover, users might positively accept a reaction that challenges solely their opinion versus a reaction that also involves verbal attacks to their personality, age, gender, race, and other social characteristics. This becomes even more important when the opinion is prevalent among people with a specific set of characteristics, which we refer to as a community (Chapter 2). For the purposes of our research, we defined a *community* as a group of social media users who all share some combination of social characteristics, e.g., all Muslim women who live in New York between the ages 22 and 25, or all teenagers of Hispanic origin.

Such communities sometimes behave in a collective way, such as when they discuss in a focused way specific topics or opinions, but due to the inconsistency between imagined and actual audiences, their words may reach other users outside their friendly community

circle. This can result in harassment from malicious users (online or offline) towards members of the community, either collectively or individually. This kind of situation introduces a novel notion of *group privacy* or community privacy as opposed to individual privacy. The general model of *group privacy* is based on the fact that social media users feel that their online (and potentially sensitive) opinions are "a needle in the haystack" where only their imagined audience will read their posts while the vast majority of the social media user-base will remain oblivious to. In reality, and especially for community-focused topics, the actual audience might involve unexpected people, and that could lead to several negative circumstances. For simplicity, we model the online opinions and discussions with the notion of a *topic*. A topic can be a simple phrase, a hashtag, an n-gram, etc. Using topics, the users that have mentioned a specific topic can be clustered in groups. If the users mentioning a topic also happen to form a community, then there is a collective *focus* of the community to this particular topic (as introduced in Chapter 1).

6.2 Group Privacy Definition

A group privacy violation may occur when the combined online actions of individuals, which are public and potentially visible to anyone, lead to the association of *groups* with specific topics and behaviors. Depending on the public reaction to this association (negative vs. positive), a group privacy violation can be harmful to the group and/or its individual members. Even with positive reaction, if it is widespread, it can still result in negative effects for the group.

Definition 1: Collective Behavior The collective behavior of a group includes all the actions (active component) or common characteristics (passive component) of the majority of its members.

Definition 2: Group Privacy Group Privacy describes the state where a group's *private* collective behavior can not be observed.

A group privacy violation involves the exposure of non-obvious collective behavior through the aggregation and mining of publicly visible actions of individuals. Group privacy can be violated through the observation of the public actions of its group members (individual behavior). In Figure 6.1 it is demonstrated how the actions of individuals in social media (discussing about certain topics) can be combined with proprietary knowledge of their characteristics to produce information that an attacker can reverse engineer to discover latent collective behavior of groups. We discuss the active and passive components of this collective behavior and how they can be targeted in the following sections.

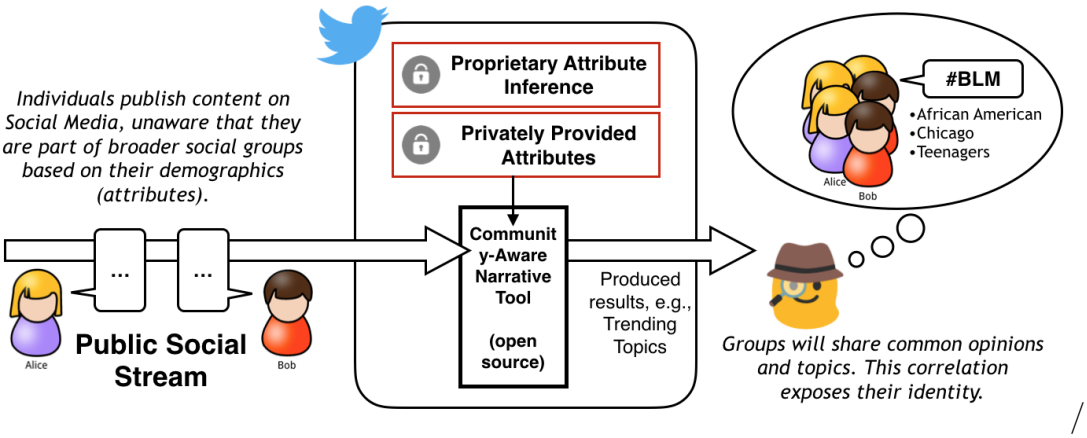
6.3 General Attack Model

Attackers in the context of group privacy have access to public information available on online social media and the necessary computational resources to analyze and aggregate the behavior of individual users which are implicit members of groups. The knowledge of whether a user is a member of a group or not is private for an arbitrary number of users and might not be available to the attacker initially.

