Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media

Timing Tweets to Increase Effectiveness of Information Campaigns*

Onkar Dabeer and **Prachi Mehendale**
School of Technology and Computer Science
Tata Institute for Fundamental Research
Mumbai, India

Aditya Karnik and **Atul Saroop**
India Science Laboratory
GM Global R&D
Bangalore, India

Abstract

Microblogging websites such as Twitter are increasingly being used by businesses/campaigners for timely dissemination of information to their followers. The diffusion of a tweet depends on several factors: the *activity* of the follower nodes, the *responsiveness* of follower nodes to tweets from the source node, the *out-degree* of the follower nodes, the *content* of recent related tweets seen by the follower node, etc. Using such factors, in this paper, we propose a framework to measure the effectiveness of an information campaign over Twitter. We consider a positive as well as a negative metric to measure the impact of a tweet: while retweets are used to measure the positive impact, the lack of a timely response from an *active* follower node is taken as a potential negative impact. We investigate the scheduling of tweets to increase the net positive impact while keeping the net negative impact below a desired level. We propose and study several scheduling algorithms by casting the problem in a Markov Decision Process (MDP) framework. In order to compare our algorithms, we estimate the model parameters from tweet data collected using the Twitter API from an arbitrarily selected node and its 6837 followers over several months. For this dataset, we find that if successive tweets in the campaign are novel, then substantial gains over user activity based scheduling can be obtained by scheduling tweets in time slots where the ratio of the expected positive and negative metrics is high. We call this the MaxRatio policy and we show that it is optimal under certain conditions. In cases where we are not certain about the response of users to successive related tweets, we identify another algorithm (which we call MaxReach) as a robust alternative.

1 Introduction

Twitter has rapidly emerged as a tool for timely dissemination of information to a wide audience. It provides businesses an opportunity to directly inform end users about their products/services and to understand their opinions. Consequently, a large number of businesses have a Twitter ID, and many have several tens of thousands of followers. While communication over Twitter is often effective, it is not perfect. On Twitter, every user is a source of tweets, and therefore, a typical user often gets flooded with a large number of tweets. A user reads tweets as per her convenience, and at any given time, she may read only a fraction of the tweets. Tweets that miss this window of opportunity are not read, and thus, a tweet reaches only a fraction of the potential readership.

The importance of timing tweets for maximizing their exposure is well acknowledged by social media experts: bad timing could mean failure to create buzz and may translate into loss of click-through-rate and even sale. Based on a study of more than 1.2 billion tweets over two months, (Sysomos 2010) reports that about 90% of all retweets happen within the first hour of the original tweet. This suggests that the time at which a tweet is sent out critically influences the diffusion of the tweet. "Schedule-your-tweets" is a feature in numerous Twitter tools (for example, (Socialoomph 2010)). A generic recommendation is to send tweets during periods of high activity such as around lunch time. But such a viewpoint ignores several key aspects of Twitter. First - all users are not equal and some have (lot) more followers than others. Second - due to the lossy nature of tweets, it is important to spread the message over time, but repeating a message carries the risk of irritating users who read multiple copies. Hence a delicate balance between the *reach* and *irritation* of an information campaign has to be achieved. Third - if the source node follows its follower node, then it knows if the follower node has responded to its original tweet. Such *feedback* may help in future timing decisions. Fourth - all followers are not equally responsive and it is important to account for this responsiveness.

In this paper, we take such a nuanced perspective. We propose a method to measure the effectiveness of a campaign that accounts for the above mentioned factors, and we address the problem of scheduling a tweet campaign to increase its effectiveness. For our study, we use a mix of a simple theoretical (but rather natural) model and actual tweet dataset collected from 6837 followers of an arbitrarily selected root node over several months. One key aspect of our model, described in Section 2, is the incorporation of an irritation state for a user: if a user does not respond to a received tweet, then her irritation level increases, which in turn may reduce her chances of responding to subsequent tweets in the information campaign. It is difficult to pin down exactly how the response probabilities of a user depend on the

*The work of Onkar Dabeer was supported by a research grant from India Science Laboratory, GM Global R&D.

irritation level, since these probabilities also depend on the content of the tweets. Hence, we consider two extreme models to gain insight: a model in which the irritation level has no impact on response, and another model in which a user does not respond once she is irritated. The parameters in our model are estimated from real data collected from Twitter. The data collection and parameter estimation procedures are described in detail in Section 3.

Given our model, it is clear that the process of scheduling has two effects: the positive impact is that the tweets reach a large number of users, while the negative impact is that in the process we may irritate some users. In Section 4, we propose a metric that balances these two effects and we study the optimal scheduling problem using Markov decision processes (MDPs) (Bertsekas 1995). We compare the performance of different (optimal as well as sub-optimal) scheduling strategies based on simulations. Based on several factors, under model uncertainty, we identify an offline policy we call MaxReach as a good choice. In situations where successive tweets are novel, a new offline scheduling policy (called MaxRatio) provides further improvement. In addition, our work also gives insight as to when feedback from followers may not be effective. We discuss the main conclusions and related future directions in Section 5.

