

# Group Chats on Twitter

James Cook<sup>\*</sup>  
UC Berkeley  
jcook@cs.berkeley.edu

Krishnamurthy Kenthapadi  
Microsoft Research  
krisken@microsoft.com

Nina Mishra  
Microsoft Research  
ninam@microsoft.com

## ABSTRACT

We report on a new kind of group conversation on Twitter that we call a group chat. These chats are periodic, synchronized group conversations focused on specific topics and they exist at a massive scale. The groups and the members of these groups are not explicitly known. Rather, members agree on a hashtag and a meeting time (*e.g.*, 3pm Pacific Time every Wednesday) to discuss a subject of interest. Topics of these chats are numerous and varied. Some are support groups, for example, post-partum depression and mood disorder groups. Others are about a passionate interest: topics include skiing, photography, movies, wine and foodie communities. We develop a definition of a group that is inspired by how sociologists define groups and present an algorithm for discovering groups. We prove that our algorithms find all groups under certain assumptions. While these groups are of course known to the people who participate in the discussions, what we do not believe is known is the scale and variety of groups. We provide some insight into the nature of these groups based on over two years of tweets. Finally, we show that group chats are a growing phenomenon on Twitter and hope that reporting their existence propels their growth even further.

## Categories and Subject Descriptors

J.4 [Computer Applications]: Social and Behavioral Sciences; H.2.8 [Information Systems]: Database Applications—*Data Mining*

## Keywords

Groups, Periodic time series, Online communities

## 1. INTRODUCTION

The term community has classically been associated with local physical meetings of groups of people such as the Lions Club and Rotary Club. Participation in these groups is primarily for social capital, *e.g.*, mutual support, cooperation, trust, good will, fellowship and sympathy. Over time, many of these physical communities have dissipated due to factors such as urban sprawl, two working parents and time pressures [22].

Virtual communities on the other hand have surged in comparison. The term *virtual community* was first coined in 1993 by

Howard Rheingold [23] who described them as “social aggregations that emerge from the Net when enough people carry on public discussions long enough, with sufficient human feeling to form webs of personal relationships in cyberspace”. Virtual communities include forums, chat rooms, discussion boards, Usenet groups and Yahoo groups [3].

In this paper, we report on a new kind of community on Twitter. We call these communities *group chats* as they are periodic, synchronized conversations focused on specific topics. Topics of these chats are vast and varied. There are support groups for post-partum depression and mood disorders. On the hobby side, conversations span skiing, photography, wine and foodie communities. The way that people communicate in a group conversation is via an agreed upon *hashtag* (a short string preceded by a ‘#’ sign). For example, in a group of passionate movie-goers, members agree to include “#mtos” in every tweet. In addition to agreeing on a hashtag, members also agree on a day and time, *e.g.*, every Sunday evening at 20:00 GMT, and hence the abbreviation Movie Talk on Sunday. Many of these groups are moderated to ensure that the meeting has a focused subject, *e.g.*, suspense movies. Live conversations among like-minded people can be quite exhilarating for the participants, whether tweeting or just watching.

Group chats are similar to virtual communities in that the primary venue for communicating is online. As with other communities, the purpose is to exchange knowledge, to share experiences, to provide empathy and generally feel part of a community. Also, groups chats can be sprawled out geographically and thus they enable a way to meet that would otherwise be difficult or impossible. However, they differ from virtual communities in that they use the Twitter platform as a real-time mechanism to communicate in groups. The real-time nature of these conversations imparts a live feeling that is hard to duplicate in other online venues such as forums. Another key difference is that these groups are implicit and therefore not easily discoverable. In contrast, the list of Yahoo/IRC (Internet Relay Chat) groups can be searched and browsed.

Twitter group chats are similar to classical communities in that they have fixed meeting times and talk live. They are different from classical communities in that their meetings are not in the same physical location and what they say is more abbreviated and public to the world. Topics of group chats can be very niche, *e.g.*, rare disease communities, whereas the physical location restriction imposed by classical communities may make niche groups harder to form.

That Twitter would be used to organize such discussions is quite surprising. The 140 character limit imposes a succinctness that seems unsuitable for group discussions. But the already large-scale adoption of Twitter, coupled with a very real-time platform, has enabled these group conversations at a massive scale. Also surprising

<sup>\*</sup>Work done at Microsoft Research.

are support group chats. It is hard to imagine how support can be given or received in 140 characters. But people have found a way!

The existence of group chats is certainly known to members who participate in these chats. We have not seen any work in the published literature reporting the number and variety of these groups. Our work is focused on algorithms for automatically finding groups at scale.

**Contributions:** We begin by presenting key properties of groups abstracted from the sociology literature that culminate in a formal definition of a group. The key components of the definition are routine bounded-length meetings together with cohesion among the active members. We found periodic, fixed duration meetings, such as ‘every Wednesday at 2-3pm PST’, to be quite common among online groups. A predictable, agreed-upon meeting time helps people plan their schedule and focus for a bounded time on a particular subject. However, periodic bounded duration meetings are not enough. For example, weekly television shows such as ‘Dancing With the Stars’ are extremely periodic, fixed duration meetings, but are not group conversations. Users just shout their opinions, but do not converse with others. The last component of the definition is cohesion in the group which we measure by communication among active group members.

Next, we propose an algorithm for finding groups. We begin with a large collection of candidate groups and repeatedly remove those that do not satisfy our definition. Every hashtag is an initial candidate. In the first step, the algorithm removes candidates that do not have routine meetings. To identify these, for each candidate, the time series of exchanged messages is computed and a Fourier transform based method is applied to the time series to identify periodic candidates. At this point, the candidates may contain those that “meet” all day, for example, ‘Follow Friday’. In order to eliminate those, we restrict ourselves to groups where the propensity of messages are exchanged in a short duration. At this point, the remaining candidates may still include those that meet but never really engage in a conversation. In the final step we remove candidates that are not cohesive. What remains are the groups.

We then prove that hashtags that are close to group chats are accepted by our algorithm and hashtags that are far from group chats are rejected. Specifically, if a group meets sufficiently many times, group meetings are well-separated, and a reasonable number of tweets are exchanged per meeting, then we prove that our algorithm will accept the hashtag, under certain assumptions. On the flip side, if a group has no synchronized time when it meets, or if the meetings are irregularly-spaced apart, or if the group is not cohesive, we prove that our algorithm will reject the hashtag.

Finally, we run our algorithm over two years of Twitter data. We find 1.4K groups involving 2.3M users. To provide a glimpse into these groups, we show a distribution of the periods of these groups, demonstrating that most groups meet weekly. To validate the quality of the groups discovered, we randomly sample 10% of the groups and report on the categories of groups represented in the sample, finding that most are interest-driven groups such as music enthusiasts, sports lovers, foodie communities. We also find many support and self-help groups. Finally, we compute the birth and death of these groups showing the cumulative number of living groups over time. The data suggests that group chats are a growing phenomena. We hope that this discovery instigates the creation of even more group chats on Twitter.

## 2. RELATED WORK

*Twitter group chats:* In the published literature, we found very little discussion of Twitter group chats. There are articles discussing the

benefits of a single education group chat called #edchat [9, 14], but nothing reporting the number and variety. Budak and Agrawal [6] investigate characteristics of education group chats that lead to continued individual participation. Our work is aimed at methods for automatically discovering group chats. While we did find a crowd-sourced spreadsheet [24] of group chats, in our experience, there are many chats listed in this spreadsheet that are now defunct, many chats that are missing (possibly because the moderator of the group chat was not aware of the spreadsheet), some that do not have predictable meetings, and others that are not truly group conversations, e.g., one-time chats.

*Definition of groups:* Many different definitions of groups have been proposed in sociology literature, for example, “A group is a collection of individuals who have relations to one another” [7], “A group is a bounded set of patterned relations among members” [1], and “A group is two or more individuals who are connected by and within social relationships” [13]. An overview of definitions of online communities provided by various disciplines is presented in the survey by Iriberry and Leroy [16]. These definitions are qualitative, and involve explicitly defined groups. We, on the other hand, provide a quantitative definition of a group and study groups that are implicit and hence not easily discoverable.

*Nature and formation of groups:* There is extensive literature on the formation of groups, the nature and purpose of groups that exist, and the causes of their success or failure. Besides focusing on the life-cycles of online communities and factors that contribute to their success, the survey by Iriberry and Leroy [16] also includes discussion of the “importance and benefits” of online communities, and of the types of communities that exist. Group formation on LiveJournal and DBLP data has been studied in [2]. The nature of the communities that exist in Yahoo! groups has been examined in [3], wherein factors influencing stickiness of a user with a group have been explored. The factors that motivate participants to stay with a group have been explored in [4]. Surveys of the public studying how many people participate in online groups and for what purpose are instructive: for example, a 2001 study found that 84% of Internet users participated in online groups, and a survey conducted in 2010 found that 23% of Internet users living with a chronic ailment have looked for support online, and people with rare conditions are even more likely to do so [12, 15, 17]. In contrast to this line of work, our focus is on algorithmically identifying groups in Twitter. Investigating the nature and formation of Twitter groups is a promising direction for future work.