The attacker has access to the public portion of a user's profile and the historical stream of their posts. Moreover, we can safely assume that they have access to a Search tool that can perform topic-based search and return the set of users discussing each topic. Such a Search functionality enables the easy and public association between users and mentioned topics. The attacker can exploit this functionality to derive the set of all users that have mentioned a specific topic, for any topic the attacker desires.

Additionally, the existence of trending topics, a feature that summarizes which topics

Figure 6.1: Group privacy visualization.



are popular at a given time, can further be exploited to discover private attributes of the users. This knowledge can then be utilized to derive group memberships. Usually, trending topics are clustered by some attribute like Location or Interest (e.g., politics, technology, pop-culture, etc.). As a very realistic example, Twitter provides trending topics by location at the city level, therefore the attacker can easily derive the location of any user who mentions location-focused topics, where a location-focused topic is a topic that is being discussed in a specific location and nowhere else. Such topics are an easy give-away for the location of involved users since none outside this location mentions the topics.

With all this information available, an attacker can aggregate the behavior of individuals, derive their group membership, and finally, discover latent collective traits of some groups' behavior.

6.4 The Dimensions of Group Privacy

In the context of group privacy and its *collective behavior* there are two components, the *static* that includes the characteristics and attributes of a group (*passive component*)

and the *dynamic* which includes the actions of the group (*active component*). An attacker can target either component in different ways.

6.4.1 Static Group Privacy

Focused topics are very prone to group privacy attacks since they mostly involve a specific group of people with common characteristics. In the presence of a tool, owned by the social media service, that publicly narrates online discussions an attacker can potentially extract the correlation between topics and focused communities, reverse engineer this information to identify all the individuals that talk about specific topics (i.e., using the search functionality), and statistically infer their social characteristics (similar to the example in Figure 6.1. This inference attack introduces two risks: (a) Some community members might not want to be publicly associated with the community but will get exposed nevertheless (i.e., true categorization threat). (b) Someone that is not part of the community but happened to discuss a focused topic, could be incorrectly associated with the community (i.e., false categorization threat).

Regarding the former risk, an inferred group membership can be extremely harmful to an individual, if for example the group is generally a target of harassment (e.g., LGBT teenagers). Individuals willing to protect themselves against such attacks are disadvantaged since they might not be aware of the groups they are a member of (especially if the group is ad-hoc, as described later). The latter risk, wrongful association with a group, can have similar negative effects for the individual.

6.4.2 Dynamic Group Privacy

While static group privacy involves the discovery of latent group characteristics and group membership, dynamic group privacy is more topic-aware and involves the correla-

tion of potentially sensitive topics with potentially sensitive communities.

Definition: Sensitive Information Sensitive information is the information that, when exposed to an audience outside the group's members, can result to negative effects for the group (backlash, harassment, etc.).

Public information shared on social media can still be sensitive since people might not realize that what they post might reach a different audience from the one they have in mind (difference between perceived and actual audience). Based on this notion of *sensitivity* we consider two categories of sensitive information: (a) The topic might be sensitive, e.g., a feminist hashtag in Pakistan. (b) The community itself is sensitive, e.g., it involves generally sensitive attributes like sexual orientation or race. And of course, there is the case that both the topic and the community are sensitive.

Topic sensitivity leads to an interesting observation: a single user might be comfortable discussing a sensitive topic within their online social circle, but when the topic is associated with the user's community characteristics and becomes a target in the context of the general group, then this can become problematic for the whole community and its members.

On the other hand, community sensitivity is more related to the stability of the group in the physical world. To better understand this, we will refer to communities that are not widely recognized in the real world using the terms "implicit" or "ad-hoc" communities. An implicit or ad-hoc community can be formed by a set of characteristics that has not been observed before and even its members might not be aware of its existence as a group. Such groups will be by definition sensitive since they lack the bonding and experience to collectively handle negative comments. However, even groups with high awareness and very specific reason for existence (e.g., member of the social movement Black Lives Matter) can become victims of identity shift when exposed to external friction.

Topic and group sensitivity are often closely correlated since a sensitive community

that is usually the target of negative comments, discussing a specific topic can render the topic sensitive as well. For to this reason, the identification of sensitive topics must be a process aware of correlated communities and their type.