To the best of our knowledge, this is the first paper that systematically studies the problem of scheduling tweets to maximize their effectiveness, and thereby, puts the art on a solid ground. Our technique of casting the scheduling task as a MDP/dynamic programming has a long history and it has been employed in diverse fields: manufacturing (Bomberger 1966), scheduling jobs on a computer (Sahni 1976), wireless transmission (Su, Tassiulas, and Tsotras 1999), (Berry and Gallager 2002), etc. While all these works use the same broad mathematical framework (MDP), they differ significantly in the details. In particular, our specific model is suited for Twitter and does appear to have arisen in the literature before.

Our scheduling strategies are simple to implement. They can be tuned to balance reach and irritation at desired levels. They can be easily implemented through "schedule-your-tweets" feature of many Twitter tools. In the process of developing these strategies, we have also studied the activity patterns of users, responsiveness of followers to the root node, irritation level of users to repetitive tweets, etc., which may also be of interest in other contexts. In particular, we bring to attention two parameters, namely, responsiveness of followers and their irritation level, which play a vital role in campaign effectiveness. While our primary motive is to propose effective strategies for information campaigns, our study also has a secondary motivation. As more nodes on Twitter employ data analytics to improve their impact, the dynamics of the social network evolves in a complex manner. Our hope is that the various effective strategies we develop in this paper will also aid future studies on the dynamics of Twitter.

2 A Probabilistic Model for Response

Consider a root node which aims to run an information dissemination campaign over Twitter. The node has several followers, say N, and each of its followers has its own followers. A tweet sent by the root reaches all its followers. But not all followers read the tweet. Amongst those who read the message, some may retweet or send out a related tweet to their followers, which reflects their interest in the tweet. The original tweet from the root node thus diffuses through the network through such retweets. The analysis of the impact of a tweet requires the analysis of the Twitter network around the root node. As a first step, in this paper, we focus only on the followers of the root node while studying strategies for scheduling of tweets. (This restriction is imposed to make the data collection easier, but can be removed in future.) While tweets can be sent at any time, we consider time slots of one hour duration. We also assume that user activity parameters described below have weekly periodicity, which allows us to focus on $24 \times 7 = 168$ time slots. Suppose the message campaign is to be completed in time T. For example, over one week, we may be interested in introducing key features of a product. The root node that sends these tweets has control over two quantities: the timing of the tweets and the content of the tweets. In this paper, we study the timing of tweets with the aid of the following model for the node behavior.

Consider the N followers of the root node. A node is said to be active at time t if it has tweeted in that time slot. Suppose at time t, $A_n(t) = 1$ if user n is active, and $A_n(t) = 0$ if the user is inactive. Let $M(t) = 1$ if a tweet is sent by the root node at time t and let $M(t) = 0$ otherwise. We say that node n *receives* a tweet sent at time t if it is active at time t, and the corresponding indicator variable is $R_n(t) = A_n(t)M(t)$. If a tweet is received by a node, then it might retweet/or send a related message to its followers. We denote the response of node n by the indicator variable $F_n(t)$, which is 1 if node n forwards a tweet received at time t and is 0 otherwise. While $F_n(t) = 1$ indicates that node n is interested in the tweet, $F_n(t) = 0$ can occur if the tweet is not read, or if it is read but there is not much interest in it. Let $U_n(t) = d_n F_n(t) R_n(t)$, where d_n is the out-degree of node n (but in general could be any other non-negative weight). We note that the response of the node may be much later than the time at which the tweet is received, but for simplicity, we consider only those responses that occur in the same time slot. Since we are interested in spreading the message to as many users as possible, a metric we may want to maximize is

$$\bar{U} = \sum_{n=1}^{N} \sum_{t=1}^{T} U_n(t) = \sum_{n=1}^{N} d_n \sum_{t=1}^{T} F_n(t) A_n(t) M(t).$$

Since the activity $A_n(t)$ and the response $F_n(t)$ of a node is not under the control of the transmitter, it is clear that $\bar{U}$ is maximized by flooding, that is, $M(t) = 1$ for all t. Such flooding however has negative effects. Suppose a node receives a message, but does not respond to it. Then it is likely that the node is not interested in the message. Moreover, further reception of similar messages may be construed as spam and may cause the node to i) start ignoring messages from the root node, which may reflect poorly on future campaigns, ii) and in the extreme case it may stop following the root node. This negative effect of broadcasting similar tweets several times can be accounted for as follows.

