Extracting Topics with Focused Communities for Social Content Recommendation

Theodore Georgiou **Amr El Abbadi** **Xifeng Yan**
University of California, Santa Barbara
Santa Barbara, California, USA
teogeorgiou@cs.ucsb.edu, amr@cs.ucsb.edu, xyan@cs.ucsb.edu

ABSTRACT
A thorough understanding of social media discussions and the demographics of the users involved in these discussions has become critical for many applications like business or political analysis. Such an understanding and its 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.

ACM Classification Keywords
Human-centered computing: Collaborative and social computing—*Social recommendation*; Information systems: Information systems applications: Data mining—*Data stream mining*

Author Keywords
Social Media; Online Communities; Topic Analysis

INTRODUCTION
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. Trending topics have been popularly used in the detection of breaking news, as well as in marketing and advertising mechanisms. In general, a *topic* is a collection of words or phrases that refer to a temporarily popular concept. 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.

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. However, 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 [6] or spammer detection [5]. Alternatively, a space-efficient framework was proposed [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 paper, we propose a novel community detection algorithm utilizing 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.

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

CSCW 2017, February 25–March 1, 2017, Portland, OR, USA.

http://dx.doi.org/10.1145/2998181.2998259

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.

We start by identifying the concept of a *focused community* to enable the efficient extraction of communities interested in topics on social media. We exploit specific properties of this definition to propose a novel framework that receives a social stream as its input and reports topics and the corresponding focused communities as its output. The framework scales linearly with the number of attributes, and reports communities that share a set of attribute combinations or sub-dimensions.

The current research makes the following contributions:

- The definition of a (maximally) focused community.
- A scalable algorithm for the discovery of maximally focused communities with amortized linear time complexity.
- The effectiveness of recommending topics with focused communities is measured through human evaluation.
- The efficiency of the proposed approach is tested through extensive experimentation and analysis on a massive dataset from Twitter and synthetic data.

In the following sections the formal definition of a focused community is provided and then the description of the proposed algorithm to extract such communities is offered. Experimental results to qualitatively showcase the algorithm on a real Twitter dataset follow. Finally, we discuss an application of our approach to rank topics and hence identify trending topics based on content rather than simply popularity or burstiness. Through human evaluations, community-based ranking is shown to be preferable over other standard baselines. The paper concludes with a discussion of related work.

FOCUSED COMMUNITIES

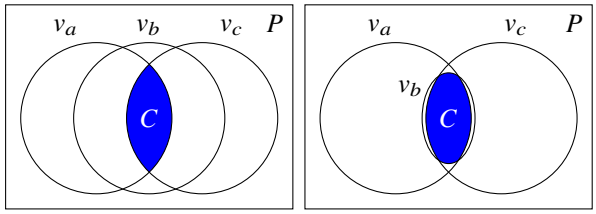
Focused communities are groups of social media users that have a focus on a specific topic and might not be related otherwise. The set of users belonging in a focused community share two properties: they all mentioned the same topic and they all share *some* characteristics. In order to extract and understand 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 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



(a) Non focused community (b) Focused community ($\epsilon > 0$)

Figure 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.

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

$$\left| \frac{|C|}{|P_v|} - 1 \right| \leq \epsilon \qquad (1)$$

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 (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).

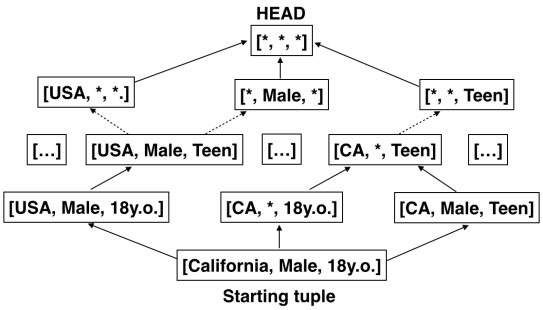


Figure 2: Partial view of the attribute lattice. Two connected nodes (solid arrow) in the lattice indicate that a tuple can be reached from the other through a single attribute generalization. A dashed arrow indicates that two nodes have other nodes between which are omitted due to space restrictions.

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: $|HEAD| = |P|$. Figure 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 re-

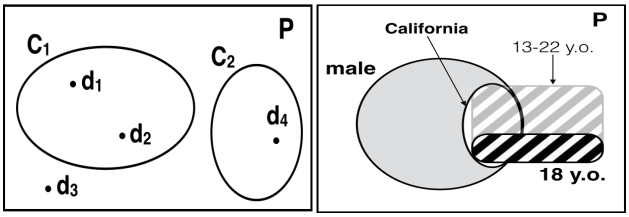


Figure 3: (a) Sampling phase example. (b) An example case where the greedy attribute selection policy can fail to select the best attribute.

quirement, 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 $\xi|P|$.

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.

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 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 (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.