*Determining periodicity:* Our algorithm for detecting group chats uses Fourier analysis to help determine whether a hashtag has regular meetings. Many different approaches have been proposed for periodicity detection in time-series data, for example, using Fourier analysis [25] and Wavelet transforms [20]. We adapt the *autoperiod* method proposed in [25].

## 3. PRELIMINARIES

In this section, we work towards a definition of a group. We are not aware of any quantitative definitions of a group in the sociology literature. Instead, there are many qualitative definitions with no convergence towards a single definition [11]. We begin by describing three key properties of a group:

1. **REGULAR:** In a group, people who share an interest meet on a regular basis over a prolonged period of time.
2. **SYNCHRONIZED:** In a group, meetings occur for a fixed duration at a specified time.

3. COHESIVE: Members in a group communicate with each other over the course of many meetings.

These properties can be instantiated in many ways. In this paper, we interpret them in one way that leads to our definition of a group.

**DEFINITION 3.1.** *A set of people form a group  $G$  if every  $\tau$  days some subset of  $G$  meet for a duration of  $l$  hours where at least  $\nu$  fraction of pairs of users in  $G$  exchange messages during each meeting.*

Observe that this definition captures all of the key properties that we outlined above. Meetings are regular because they occur every  $\tau$  days. They are synchronized because they last  $l$  hours and they are cohesive because at least  $\nu$  fraction of pairs communicate.

To motivate each component of the definition in the specific context of Twitter, we describe a few crucial examples that steered our thinking. Note first that the candidate set of groups is defined by the set of hashtags and the candidate members of a group are defined by users who tweet with the hashtag. The first component of our definition corresponds to periodic meetings. The justification for this property is that in order for group connections to form, members must meet predictably over the course of many meetings. We observed this property in the group chat hashtags that we discovered such as Movie Talk on Sunday, which meets every Sunday (see Figure 1(a) for a time series of tweets for #mtos). Even offline meetings of local support groups often occur on a weekly or monthly basis. Note that in order to satisfy Property 1, meetings need not necessarily occur every  $\tau$  days, but periodic meetings are the focus of the present work.

However, periodic meetings are not enough since there are hashtags such as #monday that surge once a week but are not group conversations (see Figure 1(b) for a time-series of #monday tweets). Users simply append #monday to their tweets on Monday. A key difference between such hashtags and group conversations is that #monday surges all day Monday, while group meetings surge for only a fixed duration (Figure 2). This is how we arrived at the second component of our definition, that the group meet for a fixed duration of time. Offline meetings also follow such a time-bounded pattern.

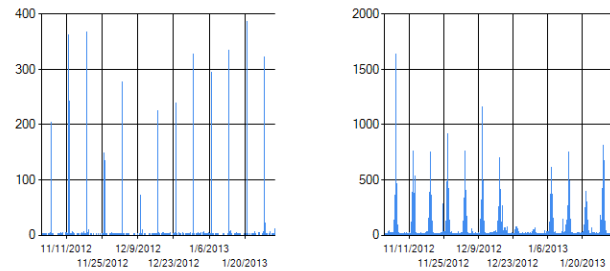
However, even these two requirements are not sufficient since there are hashtags tied to TV shows such as Dancing with the Stars which surge for one hour every week when the TV show airs. Users are not communicating with each other, just shouting with the goal of having their tweet broadcast on live television. This motivates the third aspect of our definition, cohesion. The difference between such a TV show and a real group conversation is that users actually communicate with each other during a group conversation. In the context of Twitter, we observe this behavior via direct @-messaging between group members.

We note that our definition of a group may be improved in a number of ways. For example, there may be groups that meet regularly but are not periodic, *e.g.*, meet at least once a week but schedule the next meeting at the end of each meeting. We miss such groups since they do not have a periodic structure. Other groups may meet, but not use a hashtag. Again, our work misses such groups. We leave the question of alternate group definitions as a subject for future work.

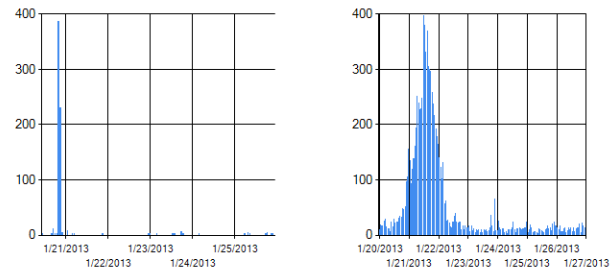
### 3.1 Notation

Each tweet  $d$  is associated with a timestamp  $t(d)$  that takes integer values between 0 and  $\mathcal{T}$ . The unit of time can be chosen to be sufficiently granular (say, seconds or milliseconds) so that tweets can be assumed to occur only at integral time units. The timestamp

0 can correspond to a point of time before Twitter existed, *e.g.*, beginning of the year 2000. Let  $H$  denote the set of distinct hashtags contained in the text of all tweets and  $H_{per} \subseteq H$  denote the set of periodic hashtags. For a hashtag  $h \in H$ , let  $f_h(\cdot)$  denote the *timeline function* for  $h$ , that is,  $f_h(t)$  equals the number of tweets containing hashtag  $h$  that occurred at time  $t$ . Denote the total number of tweets containing  $h$  as  $\alpha_h$ . Denote the period of a periodic hashtag  $h$  by  $\tau_h$ . Denote the number of meetings associated with a periodic and synchronized hashtag  $h$  by  $m_h$ . We omit the subscript when the hashtag is clear from the context. Denote the maximum allowed duration of a meeting by  $l$ . While our definition of a group assumes that the period is in days and the meeting duration is in hours, from here onwards, we assume that both  $\tau$  and  $l$  are in the same units.



**Figure 1: Number of tweets per hour for the hashtag mtos (left) and the hashtag monday (right) over a three month timeframe (11/2011 - 1/2013). Both hashtags are periodic, but monday is not a group conversation.**



**Figure 2: Number of tweets per hour for the hashtag mtos (left) and the hashtag monday (right) over a one week timeframe (last week of 1/2013). The hashtag monday is active all day, while mtos is active for a one hour duration.**

## 4. FINDING GROUP CHATS ON TWITTER

Our method for determining the set of group chats on Twitter is outlined in Algorithm 1. We first identify the set of periodic hashtags on Twitter. Next, for each periodic hashtag, we check whether meetings using the hashtag occur at a predefined time during each period. If such meetings exist, we compute the meeting time, and otherwise, exclude the hashtag from further consideration. We then check if there is sufficient communication among the top users of the hashtag, and if not, exclude the hashtag. Group chats correspond to the remaining periodic hashtags.

---

**Algorithm 1** FINDTWITTERGROUPCHATS

---

**Input:** Twitter data over a period of time, consisting of Twitter handle (user alias), timestamp, users that are @-messed in the tweet.  
**Output:** Set of hashtags that are group chats, along with the associated meeting time and other metadata for each group chat.

- 1: (REGULAR) Identify the set  $H_{per}$  of periodic hashtags on Twitter, along with the period for each hashtag. (§4.1)
  - 2: **for** each periodic hashtag  $h$  in  $H_{per}$  **do**
  - 3: (SYNCHRONIZED) Check whether meetings using hashtag  $h$  occur at a predefined time during each period. If yes, compute the meeting time, and otherwise, exclude  $h$  from the set of candidate group chats in  $H_{per}$ . (§4.2)
  - 4: (COHESIVE) Determine if there is sufficient communication among the top users of hashtag  $h$ . If not, exclude  $h$  from the set of candidate group chats in  $H_{per}$ . (§4.3)
  - 5: Return the set of remaining hashtags in  $H_{per}$ , along with the meeting time and other metadata.
- 

---

**Algorithm 2** FINDPERIODICHASHTAGS

---

**Input:** Twitter data over a period of time, consisting of Twitter handle (user alias), timestamp, tweet text and other metadata for each tweet; Periodicity threshold,  $\delta < \frac{1}{2}$ .

**Output:** List of periodic hashtags, along with the period for each hashtag.

- 1: Identify the set  $H$  of distinct hashtags contained in text of all tweets.
  - 2: **for** each hashtag  $h$  in  $H$  **do**
  - 3: Determine the timeline function  $f_h$  for hashtag  $h$ .
  - 4: Compute the Fourier transform of  $f_h$  and obtain Fourier coefficients  $\hat{f}(\cdot)$  for a large set of candidate frequencies,  $\{1/T_1, \dots, 1/T_r\}$ . (§4.1.1)
  - 5: Compute the autocorrelation  $\tilde{A}(\cdot)$  of  $f_h$  for the corresponding set of candidate periods,  $\{T_1, \dots, T_r\}$ . (§4.1.2)
  - 6: Define the periodicity coefficient for period  $T_k$  as  $S(T_k) := \frac{|\hat{f}(1/T_k)|}{|\hat{f}(0)|} \cdot \frac{|\tilde{A}(T_k)|}{|\tilde{A}(0)|}$ , for  $1 \leq k \leq r$ . (§4.1.3)
  - 7: Determine the candidate period  $\tau_h$  with the largest periodicity coefficient, that is,  $\tau_h := \arg \max_{1 \leq k \leq r} S(T_k)$ , and output  $\langle h, \tau_h \rangle$  if the corresponding coefficient exceeds the threshold, that is, if  $S(\tau_h) \geq \delta$ . (§4.1.3)
- 

## 4.1 Regular

The main technical component of our approach is to determine whether a hashtag is periodic (Property 1). We first present the desiderata through an example, and then design an algorithm to satisfy them.