6.4.3 Next Steps

The next steps in this line of research would require validation of the above theories through experimental surveys. Once this validation is performed and we form a better understanding of group privacy we will then explore algorithmic solutions to encounter each dimension of this novel privacy concept.

Chapter 7

Conclusions

In the current Dissertation we explore and demonstrate how studying the characteristics and behavior of the population behind social media discussions can greatly enhance the quality of various Data Mining tasks. This is achieved through a series of related studies that have trending topics and attribute-based communities at the very center. More specifically, we show how extracting communities that are focused on topics leads to better engagement between users and trending topic reports. We offer an algorithmic framework for the efficient identification of topics and their focused communities and explore ranking equations to produce trending topics that have the potential for high engagement by a general audience. We then study the privacy implications of such reporting algorithms and offer solutions from both the system's and the user's perspectives (privacy cyborg). In both cases we protect the users from having their sensitive attributes leaked to an attacker that tries to statistically infer them. Such an attacker would utilize reports of trending topics with focused communities to improve their knowledge about individual users' attribute since the focus property guarantees some degree of correlation between topics and attributes. Shifting further from topics, with the specific application of estimating highway traffic in mind, we identify that specific features like being a driver

and the time of the day can greatly increase the accuracy of the regression task. This demonstrates again the important of understanding the background context behind a text mining challenge rather than just studying text features. Finally, we make the first steps towards an understanding of how privacy in social media might extend to whole communities or groups instead of just individuals.

Bibliography

[1] D. Boyd and K. Crawford, *Critical questions for big data*, *Information, Communication and Society* **15** (2012), no. 5 662–679, [http://dx.doi.org/10.1080/1369118X.2012.678878].

[2] R. Singleton, B. Straits, and M. Straits, *Approaches to social research*. Oxford University Press, 1993.

[3] G. Cormode and M. Hadjieleftheriou, *Finding the frequent items in streams of data*, *Commun. ACM* **52** (Oct., 2009) 97–105.

[4] A. Metwally, D. Agrawal, and A. El Abbadi, *Efficient computation of frequent and top-k elements in data streams*, in *Database Theory - ICDT 2005, 10th International Conference, Edinburgh, UK, January 5-7, 2005, Proceedings*, pp. 398–412, 2005.

[5] C. Budak, D. Agrawal, and A. El Abbadi, *Structural trend analysis for online social networks*, in *PVLDB 4(10)*, pp. 646–656, 2011.

[6] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, *Detecting spammers on twitter*, in *In Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*, 2010.

[7] C. Budak, T. Georgiou, D. Agrawal, and A. El Abbadi, *Geoscope: Online detection of geo-correlated information trends in social networks*, in *PVLDB 7(4)*, pp. 229–240, 2013.

[8] T. Georgiou, A. El Abbadi, and X. Yan, *Extracting topics with focused communities for social content recommendation*, in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pp. 1432–1443, ACM, 2017.

[9] T. Georgiou, A. El Abbadi, and X. Yan, *Privacy cyborg: Towards protecting the privacy of social media users*, in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, 2017.

[10] T. Georgiou, A. El Abbadi, X. Yan, and J. George, *Mining complaints for traffic-jam estimation: A social sensor application*, in *Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*, pp. 330–335, IEEE, 2015.

[11] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen, *Microblogging during two natural hazards events: what twitter may contribute to situational awareness*, in *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 1079–1088, ACM, 2010.

[12] H. Achrekar, A. Gandhe, R. Lazarus, S.-H. Yu, and B. Liu, *Predicting flu trends using twitter data*, in *Computer Communications Workshops*, pp. 702–707, 2011.

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

[14] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, *Sentiment in short strength detection informal text*, *J. Am. Soc. Inf. Sci. Technol.* **61** (Dec., 2010) 2544–2558.

[15] H. Schwartz, J. Eichstaedt, M. Kern, L. Dziurzynsk, and S. Ramones, *Personality, gender, and age in the language of social media: The open-vocabulary approach*, in *PLoS ONE 8(9)*, 2013.

[16] W. Webber, A. Moffat, and J. Zobel, *A similarity measure for indefinite rankings*, *ACM Trans. Inf. Syst.* **28** (Nov., 2010) 20:1–20:38.