We assign each node a state $S_n(t) \in \{0, 1, 2, ...\}$, which represents the *degree of irritation*[1] of the node n at time t. All nodes start in the zero state: $S_n(0) = 0$ for all n. If a node receives a tweet ($R_n(t) = 1$) but does not forward it ($F_n(t) = 0$), then $S_n(t+1) = S_n(t) + 1$; otherwise $S_n(t+1) = S_n(t)$. From the above description, we see that

$$\begin{aligned} S_n(t+1) &= S_n(t) + M(t)A_n(t)(1 - F_n(t)) \\ &= f\left(S_n(t), M(t), W_n(t)\right), \end{aligned} \tag{1}$$

where $W_n(t) := [A_n(t), F_n(t)]$ and the function $f(s, m, w)$ is defined by the last equality. Thus the next state is a function of the current state, the current action $M(t)$ of the root node, and a driving signal $W_n(t)$ not under the control of the root node. A measure of the negative effect of sending tweets repeatedly is the net irritation level at the end of the campaign:

$$\bar{S} = \sum_{n=1}^{N} d_n S_n(T+1),$$

where we have used the same weights as in $\bar{U}$ to make the problem more tractable.

In order to consider average performance metrics, we next describe the probabilistic relationships between various indicator variables described above. For convenience let

$$\boldsymbol{S}(t) = [S_1(t), \ldots, S_N(t)]\,.$$

Below, we discuss the assumptions and their implications one-by-one. (If the reader is not interested in mathematical details, then he may just read the definitions of $a_n(t)$ and $\beta_{n,s}$ below, and proceed to the next section.)

A1) We assume that the activity variables $\{A_n(t)\}$ are independent across n as well as t and

$$a_n(t) = E[A_n(t)] \tag{2}$$

is known for all n, t. (In Section 3, we describe how $a_n(t)$ can be estimated from real data.) We also assume that $\{A_n(t)\}$ is independent of the state process $\{\boldsymbol{S}(t)\}$. Our mathematical formulation holds even if we consider a Markov model for $\{A_n(t)\}_{t\geq 1}$. But in practice this entails a higher number of parameters to be estimated, and hence we prefer the simpler independent evolution model.

A2) Given $M(t) = 1, A_n(t) = 1$ and the current irritation state $S_n(t)$, the response $F_n(t)$ is independent of the past:

$$\{A_m(s), M(s), S_m(s), s \leq t-1, 1 \leq m \leq N\}.$$

In words, this means that the response of node n (given that the root node sent a tweet and node n is active) depends only on the current state. The probability that node n forwards a tweet received at time t given that it is in state s is denoted by $\beta_{n,s}$, that is,

$$\beta_{n,s} = P(F_n(t) = 1 | R_n(t) = 1, S_n(t) = s). \tag{3}$$

We expect these probabilities to be non-increasing in the irritation level s. These probabilities not only depend on the user (Boyd, Golder and Lotan 2010), but we also expect them to depend on the composition of the tweets: similar tweets are expected to be met with $\beta_{n,s}$ that decreases rapidly with s, while dissimilar tweets may see almost constant $\beta_{n,s}$. The estimation of $\beta_{n,s}$ from real data also appears to be difficult. Therefore, to gain insight, we focus on the two extreme cases:

- **State Independent Response Probability (SIRP):** $\beta_{n,s} = \beta_n$ for all s;
- **No Response Under Irritation (NRUI):** $\beta_{n,0} = \beta_n \neq 0$, $\beta_{n,s} = 0$ for all $s \geq 1$.

  Note: Even though we focus on the above two choices for $\beta_{n,s}$, we note that our mathematical formulation and analysis is applicable to any arbitrary $\beta_{n,s}$.

A3) If the root node follows user n, then it receives feedback from this node. Many of our strategies do not use such feedback, but some may utilize it. For strategies which do use such feedback, we assume that $S_n(0) = 0$ for all n, and that at time t, we known $\{A_n(s), F_n(s), M(s)\}_{s\leq t-1}$, which exactly determines $S_n(t)$. We note that for strategies that do use such information, the focus is on a few important users, and hence the overhead of obtaining such information may be manageable.

A4) The control $M(t) = \mu_t(\boldsymbol{S}(t))$, where the mapping μ_t is to be designed. An offline strategy does not depend on $\boldsymbol{S}(t)$. We consider both offline as well as online strategies.

Under the above assumption, we see that $\boldsymbol{S}(t)$ is Markov process[2], which can be influenced with the control signal $\{M(t)\}$. Our goal is to choose the control signal (that is the transmission schedule) that maximizes the *mean reach* $E[\bar{U}]$ while maintaining the *mean irritation level* $E[\bar{S}]$ below a desired level. This problem is formulated and addressed in Section 4 . But first, in the next section, we describe estimation of the parameters $\{a_n(t), \beta_n\}$.