Figure 1(a) shows part of the timeline for the hashtag #mtos. The group corresponding to this hashtag meets at a predefined time every week. We observe that the hashtag is mentioned frequently during the weekly meetings and infrequently between sessions, resulting in a timeline that is visually periodic. However, the number of occurrences of the hashtag varies widely across meetings. Further, some meetings may not occur, *e.g.*, around Christmas. Additionally, there may be a few occurrences of the hashtag between meetings, *e.g.*, for generating awareness about upcoming meetings.

Our algorithm for detecting periodicity employs the Fourier transform as a key ingredient for achieving robustness to the above factors. Given a function  $f$  and a frequency  $\xi$ , the Fourier transform produces a *Fourier coefficient*  $\hat{f}(\xi) \in \mathbb{C}$ . The Fourier transform satisfies the following property: *If  $f$  is periodic over a large inter-*

*val with period  $\tau = 1/\xi$ , then the magnitude of the corresponding Fourier coefficient,  $|\hat{f}(\xi)|$  is large.* This property is robust to variations in the intensity of  $f$  from cycle to cycle and addition of a small amount of noise. Hence, a first attempt would be to check whether the largest Fourier coefficient is comparable in magnitude to the total number of tweets containing a hashtag and if so, declare the hashtag to be periodic with the corresponding period.

However, the converse of the above property is not true: in particular,  $|\hat{f}(\xi')|$  is also large whenever  $\xi' = k\xi$  is an integer multiple of  $\xi$ . For example, a group chat that occurs at noon every alternate Monday would have a strong Fourier coefficient at the once-per-two-week frequency, but also at a frequency of once per week and even once per day. To distinguish the base frequency  $\xi$  from multiples  $k\xi$ , we measure the *autocorrelation* of a hashtag’s timeline. Given a function  $f$ , the autocorrelation  $A_f(t)$  is a measure of the similarity between  $f$  and the same function  $f$  shifted by  $t$ .

The autocorrelation satisfies a property similar to that of the Fourier transform: *if  $f$  is periodic over a large interval with period  $\tau$ , then  $A_f(\tau)$  is large.* Intuitively, a periodic function shifted by its period (or integer multiples of period) aligns well with the original function, resulting in a large autocorrelation comparable to the autocorrelation at  $t = 0$  (the function with itself). In other words, the autocorrelation is large for integer multiples  $k\tau$  of the base period, rather than periods  $1/(k\xi) = \tau/k$  associated with integer multiples of the frequency  $\xi$  as is the case with the Fourier transform. The only periods that have both large Fourier coefficients and autocorrelations should be close to the true period  $\tau$ .

Algorithm 2 formalizes the above intuition. We first determine the timeline function for a hashtag. Then, we compute its Fourier transform for a large set of candidate frequencies as well as autocorrelation for the corresponding candidate periods. The periodicity coefficient for each candidate period is computed as the product of two ratios: the ratio of the corresponding Fourier coefficient to the total number of tweets and the ratio of the autocorrelation for this period to the autocorrelation of the function with itself. Since these two ratios will be large at  $\tau$  for a periodic function with period  $\tau$ , the algorithm checks if the largest periodicity coefficient exceeds a given threshold, and if yes, outputs the hashtag along with the corresponding period. We provide the details below.

The idea of combining the Fourier transform with autocorrelation has been explored by Vlachos et al. [25] in their work on detecting periodicity. Our method differs in two respects. First, while they use the discrete Fourier transform (DFT) of  $f$ , we obtain samples from the continuous Fourier transform. At the cost of requiring more computation, this choice allows us to measure Fourier coefficients for all frequencies of interest to us, for the whole data set at once. The DFT only produces frequencies which are multiples of the inverse total window length: for example, when examining six weeks of data, the DFT could measure frequencies of once every 0.75 weeks, once every 1.5 weeks or once every 3 weeks, but not once per week or once per two weeks. Second, they distinguish between ‘hills’ and ‘valleys’ of the autocorrelation, likely to compensate for the lack of precision in their choice of Fourier coefficients. In our implementation, we combine Fourier and autocorrelation coefficients by simply multiplying the closest known Fourier and autocorrelation coefficients to a candidate period.

For detecting periodic hashtags, we first attempted to use Kleinberg’s burst detection algorithm [19], which detects periods of high activity using a generative model that switches between a low-activity state and a high-activity “bursty” state. However, we chose the Fourier analysis based method instead, for two reasons. First, Kleinberg’s model has a parameter that determines how easily the underlying model switches to a bursty state, and we had trouble

finding a value which worked for all group chats. Second, Kleinberg’s method does not detect whether or not the bursts are of a periodic nature, nor does it produce the period; both of these are natural outputs of the Fourier analysis based method. Naïve attempts to measure the period as an average time between bursts produced by Kleinberg’s method are thwarted by missing meetings or short bursts that occur between meetings, although it is possible that some adaptation of Kleinberg’s algorithm could overcome these limitations.

#### 4.1.1 The Fourier Transform

Given a function  $f : \mathbf{R} \rightarrow \mathbf{C}$ , the Fourier transform  $\hat{f} : \mathbf{R} \rightarrow \mathbf{C}$  applied to a frequency  $\xi \in \mathbf{R}$  is defined as:

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i t \xi} dt. \quad (1)$$

Since  $f_h$  is a discrete function whereas the above definition is for continuous functions, we can define  $f$  as a function that has a spike of area  $f_h(t)$  around discrete times  $t$ :  $f(t') := f_h(t)/\epsilon$  for  $t - \epsilon/2 \leq t' \leq t + \epsilon/2$ . A spike of area  $f_h(t)$  around time  $t$  contributes approximately  $f_h(t) \cdot e^{-2\pi i t \xi}$  to the Fourier transform (1), which becomes exact as the width of the spike tends to 0. Hence, we compute the Fourier transform as:

$$\hat{f}(\xi) = \sum_{0 \leq t \leq T} f_h(t) \cdot e^{-2\pi i t \xi}. \quad (2)$$

The Fourier coefficient with the largest magnitude is achieved at  $\xi = 0$  and equals the total number of tweets with the hashtag, that is, for any  $\xi$ ,  $|\hat{f}(\xi)| \leq \hat{f}(0) = \alpha_h$ .

#### 4.1.2 Autocorrelation

The autocorrelation of a function  $f : \mathbf{R} \rightarrow \mathbf{C}$  with respect to a period  $\tau$  is defined as:

$$A_f(\tau) = \int_{-\infty}^{\infty} f(t + \tau)\bar{f}(t)dt.$$

The magnitude of the autocorrelation  $|A_f(\tau)|$  is always highest at  $\tau = 0$ . Autocorrelations can be computed from the Fourier transform as:

$$A_f(\tau) = \int_{-\infty}^{\infty} |\hat{f}(\xi)|^2 e^{-2\pi i \xi \tau} d\xi. \quad (3)$$

#### 4.1.3 The Periodicity Coefficient

Given the timeline function  $f_h$  for a hashtag, we first compute the Fourier coefficients  $\hat{f}(\xi_j)$  for  $N_F$  equally spaced frequencies  $\xi_j$  in a fixed range  $[-1/\tau_F, 1/\tau_F]$  using (2). ( $F$  stands for Fourier. In our implementation,  $\tau_F$  was twelve hours.) Then, we use the computed Fourier coefficients to approximate the autocorrelation (3) as:

$$\tilde{A}(\tau_k) = \sum_{j=1}^{\tau} |\hat{f}(\xi_j)|^2 e^{-2\pi i \xi_j \tau_k},$$

for a large but fixed set of periods  $\tau_k$ .

Now, define the *periodicity coefficient* for a candidate period  $T$  as:

$$S(\tau) := \frac{|\hat{f}(\xi^*)|}{|\hat{f}(0)|} \cdot \frac{|\tilde{A}(\tau^*)|}{|\tilde{A}(0)|},$$

where  $\xi^*$  and  $\tau^*$  are the closest available frequency and period in the sets  $\{\xi_i\}$  and  $\{\tau_k\}$ , respectively. Note that  $S(\tau)$  is always between 0 and 1. Then, we determine the period  $\tau$  with the largest

---

#### Algorithm 3 SYNCHRONIZEDMEETINGS

---

**Input:** Timeline function  $f$  and period  $\tau$  for a given periodic hashtag  $h \in H_{per}$ ; Maximum meeting duration,  $l$ ; Synchronization threshold,  $\gamma$ .

**Output:** Boolean flag (‘valid’/‘invalid’) denoting whether  $h$  has synchronized meetings; (if  $h$  is valid) Meeting start time  $\tilde{t}$ .