[17] F. E. Walter, S. Battiston, and F. Schweitzer, *A model of a trust-based recommendation system on a social network*, *Autonomous Agents and Multi-Agent Systems* **16** (2008), no. 1 57–74.

[18] J. Golbeck, J. Hendler, *et. al.*, *Filmtrust: Movie recommendations using trust in web-based social networks*, in *Proceedings of the IEEE Consumer communications and networking conference*, vol. 96, pp. 282–286, Citeseer, 2006.

[19] T. DuBois, J. Golbeck, J. Kleint, and A. Srinivasan, *Improving recommendation accuracy by clustering social networks with trust*, *Recommender Systems & the Social Web* **532** (2009) 1–8.

[20] Z. Wang, W. Zhu, P. Cui, L. Sun, and S. Yang, *Social media recommendation*, in *Social Media Retrieval*, pp. 23–42. Springer, 2013.

[21] J. Nichols, J. Mahmud, and C. Drews, *Summarizing sporting events using twitter*, in *Proceedings of the International Conference on Intelligent User Interfaces*, IUI, pp. 189–198, 2012.

[22] T. Sakaki, M. Okazaki, and Y. Matsuo, *Earthquake shakes twitter users: Real-time event detection by social sensors*, in *Proceedings of the International Conference on World Wide Web (WWW)*, pp. 851–860, 2010.

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

[24] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling, *Twitterstand: News in tweets*, in *Proceedings of the International Conference on Advances in Geographic Information Systems (GIS)*, pp. 42–51, 2009.

[25] S. Petrović, M. Osborne, and V. Lavrenko, *Streaming first story detection with application to twitter*, in *Human Language Technologies: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT, pp. 181–189, 2010.

[26] H. Abdelhaq, C. Sengstock, and M. Gertz, *Eventweet: Online localized event detection from twitter*, *Proc. VLDB Endow.* **6** (Aug., 2013) 1326–1329.

[27] S. R. Kairam, M. R. Morris, J. Teevan, D. Liebling, and S. Dumais, *Towards supporting search over trending events with social media*, in *ICWSM (International Conference on Weblogs and Social Media)*, AAAI, 2013.

[28] R. Agrawal, T. Imieliński, and A. Swami, *Mining association rules between sets of items in large databases*, *SIGMOD Rec.* **22** (June, 1993) 207–216.

[29] H. Toivonen, *Sampling large databases for association rules*, in *Proceedings of the 22th International Conference on Very Large Data Bases*, VLDB '96, (San Francisco, CA, USA), pp. 134–145, 1996.

[30] V. T. Chakaravarthy, V. Pandit, and Y. Sabharwal, *Analysis of sampling techniques for association rule mining*, in *Proc. of the International Conference on Database Theory (ICDT)*, pp. 276–283, 2009.

[31] H.-P. Kriegel, P. Kröger, and A. Zimek, *Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering*, *ACM Trans. Knowl. Discov. Data* **3** (2009), no. 1 1:1–1:58.

[32] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, *Automatic subspace clustering of high dimensional data for data mining applications*, in *Proceedings of the International Conference on Management of Data(SIGMOD)*, pp. 94–105, 1998.

[33] B. Perozzi, L. Akoglu, P. Iglesias Sánchez, and E. Müller, *Focused clustering and outlier detection in large attributed graphs*, in *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1346–1355, 2014.

[34] E. Zheleva and L. Getoor, *To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles*, in *Proceedings of the International Conference on World Wide Web*, pp. 531–540, 2009.

[35] P. Samarati and L. Sweeney, *k-anonymity: a model for protecting privacy*, in *Proceedings of the IEEE Symposium on Research in Security and Privacy (S&P)*, 1998.

[36] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, *L-diversity: Privacy beyond k-anonymity*, *ACM Trans. Knowl. Discov. Data* **1** (Mar., 2007).

[37] N. Li, T. Li, and S. Venkatasubramanian, *t-closeness: Privacy beyond k-anonymity and l-diversity*, in *ICDE 2007*, pp. 106–115, 2007.

[38] C. Dwork, *Differential Privacy*, pp. 1–12. Springer, 2006.

[39] A. Culotta, N. K. Ravi, and J. Cutler, *Predicting the demographics of twitter users from website traffic data*, in *Proc. of the Conference on Artificial Intelligence*, pp. 72–78, 2015.