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 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 3a 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 → California → USA), the gender value once (Male → *), and the age value once (18 → 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 $t \geq C_t$. The number of trials is equal to the size of the sample: $|S| = k$. The probability of success in a single trial is equal to $p = |C|/|P|$. 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 without replacement) is equal to 1 minus the probability of getting 0 successes. This probability is defined by the geometric equa-

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};
```

tion that describes the CDF of k Bernoulli trials: $1-(1-p)^k$. Therefore, we have:

$$p_b = 1-(1-|C|/|P|)^k = 1-(1-p)^k \quad (2)$$

We want to find the *minimum* value of k so that the right hand of Equation (2) 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.

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 (1) is utilized to *climb* the *lattice* (see Figure 2) from t to a new tuple $t' \leq t$, 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* $t \geq C_t$. 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:

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

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 $t \geq C_t$.*

Proof. Let C be a *maximally focused* community with size $|C| \geq \xi P$ and with a focus metric less than ϵ. Let t be a starting tuple with n attribute values so that $t \geq C_t$ (C_t can be reached by generalizing attribute values in t). C_t can be cor-

rectly 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 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:

$$\underset{a_g \in t}{\operatorname{argmin}}\, support(a_g.value, P) \tag{3}$$

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 3b 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.

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.

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: $m_f(1-\xi)|P|$. 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 $\xi \to 1$.

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.

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 [23, 2]). 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 [9]. 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 [21]. 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. [20] 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.

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 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, $\epsilon : 0.15$, $\xi : 0.3$. 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).

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 1 (general interest trends) and Table 2 (trends with a sports related focus). A "*" value indicates that the attribute got generalized to its top level of the hierarchy. A "$\perp$" 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 *#Disney-Side* which was a social 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 *#Bring1DtoGreece* , 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.

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.

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

Table 1: 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

Table 2: 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

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:

$$score(T) = |P| \times rp(C) \times icf(C) \quad (4)$$

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.

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 [26]. 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 provide 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 for the first experiment (f-topics baseline) and Table 4 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.

RELATED WORK

Existing social content recommendation systems have mainly relied on the similarity of users in the social network. Walter et al. [24] have proposed a model to use the users' social connections to reach contents and filter the contents by their trust relationship. Golbeck et al. [11] have considered online social networks as recommendation networks by exploiting the easiness of information cascades on such platforms. DuBois

Table 3: 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**	**All Pools 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 4: 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**	**All Pools 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%

et al. [10] 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. [25]. 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 [15], Earthquakes [18], News reporting [14, 19], general event detection [17, 1], or search support on trending events [12]. In this paper we research the novel idea of identifying the underlying user communities 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 [4] 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 [22] and Chakaravarthy et al. [8]. 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 [13]. 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 [4] and [3]). 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 community extraction have also been explored in Perozzi et al. [16]. 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.

CONCLUSIONS

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 (#ReasonsTo-

VisitEgypt). Such data can be used to better understand a topic's population and, essentially, recommend more relevant and interesting social content to the users.

ACKNOWLEDGMENTS

This work is supported by NSF grant CNS 1649469.

REFERENCES

1. H. Abdelhaq, C. Sengstock, and M. Gertz. Eventweet: Online localized event detection from twitter. *Proc. VLDB Endow.*, 6(12):1326–1329, Aug. 2013.
2. H. Achrekar, A. Gandhe, R. Lazarus, S.-H. Yu, and B. Liu. Predicting flu trends using twitter data. In *Computer Communications Workshops (INFOCOM WKSHPS)*, pages 702–707, 2011.
3. 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)*, pages 94–105, 1998.
4. R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993.
5. 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.
6. C. Budak, D. Agrawal, and A. El Abbadi. Structural trend analysis for online social networks. In *PVLDB 4(10)*, pages 646–656, 2011.
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)*, pages 229–240, 2013.
8. 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)*, pages 276–283, 2009.
9. Maxmind world cities with population. `http://www.maxmind.com/app/worldcities`.
10. T. DuBois, J. Golbeck, J. Kleint, and A. Srinivasan. Improving recommendation accuracy by clustering social networks with trust. *Recommender Systems & the Social Web*, 532:1–8, 2009.
11. 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*, volume 96, pages 282–286. Citeseer, 2006.
12. 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.
13. 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(1):1:1–1:58, 2009.
14. 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, pages 591–600, 2010.
15. J. Nichols, J. Mahmud, and C. Drews. Summarizing sporting events using twitter. In *Proceedings of the International Conference on Intelligent User Interfaces*, IUI, pages 189–198, 2012.
16. 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)*, pages 1346–1355, 2014.
17. 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, pages 181–189, 2010.
18. 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)*, pages 851–860, 2010.
19. 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)*, pages 42–51, 2009.
20. 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.
21. 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(12):2544–2558, Dec. 2010.
22. H. Toivonen. Sampling large databases for association rules. In *Proceedings of the 22th International Conference on Very Large Data Bases*, VLDB '96, pages 134–145, San Francisco, CA, USA, 1996.
23. 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*, pages 1079–1088. ACM, 2010.
24. 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(1):57–74, 2008.
25. Z. Wang, W. Zhu, P. Cui, L. Sun, and S. Yang. Social media recommendation. In *Social Media Retrieval*, pages 23–42. Springer, 2013.
26. W. Webber, A. Moffat, and J. Zobel. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4):20:1–20:38, Nov. 2010.