- 1: Define compressed timeline function,
 
$$g(t) := \sum_{0 \leq i \leq \lfloor T/\tau \rfloor - 1} f(t + i \cdot \tau), \text{ for } 0 \leq t < \tau.$$
  - 2: Define score associated with a potential meeting that starts at offset  $t$  during every period as
 
$$\beta(t) := \frac{1}{\alpha} \cdot \sum_{0 \leq z < \tau} g((t + z) \bmod \tau), \text{ for } 0 \leq t < \tau.$$
  - 3: Determine the candidate meeting start time  $\tilde{t}$  with the largest score, that is,  $\tilde{t} := \arg \max_{0 \leq t < \tau} \beta(t)$ , and output  $\langle h, \text{‘valid’}, \tilde{t} \rangle$  if the corresponding score exceeds the threshold, that is, if  $\beta(\tilde{t}) \geq \gamma$ , and output  $\langle h, \text{‘invalid’} \rangle$  otherwise.
- 

periodicity coefficient out of a set of candidate periods and return it if the corresponding coefficient exceeds the periodicity threshold  $\delta$ , that is, if  $S(\tau) \geq \delta$ .

It is worth noting that using the true autocorrelation  $A_{f_h}(\cdot)$  values corresponding to  $f_h$  is problematic. If the unit of time is sufficiently granular (say, seconds), then it is very likely that the  $f_h$  shifted by its period will not align with itself. For example, the timelines for two consecutive weekly meetings for #mtos would look sufficiently different when zoomed in. In the extreme case, tweets may occur during even seconds for even numbered weeks and during odd seconds for odd numbered weeks so that there is no alignment for a shift by one week, resulting in an autocorrelation of zero.

We address this problem by bounding the frequencies  $\xi_j$  at which we sample the Fourier transform so that the maximum frequency is at most two cycles per day. Then, the approximate autocorrelation  $\tilde{A}(\cdot)$ , loosely speaking, can only see the approximate time at which a tweet happened, to within half a day or so.

#### 4.1.4 Incremental Updates

As more tweets arrive, it is possible to recompute the periodicity coefficients and the estimated period of a hashtag in time proportional to the number of new tweets. Given the Fourier coefficients,  $\hat{f}(1/T_1), \dots, \hat{f}(T_r)$  computed for timeline up to time  $\mathcal{T}$ , and given additional timeline from  $\mathcal{T}$  to  $\mathcal{T}'$ , the new Fourier coefficients can be computed as:

$$\begin{aligned} \hat{f}'(1/T_j) &= \sum_{0 \leq t \leq \mathcal{T}} f_h(t) \cdot e^{-2\pi i t / T_j} + \sum_{\mathcal{T} < t \leq \mathcal{T}'} f_h(t) \cdot e^{-2\pi i t / T_j} \\ &= \hat{f}(1/T_j) + \sum_{\mathcal{T} < t \leq \mathcal{T}'} f_h(t) \cdot e^{-2\pi i t / T_j}. \end{aligned}$$

The remaining steps of recomputing the autocorrelation and finding the highest-scoring period have a running time that depends on the number of candidate periods ( $r$ ) but not the total number of tweets.

## 4.2 Synchronized

The intuition underlying the algorithm for this component is that synchronized meetings usually last for an hour or two, but not for the whole day: most people simply do not have the time to participate and listen to others for very long. Just like in the case of offline physical groups, it is unrealistic to expect online users to participate in a focused meeting and converse with each other for more than a few hours. Hence, we require that a meeting lasts for a short

duration, say, at most 3 hours. Further, for hashtags used for synchronized meetings, the usage of the hashtag predominantly occurs during the meeting window as opposed to outside the meeting.

Algorithm 3 formalizes the above intuition. Given a periodic hashtag  $h$  with its period,  $\tau$  and timeline function  $f$ , we obtain a compressed timeline  $g$  by aggregating activity at a given time offset across different periods. Then, for various candidate time offsets, we check how often  $h$  is used within a window of  $l$  (maximum duration of a meeting) from the candidate time, aggregated across different periods and thereby compute the most likely meeting start time. Then, we check whether there are sufficient number of tweets containing the hashtag during the corresponding meeting window, compared to the total number of tweets containing the hashtag. If yes, we output the most likely meeting start time and otherwise, mark  $h$  as ‘invalid’.

Note that Algorithm 3 requires the period  $\tau$  to exactly match the true period of the hashtag so that meetings across different periods are aligned in the compressed timeline. However, the timeline associated with the hashtag may not correspond to a perfect periodic function and hence the period computed by Algorithm 2 may deviate slightly from the true period. We make use of the fact that group chats on Twitter have routine meetings whose period is in multiple of days, and address this issue by rounding the period computed by Algorithm 2 to the nearest day.

### 4.3 Cohesive

In the final step, our algorithm excludes hashtags that are not cohesive. To measure this quantity, we estimate the communication among the  $k$  most active members who use the hashtag  $h$  during regular, synchronized meetings. Our intuition is that members of a healthy group will look forward to communicating with one another during each meeting. Let  $V$  be the set of  $k$  users who participated in the most meetings of the hashtag. For  $1 \leq i \leq m$ , let  $E_i$  capture the directed communication edges among users in  $V$  during the  $i^{\text{th}}$  meeting. For example, if user  $u \in V$  @-messages two other top users  $v$  and  $w$  during the  $i^{\text{th}}$  meeting, we include the two directed edges  $(u, v)$  and  $(u, w)$  in  $E_i$ . We define the cohesiveness of a hashtag by the average number of edges across all meetings, that is,  $\text{cohesive}(h) := \frac{1}{m} \sum_{i=1}^m |E_i|$ , where  $m$  is the number of meetings for hashtag  $h$ . The larger this average interaction, the more cohesive the group. Given a threshold  $\psi$ , we exclude hashtags where  $\text{cohesive}(h) < \psi$ .

## 5. THEORETICAL ANALYSIS OF ALGORITHM 1

We next prove that hashtags that are close to group chats are accepted by our algorithm and hashtags that are far from group chats are rejected. Specifically, if a group meets sufficiently many times, if group meetings are well-separated, and if a reasonable number of messages are exchanged per meeting, then we prove that our algorithm will accept the hashtag, under certain assumptions (§5.1). On the flip side, we prove that our algorithm will reject hashtags that are not cohesive, or are generated from models of not-regular or not-synchronized hashtags (§5.2).

### 5.1 Algorithm 1 accepts group chats

We next define the notion of a *well-behaved group chat* and show that our algorithm will accept a well-behaved group chat under certain assumptions.

**DEFINITION 5.1.** *A set of tweets forms a well-behaved group chat if all of the following are true.*

- There are  $m$  meetings of duration  $l$  separated by a period  $\tau$ , for some  $m$ ,  $l$  and  $\tau$ . (The  $j$ th meeting interval is  $[j\tau, j\tau + l]$ .)
- During each of the meeting intervals, at least  $\psi$  pairs of the top  $k$  group members exchange messages.
- Tweets are sent at a higher rate within meetings than outside of meetings. That is: at least  $\gamma(n_{\min} + n_-)$  tweets are sent during every meeting, where  $\gamma \geq 1/\tau$  is the threshold of Algorithm 3,  $n_{\min}$  is the smallest number of tweets sent during any meeting, and  $n_-$  is the average number of tweets sent between two adjacent meetings. No tweets are sent before the first meeting or after the last. We denote by  $n_j$  the number of tweets in the  $j$ -th meeting, and by  $n_{\min}$  and  $n_{\max}$  the smallest and largest values  $n_j$ .

We empirically validated that a typical group chat resembles a well-behaved group chat.

**THEOREM 5.2.** *Consider any well-behaved group chat that also satisfies the following properties:*

- (Technical conditions.) *There are at least three meetings. The period  $\tau$  is not shorter or longer than the range of periods considered by Algorithm 2 (periodicity). The duration  $l$  is shorter than half the shortest period considered by Algorithm 2, and also no longer than the duration  $l$  of Algorithm 3 (synchronization).*
- (Quantifying well-behavedness.) *The following inequality holds, where  $\eta = n_{\max}/n_{\min}$ ,  $\rho = n_-/n_{\min}$  and  $\delta < \frac{1}{2}$  is the threshold used by Algorithm 2’.*

$$\frac{1}{2\eta^2} \left( 1 - \frac{2\pi l}{\tau} - \rho \right) > \max \left\{ \delta, 3\rho, \frac{2\pi l}{\tau} + \frac{\eta^2}{4} \left( 1 + 6 \frac{\eta + \rho - 1}{\rho + 1} \right) \right\} \quad (4)$$

- (\*) *The timeline is a step function, where the number of tweets during each second of the  $j$ -th meeting is  $\lambda_j = n_j/l$ , and the number of tweets during any second which is between meetings is exactly  $\lambda_- = n_-/(\tau - l)$ .*

Consider a modified (\*\*) version of Algorithm 2 which computes autocorrelations exactly: call this Algorithm 2’, and the resulting group chats algorithm Algorithm 1’. Then, Algorithm 1’ will accept this chat as a group chat. It will also return the correct meeting start time to within  $l$ , and will report the correct period  $\tau$  with error of at most  $l + \epsilon$ , where  $\epsilon$  is the largest difference between two adjacent periods considered by either the Fourier or autocorrelation parts of Algorithm 2.