[40] N. Talukder, M. Ouzzani, A. K. Elmagarmid, H. Elmeleegy, and M. Yakout, *Privometer: Privacy protection in social networks*, in *Workshops Proceedings of the International Conference on Data Engineering, ICDE*, pp. 266–269, 2010.

[41] F. Bonchi, A. Gionis, and T. Tassa, *Identity obfuscation in graphs through the information theoretic lens*, in *Proceedings of the International Conference on Data Engineering*, ICDE, (Washington, DC, USA), pp. 924–935, 2011.

[42] H. Raymond, K. Murat, and T. Bhavani, *Preventing private information inference attacks on social networks*, *IEEE Trans. Knowl. Data Eng.* **25** (2013), no. 8 1849–1862.

[43] P. Boldi, F. Bonchi, A. Gionis, and T. Tassa, *Injecting uncertainty in graphs for identity obfuscation*, *Proc. VLDB Endow.* **5** (July, 2012) 1376–1387.

[44] E. Ryu, Y. Rong, J. Li, and A. Machanavajjhala, *Curso: Protect yourself from curse of attribute inference: A social network privacy-analyzer*, in *Proceedings of the ACM SIGMOD Workshop on Databases and Social Networks*, DBSocial '13, (New York, NY, USA), pp. 13–18, 2013.

[45] E. Zheleva and L. Getoor, *Preserving the privacy of sensitive relationships in graph data*, in *International Conference on Privacy, Security, and Trust in KDD*, pp. 153–171, 2008.

[46] A. Campan and T. M. Truta, *Data and structural k-anonymity in social networks*, in *PinKDD 2008*, pp. 33–54, 2009.

[47] T. Tassa and D. J. Cohen, *Anonymization of centralized and distributed social networks by sequential clustering*, *IEEE Transactions on Knowledge and Data Engineering* **25** (Feb, 2013) 311–324.

[48] C. Dwork, M. Naor, T. Pitassi, G. N. Rothblum, and S. Yekhanin, *Pan-private streaming algorithms*, in *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pp. 66–80, 2010.

[49] H. Zhang, *The optimality of naive bayes*, *AA* **1** (2004), no. 2 3.

[50] C. E. Shannon, *A mathematical theory of communication*, *SIGMOBILE Mob. Comput. Commun. Rev.* **5** (Jan., 2001) 3–55.

[51] P. Samarati and L. Sweeney, *Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression*, tech. rep., SRI International, 1998.

[52] N. J. Nilsson, *Problem-Solving Methods in Artificial Intelligence*. McGraw-Hill Pub. Co., 1971.

[53] A. Rajaraman, *Data-driven disruption: The view from silicon valley*, *PVLDB* **9** (2016), no. 13 1620.

[54] "Waze: Community-based traffic app." `https://www.waze.com`.

[55] "Twaffic - will twitter and tweets about traffic change the way we drive?." `http://www.slate.com/articles/life/transport/2011/04/twaffic.html`.

[56] "Text messaging law, california, 2008." `https://www.dmv.ca.gov/cellularphonelaws`.

[57] R. Cervero, *Road expansion, urban growth, and induced travel: A path analysis*, university of california transportation center, working papers, University of California Transportation Center, 2001.

[58] H. Summala, *Accident risk and driver behaviour*, *Safety Science* **22** (1996), no. 1-3 103 – 117. Risk Homeostasis and Risk Assessment.

[59] W. Knospe, L. Santen, A. Schadschneider, and M. Schreckenberg, *Human behavior as origin of traffic phases*, *Phys. Rev. E* **65** (Dec, 2001) 015101.

[60] T. Golob and W. Recker, *Relationships among urban freeway accidents, traffic flow, weather, and lighting conditions*, *Journal of Transportation Engineering* **129** (2003), no. 4 342–353, [http://dx.doi.org/10.1061/(ASCE)0733-947X(2003)129:4(342)].

[61] Kai Nagel and Michael Schreckenberg, *A cellular automaton model for freeway traffic*, *J. Phys. I France* **2** (1992), no. 12 2221–2229.

[62] Y. Kryvasheyeu, H. Chen, E. Moro, P. V. Hentenryck, and M. Cebrián, *Performance of social network sensors during hurricane sandy*, *CoRR* **abs/1402.2482** (2014).