*Note 1.* The parts marked (\*) and (\*\*) are added to simplify the proof; we believe a version could be proved which does not have the step function condition (\*) and applies to the true Algorithm 1 (\*\*). (§A.1 contains the only results that depend on (\*) and (\*\*).) Although modifying the algorithm to compute autocorrelation exactly makes the proof simpler, it also forces us to impose a strong condition on the timeline of tweet rates for the given hashtag. The true second-by-second timeline of a typical chat has zero tweets in most seconds, with spikes at certain seconds where one or two tweets occurred, and the exact autocorrelation of such a timeline will typically be close to zero: this is why we are forced to assume the timeline is a step function. Our true implementation of autocorrelation uses a sample of Fourier coefficients, and is therefore insensitive to variations on the order of half a day or less.

*Note 2.* To interpret (4), note that that for a very strongly-structured group chat, we can expect  $\eta$  to be close to 1 (meaning all meetings have similar attendance),  $\rho$  close to 0 (the group’s hashtag is rarely used outside meetings), and  $l/\tau$  to be quite small (meetings are short compared to the time between). Therefore the left side of the equation will be about  $1/2$ , and the right side will be about  $\max\{\delta, 1/4\}$ .

We defer the proof to Appendix A.

## 5.2 Algorithm 1 rejects non-group chats

We next show that, with high probability, our algorithm will reject hashtags that are not cohesive, or are obtained from generative models representing non-synchronized or non-regular hashtags. As our algorithm explicitly rejects non-cohesive hashtags, we focus on the two generative models.

We begin by describing a random process that captures the rhythm of a typical non-synchronized hashtag. Our observations suggest that a typical hashtag has a daily cycle where its usage increases during waking hours and declines during sleeping hours. Such a hashtag has no short (e.g., 1 hour) time window where most messages are exchanged, as corroborated by [21]. Prior to describing the model, we describe what it means for a group’s messages to be diffuse, i.e., not sharply focused on a short time window.

**DEFINITION 5.3** ( $(\gamma, l)$ -DIFFUSE). *We say a probability distribution  $p$  over an interval  $[a, b]$  is  $(\gamma, l)$ -diffuse if there is no length- $l$  sub-interval  $I = [t, t + l]$  such that  $\Pr_{x \sim p}[x \in I] > \gamma$ . A function  $\mu : \mathbf{R} \rightarrow \mathbf{R}^{\geq 0}$  is  $(\gamma, l)$ -diffuse over an interval  $[a, b]$  if the normalized distribution  $\mu / \left(\int_a^b \mu(t) dt\right)$  is. Similarly, we say a timeline  $t_1, \dots, t_m \in [a, b]$  is  $(\gamma, l)$ -diffuse if at most  $\gamma m$  messages lie in any length- $l$  subinterval  $[t, t + l]$ .*

**DEFINITION 5.4.** *In the non-synchronized model, messages are generated according to a Poisson process with a varying rate  $\lambda(t)$ . We assume that  $\lambda(t)$  is periodic over an interval  $[0, T]$  with a period of one day, and is not concentrated in any short daily interval, that is,  $(\gamma, l)$ -diffuse over a daily period.*

We next present a random process that captures non-regular hashtags that do not have periodic meetings but have a fixed rate of meetings on average.

**DEFINITION 5.5.** *In the non-regular model, meeting times  $\mu_1, \dots, \mu_m \in [0, T]$  are sampled according to a Poisson process with a fixed rate  $\lambda$ . Within the  $j$ -th meeting,  $N$  messages are emitted in the range  $[\mu_j, \mu_j + l]$  (the exact times are allowed to be arbitrary and non-random).*

We state the theorem below and defer the proof to Appendix B.

**THEOREM 5.6.** *If a set of messages for a hashtag is generated from the non-synchronized model (Definition 5.4), from the non-regular model (Definition 5.5), or does not satisfy cohesion (§4.3), then Algorithm 1 will reject it with probability  $1 - o(1)$ .*

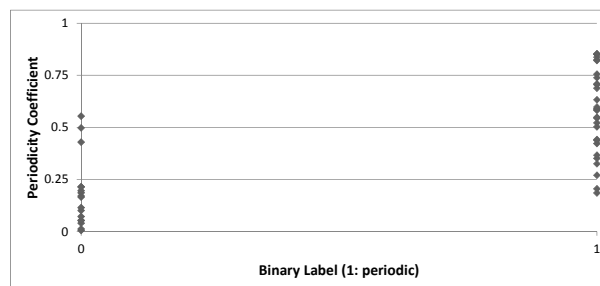
## 6. EXPERIMENTS

We present the results of running our algorithm on over two years of Twitter data. We begin with the experimental setup and then explain we found a periodicity threshold to select periodic hashtags. We describe how we selected synchronized and cohesive hashtags from periodic ones. We give a histogram showing how often groups typically meet. To provide a view into the subjects of these groups, we sample a subset of group chats and report on the distribution of categories into a popular taxonomy. Finally, we show that the number of living groups has grown over time.

## 6.1 Experimental Setup

Our experiments are based on over 28 months of English tweets starting from 9/2010. Given the scale of this data (several petabytes), we implemented our algorithm in the SCOPE language [8] and ran it over a large distributed computing cluster. We first obtained the set of all distinct hashtags used in this timeframe, and the timeline of tweets associated with each hashtag. We also removed under-utilized hashtags, that is, those that received less than 20 tweets, as well as those used by less than 10 users over the duration of the experiment. Then we identified the periodic hashtags, as we describe next.

**Periodicity Threshold:** The method described in §4.1 produces the most likely period and the corresponding periodicity coefficient for every hashtag. We now discuss how we selected the threshold,  $\delta$  to distinguish between periodic and not periodic hashtags. Our methodology was to randomly sample hashtags and then manually label whether the hashtag was periodic, by looking at the timeline of tweets for the hashtag. A purely random sample however favors hashtags with score between 0 and 0.1 since most hashtags are not periodic. Instead, we drew a stratified sample where there were five hashtags with score in  $[0, 0.1]$ , five in  $[0.1, 0.2]$ , and so on. The result of this manual tagging is shown in Figure 3 where the binary label of the hashtag is shown on the x-axis (1 denotes periodic) and the periodicity coefficient is shown on the y-axis. There is a good separation between the periodic and not periodic hashtags when the score is set to  $\frac{1}{4}$ . We also computed the F-measure (harmonic mean of precision and recall) for different choices of the threshold, and confirmed that the maximum is achieved at  $\delta = \frac{1}{4}$ . We then kept all hashtags with a periodicity score  $\geq \frac{1}{4}$ .



**Figure 3: Result of manually labeling a stratified sample of hashtags: 1 denotes periodic and 0 not periodic.**

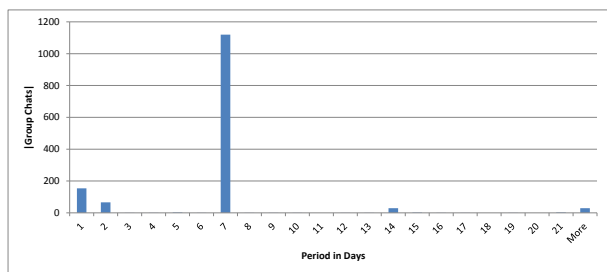
**Group Chats Analysis:** Next we describe how we found the group chats. We selected all hashtags with periodicity threshold at least  $\frac{1}{4}$ , as justified above. In addition, we only kept periodic hashtags that met some minimum number of times, set to 5 in our experiments. In other words, if the period of a hashtag is one week, we require at least five weekly meetings. Without this requirement, groups that meet just twice but spaced say seventeen days apart, appear to have a period of seventeen days.

For each remaining hashtag, we then checked whether the meetings were synchronized during each period. We set the maximum allowed duration of a meeting ( $l$ ) to be two hours and the synchronization threshold,  $\gamma = 0.2$ . These choices require that at least a sizable fraction of tweets ( $\geq 20\%$ ) appear within two hours of the meeting start time.

Finally, we consider cohesion (see §4.3). For each candidate hashtag, we compute cohesion with respect to the  $k$  most committed users measured by meeting attendance, and set the cohesion threshold,  $\psi = k - 1$ . Thus, we restrict ourselves to hashtags for

which the average interaction among the top  $k$  members is at least  $k - 1$ . This structure is realized by, for example, a moderator that routinely converses with  $k - 1$  other members. Note that  $\psi = k - 1$  does not guarantee connectivity, just that  $\psi < k - 1$  guarantees disconnectivity. The latter could reflect a group that is just forming or dissolving or simply not cohesive. In our experiments, we set  $k = 5$ . We are left with 1.4K groups involving 2.3M users (counting each user multiple times if they participate in multiple group chats) which are the subject of this study.

We show the distribution of periods for group chats in Figure 4. For every candidate value of period in days, the chart shows a histogram of the number of group chats with that period. Most group chat hashtags have a weekly period (80% of the chats). However, there are some that meet every day, e.g., those tied to daily radio shows, and some that meet biweekly. For the rest of this section, we focus our discussion on weekly group chats.



**Figure 4: Distribution of periods of hashtags with periodicity coefficient at least  $\frac{1}{4}$ .**

To understand the types of groups we found, we randomly sampled 10% of the recent group chat hashtags and manually categorized the hashtags into the top-level of the Open Directory Project (ODP) taxonomy. For 95% of the group chats, we could categorize the group and a breakdown appears in Table 1. For 5%, we could not categorize based on recent tweets. The table shows that a large fraction of the groups are Arts related, including groups tied to weekly TV shows, radio shows, book clubs, craft clubs, and so on. Science is the next largest category, with education chats dominating the category. In these chats, teachers discuss ways to be more effective educators. There are also a large number of Health-related groups including support groups for coping with addiction and borderline personality disorder. Examples of groups in the Business category include those discussing ways to prevent fraud (e.g., preventing financial crime against seniors), and how to be a digital leader. A small number of groups are Foreign – while we restricted to English tweets only, some groups communicate in multiple languages. In the remaining categories, there are sports enthusiasts and foodie groups. Finally, there are contest-driven chats where the goal is to give away a prize to the person who can answer the most trivia questions.

**Progression of a Group Chat:** Group chats on Twitter follow a particular etiquette. Many chats are moderated and the moderator typically keeps the chat on pace by asking a series of questions. Tweets are typically in the format “q1: ...” for the first question. The moderator typically spaces out questions over an even interval and tries to keep the conversation from deviating off topic. Some of the more successful moderators are able to attract guest tweeters, who are typically celebrities in that area. For example, a wine chat meeting featuring the celebrity Rodney Strong was quite successful (well attended). Moderators typically start the session by tweeting frequent members to join the conversation “chat starting in X min-

Category	% Groups	Examples
Arts	47%	TV/Radio, Writing, Music, Crafts
Science	12%	Education, Agricultural
Health	10%	Addiction, Self-help, Mood Disorder
Business	9%	Preventing Fraud, Digital Leaders
Don't Know	5%	
World	5%	Foreign
Sports	3%	Basketball, Soccer
Society	3%	Better blogger, Rights activist
Recreation	3%	Foodie
Games	2%	Prize-driven

**Table 1: Category distribution of 10% random sample of Twitter weekly group chats.**

utes”. Some chats have web pages announcing the topic of the next meeting, together with archives of previous chats. Users answer the questions posed by the moderator with tweets containing “a1: ...” to answer the first question. The chat ends with the moderator thanking all the participants for a successful chat and announcing the subject of the next chat.

**Comments on Group Chats:** We found the existence of support groups quite surprising. One typically associates support groups with a small number of people sitting around a circle, announcing their name and telling their story. It is hard to imagine giving or receiving support in 140 characters! But, with the same people meeting week after week, getting to know each other better, the platform has proven to be a place for support. Paraphrasing from a mental health chat, users state that social media enables them to access a support network, both those they know in real life, as well as online contacts. They add that having a child with autism spectrum disorder is very isolating.

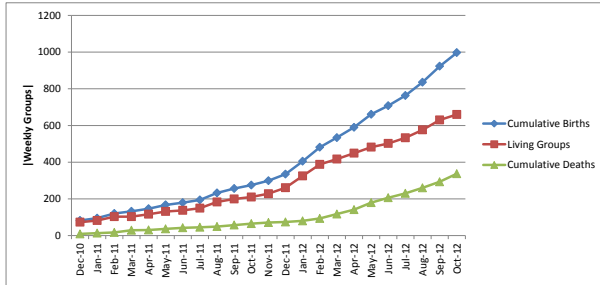
Support groups are successful for a variety of reasons. Empathy is the driving force, with users stating that “there is a family of us out there”. In contrast to offline groups, the success of some online support groups can be attributed to the pseudo-anonymous online communication that is more comfortable than physical group meetings, e.g., people say that they find group sessions hard, that they hate opening up in front of others. In some cases finding others who are in a similar situation is challenging. One user said that he knew no one in his state that was transgendered. His trans world was him, only him. In our experience as observers, we felt that we had accidentally walked into a room full of people sharing personal experiences. No one seemed bothered by the fact that anyone could hear what they were saying.

Passion-related groups are also quite fascinating. In Movie Talk on Sunday, a moderator pre-selects a theme, e.g., suspense movies, and posts 10 questions ahead of time on a website that will be tweeted every ten minutes. Participants tweet answers to questions. The moderator and members retweet the answers they like. In this group, participants derive value from the discussion. For example, we find evidence of a user who decided to give a movie ‘Cabin in the Woods’ a second try because so many in the group felt it was the best movie of the summer of 2012. In other cases, we found people thanking the group for gaining many new followers during the chat. Finally, we even found evidence of two users now dating after meeting on a group chat hashtag.

**Group Chats over Time:** Finally, we ask if group chats are a growing or shrinking phenomenon on Twitter. To answer, for each weekly group, we computed the birth of the group by finding the first weekly meeting where at least 10 members tweeted during the meeting and similarly the death of the group by the last weekly



meeting where at least 10 members tweeted. Since our data is restricted to a time-window, we cannot accurately find the birth date of a group that was born before our data time-window. Also, if a group is born towards the end of our time window, but has not had five meetings yet, we will not be able to find it. Consequently, we throw out the first few months and the last few months of the time range and compute the cumulative number of births over time, the cumulative number of deaths over time, as well as the net number of living groups. The chart is shown in Figure 5. Interestingly, the number of groups has indeed grown over time. However, even though the number of births has increased, so has the number of deaths. We do not understand what causes a group to grow or to die, but it is a great subject for future work.



**Figure 5: The top curve shows the number of weekly group chats born over time, the bottom curve shows the number that died over time, while the middle curve shows the number of weekly living groups over time.**

**Limitations:** While our methods do indeed find group chats, we know that there are some group chats that we miss. For example, groups that meet the first Monday of the month are not found by our method because the separation between meetings could be either four or five weeks. If a group misses many meetings, it may be difficult for our method to find. For example, ski chats are typically on hiatus over the summer. Other group chats that we miss have irregularly spaced meetings, for example, if users agree on a day/time for the next meeting at the end of each meeting. Also, since our processing is hashtag related, our implementation overlooks conversations that happen between users outside the context of the group chat. Finally an ambiguously named group chat such as tchat makes it difficult to tease out the group chat from alternate uses of the hashtag.

## 7. CONCLUSIONS AND FUTURE WORK

We discovered a new phenomenon on Twitter - the use of hashtags to organize periodic, synchronized conversations. We gave a formal definition of a group inspired by definitions from sociology. We developed an algorithm for finding group chats and proved conditions under which the algorithm successfully finds groups. We ran our algorithm on over two years of tweets and discovered, hidden in plain view, 1.4K group chats involving 2.3 million users. The data suggests that most of these groups meet weekly and that the number of groups has been growing over time. Group chats are an unintended use of the Twitter platform. The grassroots nature of these organized groups demarcates a subset of Twitter containing passionate users producing seemingly higher quality tweets.

Many fascinating questions are yet to be answered. We are very curious to understand how these groups form. Since many of the founders/moderators of these groups have public personas, it is in theory possible to ask the founders how they created their group.

We also do not know how these groups grow and eventually die [5]. Previous studies of group formation [2] suggest that groups grow by the number of friends one has in the group. However, we do not know if this explains the growth of all kinds of groups, such as support groups. Other studies suggest the importance of first impressions [18, 3]. The general question of group dynamics [11] in Twitter chats is untouched by the present work. We also do not know why Twitter is the choice platform for these chats. The 140 character limit seems restrictive, but the resulting improved social standing one receives from tweeting in passion-oriented groups may outweigh the character limit.

Some of the users who participate in these chats are quite knowledgeable about the subject matter they are discussing. Understanding and quantifying their expertise level is a promising direction for future work. In addition, it is useful to find ways to summarize a meeting of a group chat [10] in a manner that takes advantage of the structure of a typical conversation. Finally, it will be important to understand if the URLs exchanged in these group chats are of particularly high quality.

The value that these chats bring to the individuals that participate as well as to the community as a whole is not well-understood. We believe that the benefit that users derive cannot be found without the group. For example, the information that users learn from passion-oriented groups may be hard to find without the group. Similarly, the support that a user receives from a support group may be hard to find without the group. We leave the question of measuring the value of these groups as a promising direction for future work.