[63] M. García-Herranz, E. M. Egido, M. Cebrián, N. A. Christakis, and J. H. Fowler, *Using friends as sensors to detect global-scale contagious outbreaks*, *CoRR* **abs/1211.6512** (2012).

[64] S. Zhao, L. Zhong, J. Wickramasuriya, and V. Vasudevan, *Human as real-time sensors of social and physical events: A case study of twitter and sports games*, *CoRR* **abs/1106.4300** (2011).

[65] C. Aggarwal and T. Abdelzaher, *Social sensing*, in *Managing and Mining Sensor Data* (C. C. Aggarwal, ed.), pp. 237–297. Springer US, 2013.

[66] J. He, W. Shen, P. Divakaruni, L. Wynter, and R. Lawrence, *Improving traffic prediction with tweet semantics*, in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI'13, pp. 1387–1393, AAAI Press, 2013.

[67] A. I. J. a. T. Ribeiro, T. H. Silva, F. Duarte-Figueiredo, and A. A. Loureiro, *Studying traffic conditions by analyzing foursquare and instagram data*, in *Proceedings of the 11th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, PE-WASUN '14, (New York, NY, USA), pp. 17–24, ACM, 2014.

[68] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, *Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones*, in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, (New York, NY, USA), pp. 85–98, ACM, 2009.

[69] E. M. Daly, F. Lecue, and V. Bicer, *Westland row why so slow?: Fusing social media and linked data sources for understanding real-time traffic conditions*, in *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, (New York, NY, USA), pp. 203–212, ACM, 2013.

[70] S. S. Ribeiro, Jr., C. A. Davis, Jr., D. R. R. Oliveira, W. Meira, Jr., T. S. Gonçalves, and G. L. Pappa, *Traffic observatory: A system to detect and locate traffic events and conditions using twitter*, in *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, LBSN '12, (New York, NY, USA), pp. 5–11, ACM, 2012.

[71] "Microsoft azure helps researchers predict traffic jams." http://blogs.msdn.com/b/msr_er/archive/2015/04/02/microsoft-azure-helps-researchers-predict-traffic-jams.aspx.

[72] B. Pan, Y. Zheng, D. Wilkie, and C. Shahabi, *Crowd sensing of traffic anomalies based on human mobility and social media*, in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'13, (New York, NY, USA), pp. 344–353, ACM, 2013.

[73] T. Georgiou, A. El Abbadi, X. Yan, and J. George, *Mining complaints for traffic-jam estimation: A social sensor application*, in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2015, Paris, France, August 25 - 28, 2015*, pp. 330–335, 2015.

[74] "Caltrans performance measurement system (pems)." http://pems.dot.ca.gov.

[75] J. Kwon, B. McCullough, K. Petty, and P. Varaiya, *Evaluation of PeMS to improve the congestion monitoring program*, tech. rep., Final Report for PATH TO 5319, UC Berkeley, Berkeley, CA, 2006.

[76] C. Chen, A. Skabardonis, and P. Varaiya, *Systematic identification of freeway bottlenecks*, in *Proceedings of 83rd Transportation Research Board Annual Meeting*, 2004.

[77] C. Winston, *On the performance of the u.s. transportation system: Caution ahead*, *Journal of Economic Literature* **51** (2013), no. 3 773–824.

[78] "Twitter streaming api." https://dev.twitter.com/docs/streaming-apis/streams/public.

[79] "Twitter search api." https://dev.twitter.com/docs/using-search.

[80] "Los angeles freeways: The great 405." http://www.davestravelcorner.com/guides/losangeles/LA-Freeways.

[81] "Traffic waves." http://trafficwaves.org/trafexp.html.

[82] "Why do we complain? and when should we stop?." http://stress.about.com/od/positiveaffirmations/a/Why-Do-We-Complain-And-When-Should-We-Stop.htm.

[83] S. A. Golder and M. W. Macy, *Diurnal and Seasonal Mood Vary with Work, Sleep, and Daylength Across Diverse Cultures*, *Science* **333** (Sept., 2011) 1878–1881.

[84] A. E. Marwick and D. Boyd, *I tweet honestly, i tweet passionately: Twitter users, context collapse, and the imagined audience*, *New media & society* **13** (2011), no. 1 114–133.