**Acknowledgments:** We thank Rakesh Agrawal and Abhimanyu Das for thoughtful discussions.

## 8. REFERENCES

- [1] H. Arrow, J. E. McGrath, and J. L. Berdahl. *Small groups as complex systems: Formation, coordination, development, and adaptation*. Sage Publications, 2000.
- [2] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: Membership, growth, and evolution. In *KDD*, 2006.
- [3] L. Backstrom, R. Kumar, C. Marlow, J. Novak, and A. Tomkins. Preferential behavior in online groups. In *WSDM*, 2008.
- [4] P. Bateman, P. Gray, and B. Butler. The impact of community commitment on participation in online communities. *Information Systems Research*, 22(4):841–854, 2011.
- [5] P. Brandtzæg and J. Heim. User loyalty and online communities: Why members of online communities are not faithful. In *INTETAIN*, 2008.
- [6] C. Budak and R. Agrawal. Participation in group chats on Twitter. In *WWW*, 2013.
- [7] D. Cartwright and A. Zander. *Group dynamics: Theory and research*. New York: Harper & Row, 1968.
- [8] R. Chaiken, B. Jenkins, P. Larson, B. Ramsey, D. Shakib, S. Weaver, and J. Zhou. SCOPE: Easy and efficient parallel processing of massive data sets. *Proceedings of the VLDB Endowment*, 1(2), 2008.
- [9] M. Davis. Social networking goes to school. *Education Digest*, 76(3):14, 2010.
- [10] Y. Duan, Z. Chen, F. Wei, M. Zhou, and H. Y. Shum. Twitter topic summarization by ranking tweets using social influence and content quality. In *COLING*, 2012.
- [11] D. Forsyth. *Group dynamics*. Wadsworth Publishing Company, 2009.

- [12] S. Fox. Peer-to-peer healthcare. *Pew Internet & American Life Project*, Feb. 2011. <http://www.pewinternet.org/Reports/2011/P2PHealthcare.aspx>, accessed on April 26, 2012.
- [13] J. D. Greenwood. *The disappearance of the social in American social psychology*. Cambridge University Press, 2003.
- [14] M. Herbert. Why all the chatter about# edchat?. *District Administration*, 48(4):n4, 2012.
- [15] J. Horrigan. Online communities. *Pew Internet & American Life Project*, Oct. 2001. <http://www.pewinternet.org/Reports/2001/Online-Communities.aspx>, accessed November 26, 2012.
- [16] A. Iriberry and G. Leroy. A life-cycle perspective on online community success. *ACM Computing Surveys (CSUR)*, 41(2):11, 2009.
- [17] G. Johnson and P. Ambrose. Neo-tribes: The power and potential of online communities in health care. *Communications of the ACM*, 49(1):107–113, 2006.
- [18] S. Johnson. Should I stay or should I go? Continued participation intentions in online communities. In *Proceedings of Academy of Management Annual Conference*, 2010.
- [19] J. Kleinberg. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397, 2003.
- [20] G. E. Pfander and J. J. Benedetto. Periodic wavelet transforms and periodicity detection. *SIAM Journal on Applied Mathematics*, 62(4), 2002.
- [21] Pingdom. In-depth study of Twitter: How much we tweet, and when. <http://royal.pingdom.com/2009/11/13/in-depth-study-of-twitter-how-much-we-tweet-and-when>, November 2009. Accessed March 17, 2013.
- [22] R. Putnam. *Bowling alone: The collapse and revival of American community*. Simon & Schuster, 2001.
- [23] H. Rheingold. *The virtual community: Homesteading on the electronic frontier*. MIT press, 2000.
- [24] Twitter chat schedule. <https://docs.google.com/spreadsheet/ccc?key=0AhisaMy5TGiwcnVhejNHWNz1T3NvWfVPT3Q4NkIzQzVE> accessed July 4, 2012.
- [25] M. Vlachos, P. Yu, and V. Castelli. On periodicity detection and structural periodic similarity. In *SDM*, 2005.

## APPENDIX

### A. PROOF OF THEOREM 5.2

Notice that Definition 5.1 requires immediately that the any well-behaved group chat pass the cohesion test. Further, any well-behaved group chat will pass the test for being synchronized: Algorithm 3 will accept the chat because  $\beta(0) \geq mn_{\min}/(mn_{\min} + (m - 1)n_-) \geq \gamma$ . The only real difficulty is in showing that (the modification of) Algorithm 2 will accept the timeline of tweets. Lemma A.1 completes this last step, and the remainder of this section is devoted to its statement and proof.

LEMMA A.1 (CORRECTNESS OF ALGORITHM 2). *Let  $f$  be the timeline satisfying the hypotheses of Theorem 5.2. Then the periodicity coefficient of the correct period is at least*

$$S(\tau) \geq (2\eta^2)^{-1}(1 - 2\pi l\tau^{-1} - \rho) \quad (5)$$

and for any period  $\tau' \geq l$  for which  $|\tau' - \tau| \geq l$ , the periodicity coefficient is at most

$$S(\tau') \leq \max \left\{ 3\rho, \frac{2\pi l}{\tau} + \frac{\eta^2}{4} \left( 1 + 6 \frac{\eta + \rho - 1}{\rho + 1} \right) \right\}. \quad (6)$$

PROOF. The periodicity coefficient of a period  $S(\tau)$  is the product of two terms, which we analyze in Sections A.2 (Fourier) and A.1 (autocorrelation). The lower bound (5) on the true period's coefficient follows from Lemmas A.2 and A.7. The bound (6) on other coefficients  $\tau'$  is proved in cases. For  $\tau' \in [l, \tau - l] \cup [\tau + l, 2\tau - l]$ , Lemma A.3 gives  $S(\tau') \leq |A_f(\tau')|/|A_f(0)| \leq 3\rho$ .

The remaining case is that  $\tau' \geq 2\tau - l$ . We begin by approximating the term  $s = 2|\sin \pi\tau\xi|$  from the statement of Lemma A.8, where  $\xi = 1/\tau'$ . By the concavity of the sine function on  $[0, \pi]$ , we have  $\sin \psi\pi \geq \psi$  for  $\psi \in [0, \frac{2}{3}]$ . So for  $\tau' \geq 2\tau - l \geq 3\tau/2$ , we have  $s \geq 2\tau/\tau'$ . If  $\tau' > m\tau$ , then by Lemma A.4 the periodicity coefficient is zero. Otherwise,

$$\left( \frac{m - \tau'/\tau}{m} \right) s^{-1} \leq \left( \frac{m - \tau'/\tau}{m} \right) \frac{\tau'}{2\tau} \leq \frac{m}{8}.$$

So by Lemmas A.4 and A.8,

$$\begin{aligned} & |\hat{f}(1/\tau')||A_f(\tau')|/|\hat{f}(0)||A_f(0)| \\ & \leq \frac{2\pi l}{\tau} + \eta^2 \left( \frac{m - \tau'/\tau}{m} \right) \left( \frac{2s^{-1}}{m} + \frac{3}{2} \cdot \frac{\eta + \rho - 1}{\rho + 1} \right) \\ & \leq \frac{2\pi l}{\tau} + \eta^2 \left( \frac{1}{4} + \frac{3}{2} \cdot \frac{\eta + \rho - 1}{\rho + 1} \right). \end{aligned}$$

□

### A.1 Results about Autocorrelation

Here we state results which say that the autocorrelation of the true period is high, and that the autocorrelations of certain other periods are low. We assume  $f$  satisfies the hypotheses of Theorem 5.2. Proofs omitted for brevity.

LEMMA A.2.  $|A_f(\tau)|/|A_f(0)| \geq 1/(2\eta^2)$ .

LEMMA A.3. *Let  $\tau'$  be a period, and assume meetings in the timeline  $f(t)$  are disjoint from meetings in the timeline  $f(t - \tau')$ . Then  $|A_f(\tau')|/|A_f(0)| \leq 2m\rho/(m - 1)$ .*

LEMMA A.4.  $\frac{|A_f(\tau')|}{|A_f(0)|} \leq \max \left\{ 0, \eta^2 \left( \frac{m - \tau'/\tau}{m} \right) \right\}$ .

### A.2 Results about Fourier Coefficients

Here we state analogous results to those in Section A.1, but about Fourier coefficients. The notation  $f$  will represent the timeline of a well-behaved group chat. Note that for any function  $g$ ,  $|\hat{g}(0)| = |\int_{-\infty}^{\infty} g(x)dx|$ : for timelines, the total number of tweets.

PROPOSITION A.5 (EFFECT OF NOISE).

*Suppose  $g = g_+ + g_-$ . Then, for any frequency  $\xi$ , we have  $|\hat{g}(\xi) - \hat{g}_+(\xi)| \leq |\hat{g}_-(\xi)| \leq |\hat{g}_-(0)|$ .*

PROPOSITION A.6 (EFFECT OF TIMING WITHIN MEETINGS).

*Consider a timeline  $g$  consisting of  $\alpha$  tweets at times  $t_0, \dots, t_\alpha$ , and a distorted version  $g'$  with tweets at times  $t'_0, \dots, t'_\alpha$ , where  $\forall j |t_j - t'_j| < l$ . Note that  $\hat{g}(0) = \hat{g}'(0)$ . (Think of  $g$  with tweets at starts of meetings, and  $g'$  with tweets throughout meetings.) Then, for any frequency  $\xi$ ,  $|\hat{g}(\xi) - \hat{g}'(\xi)| < 2\pi l\xi\alpha$ .*

Combining Propositions A.5 and A.6, we have:

LEMMA A.7 (THE CORRECT FOURIER COEFFICIENT).

$|\hat{f}(1/\tau)| \geq (1 - 2\pi l/\tau - \rho)\alpha$ , where  $\alpha$  is the number of tweets.

Finally, we show:

LEMMA A.8 (THE WRONG FOURIER COEFFICIENTS). *Let  $\xi$  be any frequency, and let  $s = 2\lceil \sin \pi \tau \xi \rceil$ . Then,*

$$|\hat{f}(\xi)| < n_{\min}(2\pi l \xi m + 2s^{-1}) + m(n_{\max} - n_{\min}) + (m-1)n_-.$$

In particular, if  $m \geq 3$ , then

$$|\hat{f}(\xi)|/|\hat{f}(0)| < 2\pi l \xi + 2s^{-1}m^{-1} + 3(\eta + \rho - 1)/2(\rho + 1).$$

PROOF. We will replace  $f$  by a simpler version  $f_{\text{simple}}$ . This version will have no tweets between meetings. Every meeting will have the same number of tweets  $n_{\min}$ , and all the tweets will happen at meeting starts. Fourier coefficients of  $f_{\text{simple}}$  behave well, and we can relate  $f_{\text{simple}}$  to  $f$  using Propositions A.5 and A.6.

Let  $f_+$  consist of tweets during meetings and  $f_-$  the other tweets. Now, change  $f_+$  to have  $n_{\min}$  tweets in every meeting, by removing tweets from meetings that have more. Let  $f_+^*$  be the resulting timeline, and  $f_+^{\text{extra}} = f_+ - f_+^*$ . Notice  $|\hat{f}_+^{\text{var}}(0)| \leq m(n_{\max} - n_{\min})$ . Now, let  $f_{\text{simple}}$  consist of  $n_{\min}$  tweets at the start of each of the  $m$  meetings. By Prop. A.6,  $|\hat{f}_+^*(\xi) - \hat{f}_{\text{simple}}(\xi)| < 2\pi l \xi m n_{\min}$ . Fourier coefficients of  $f_{\text{simple}}$  are geometric series:

$$\begin{aligned} |\hat{f}_{\text{simple}}(\xi)| &= \left| \sum_{k=1}^m n_{\min} e^{-2\pi i(k\tau)\xi} \right| = n_{\min} \left| \sum_{k=1}^m (e^{-2\pi i\tau\xi})^k \right| \\ &= n_{\min} \frac{|(e^{-2\pi i\tau\xi})^m - 1|}{|e^{-2\pi i\tau\xi} - 1|} \leq \frac{2n_{\min}}{|e^{-2\pi i\tau\xi} - 1|} = \frac{2n_{\min}}{s} \end{aligned}$$

Then  $|\hat{f}_+^*(\xi)| < n_{\min}(2\pi l \xi m + 2s^{-1})$ . By Proposition A.5,  $|\hat{f}(\xi)| < n_{\min}(2\pi l \xi m + 2s^{-1}) + m(n_{\max} - n_{\min}) + (m-1)n_-$ .  $\square$

## B. PROOF OF THEOREM 5.6

The case of a non-coherent set of messages is trivial (our algorithm explicitly rejects such hashtags). Lemmas B.3 and B.4 complete the proof in the case of the non-synchronized and non-regular models.

### B.1 Non-synchronized messages

For a rate function  $\lambda(t)$ , let  $|\lambda|_1 = \int_{-\infty}^{\infty} \lambda(t) dt$ .

We will think of Algorithm 3 as having two parts. First, a hashtag's timeline is *compressed* over an interval  $\tau$ : each message time  $t_i$  is replaced with a message time  $0 \leq t'_i < \tau$  by subtracting a multiple of the period  $\tau$ . Second, the algorithm checks whether the resulting timeline ( $t'_i$ ) is  $(\gamma, l)$ -diffuse, and if so, classifies the hashtag as a non-group chat. Lemma B.1 will show that the timeline remains diffuse after the compression step, and then Lemma B.2 shows that Algorithm 3 will reject the hashtag with high probability.

LEMMA B.1. *Let  $\lambda$  be a function which is periodic with period  $\sigma$  over the interval  $[0, m\sigma]$ . Assume  $\lambda$  is  $(\gamma^*, l)$ -diffuse over the interval  $[0, \sigma]$ . Given a possibly longer period  $\tau \geq \sigma$ , define the compressed function  $\lambda_\tau(t) = \sum_{i=0}^{\lceil m\sigma/\tau \rceil} \lambda(t + i\tau)$ . Then  $\lambda_\tau$  is  $(2\gamma^*, l)$ -diffuse over  $[0, \tau]$ .*

We omit the proof due to space constraints.

LEMMA B.2. *Suppose  $\lambda(t)$  is  $(\gamma^*, l^*)$ -diffuse on an interval  $[0, \tau]$ , and  $\gamma > \gamma^*$  and  $l < l^*$ . If timeline  $t$  is sampled with a Poisson process of varying rate  $\lambda(t)$ , then  $t$  is  $(\gamma, l)$ -diffuse with probability  $1 - O(1)2^{-\Omega(|\lambda|_1)}$ . (The coefficients in the  $O$  and  $\Omega$  depend on  $\gamma^*$ ,  $\gamma$ ,  $l^*/\tau$  and  $l/\tau$ .)*

PROOF. Let  $\epsilon = l^* - l$ . We divide the interval  $[0, \tau]$  into overlapping intervals  $I_1 = [0, l^*]$ ,  $I_2 = [\epsilon, l^* + \epsilon]$ ,  $\dots$  (If an interval  $[a, b]$  has  $b > \tau$ , we consider it to “wrap around”: we replace it by  $[0, b - \tau] \cup [a, \tau]$ , since Algorithm 3 works modulo  $\tau$ .) There are a total of  $\tau/\epsilon$  such intervals. Notice that any interval of length  $l$  must fall completely into one of these intervals: so it suffices to show that at most a  $\gamma$  fraction of messages will fall in any of the intervals  $I_j$ , and then take a union bound, losing a factor of  $\tau/\epsilon$ .

We can think of the message times as being generated as follows: first, the total number of messages  $\alpha$  is sampled from a Poisson distribution with rate  $|\lambda|_1$ , and then each message time  $t_1, \dots, t_\alpha$  is sampled independently from the probability distribution  $\lambda(t)/|\lambda|_1$ . Then with high probability, we have  $\alpha > |\lambda|_1/2$ . A Chernoff bound shows that at most a  $\gamma$  fraction of these will fall in a given interval  $I_j$  with probability  $2^{-\Omega(\alpha)}$ .  $\square$

Combining Lemmas B.1 and B.2, we have the following:

LEMMA B.3. *Let  $\lambda$  be a rate function that is periodic with a period of one day on the interval  $[0, m \text{ days}]$ , and is  $(\gamma/2 - \epsilon, l + \epsilon)$ -diffuse on the interval  $[0, 1 \text{ day}]$  (where  $\gamma$  and  $l$  are parameters of the algorithm, and  $\epsilon$  is any positive constant). Then with probability  $1 - O(1)2^{-\Omega(|\lambda|_1)}$ , a timeline sampled with rate  $\lambda(t)$  will be rejected by Algorithm 3 for any period  $\tau$  which is at least one day. (The coefficients in the  $O$  and  $\Omega$  depend on  $\epsilon$  and the parameters of the algorithm.)*

### B.2 Non-regular messages

LEMMA B.4. *Let  $f$  be the timeline of a set of messages generated according to the non-regular model. Assume the threshold  $\gamma$  of Algorithm 3 is strictly greater than  $2l/\tau_{\min}$ , where  $\tau_{\min}$  is the shortest period considered by Algorithm 2. Then with probability  $1 - O(1)2^{-\Omega(T^\lambda)}$ , Algorithm 3 (and therefore Algorithm 1) will reject  $f$ . The coefficients in the  $O$  and  $\Omega$  depend on  $\gamma$ ,  $l$  and  $\tau_{\min}$ .*

PROOF. We follow a similar strategy to the proof of Lemma B.2, except that in order to bound the number of messages that occur in an interval  $I = [a, a + l^*]$ , we instead bound the number of meetings that start at any point in the interval  $[a - l, a + l^*]$ . This is sufficient, since only those meetings can have any messages in the interval  $I$ .

First, let  $\tau \geq \tau_{\min}$  be the period found by Algorithm 2. Set  $l^* = \gamma\tau_{\min}/2 > l$ , and set  $\epsilon = l^* - l$ . As before, we divide  $[0, \tau]$  into  $\tau/\epsilon$  different intervals  $I_1 = [0, l^*]$ ,  $I_2 = [\epsilon, l^* + \epsilon]$ ,  $\dots$ . It is enough to show that after compression (mapping each time to one in the interval  $[0, \tau]$ ) at most a  $\gamma$  fraction of the messages will fall in any interval  $I_j$ ; then we complete the proof by taking a union bound.

To bound the number of tweets that arrive in an interval  $I_j = [a, a + l^*]$ , it is enough to bound the number of meetings that start in the interval  $I'_j = [a - l, a + l^*]$ , since only such meetings can contribute messages to the interval  $I_j$ . This can be done in the same way as in the proof of Lemma B.2.  $\square$