3 Estimation of Model Parameters

In this section, we describe how we estimate $a_n(t)$ and β_n from Twitter data. Our dataset consists of tweets written by the followers of an arbitrarily chosen root node, whose details are not provided for privacy reasons. The total number of followers of the root node is 11,916. Of these, only 6,837 have set their security settings so that their tweets are visible to the external world. We collect tweets posted by these users and the average number of tweets per user is around 800. The time span of these tweets range from few months to few years depending upon the activity rate of the user.

The tweets were collected using Twitter API (Twitter 2011) implemented in Perl programming language. With each tweet, Twitter returns a `Tweet_ID`, the tweet text, the posting time of the tweet, and a

[1]In Section 4, we introduce a scaling parameter to control the importance given to this degree of irritation in relation to the reach of the tweet.

[2]For a Markov process, given the current state, the future and the past states are independent (Durrett 1996).

tweet_in_reply_to_status_ID field, among other details. The tweet_in_reply_to_status_ID field indicates the Tweet_ID of the original tweet to which the current tweet is either a reply to, or a retweet of. The field is populated only if the reply-to or retweet functionality of Twitter is used. The field does not get populated if the tweet is an original tweet, or if the users cut-and-paste the content of the original tweet into a new tweet.

While estimating the parameters, we need to ensure that the data used is related to the current behavior of the user and also that sufficient amount of data is available for reliable estimation. Hence, we place some restrictions and consider only a subset of the followers. This process is described in detail below.

Estimating $a_n(t)$: We say user n is active in slot t if she tweets in the slot t. To estimate the activity probability $a_n(t)$, we consider the latest 26 weeks (which roughly corresponds to six months). In each of these weeks, we look at slot t and check if user n was active in the slot. Let $N_n(t)$ be the number of times user n is active in the past 26 weeks in slot t. Then our estimate of $a_n(t)$ is simply $N_n(t)/26$.

We note that we ignore users who are active for a period of less than six months, since it is not possible to estimate the activity probability reliably for them. Also, for users with activity periods longer than six months, we only consider the recent six months. Thus we try to ensure that we have enough samples to estimate reliably, but at the same time we have a short enough time window to obtain estimates relevant to the current behavior.

Estimating β_n: We recall that under SIRP as well as NRUI, β_n is the probability that user n responds to the *first* tweet (in the campaign) that she receives from the root. To estimate β_n, we make the assumption that the tweets sent out by the root node over the duration of interest are *independent* in the sense that they do not contain correlated information. In this case, we expect the irritation level of the users to not change from tweet-to-tweet. Hence, we can estimate β_n by simply counting the number of tweets sent by the root node when user n is active (say $N_{1,n}$) and the number of responses (say $N_{2,n}$). To find $N_{1,n}$, we count every root node tweet for which user n is found to be active in a duration of one hour after the root node tweet. The choice of an hour stems from previous studies, which suggest that more than 90% of retweets occur in one hour after the original tweet (Sysomos 2010). To count $N_{2,n}$, we need to define what we mean by *response*. From the available data, we observe that often users do not use the retweet facility for responding to the tweets. Users appear to cut-and-paste the original tweet and then add some comments of their own. Based on this, we say that a tweet (from the root node) elicits a *response* from user n, if in a duration of one hour from the time of the tweet, the user sends a tweet satisfying any one of the following.

1. The tweet sent by the user contains the root node tweet ID in the tweet_in_reply_to_status_id field.
2. The content of the tweet sent by the user exactly matches with that of the tweet sent by the root node.
3. The root node tweet contains more than 50 characters and the tweet sent by the user contains the content of the tweet by the root node. The condition of minimum 50 characters ignores small messages that often correspond to individual exchanges.

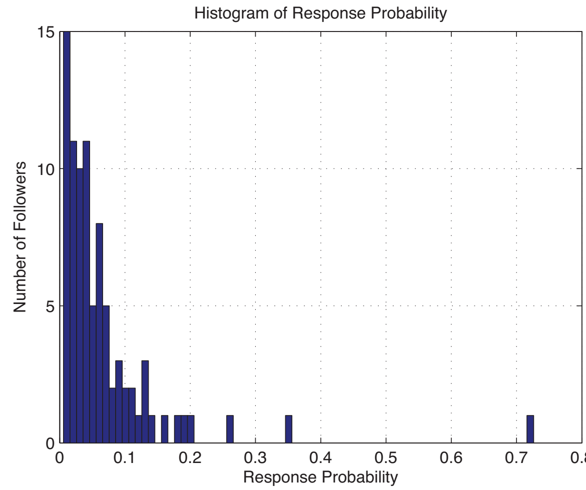

Figure 1: Histogram of response probabilities for 86 followers

While there is a large number of followers of the root node, only a small fraction, a total of 291, have responded to the root node in the last six months.

To ensure that our estimate $\hat{\beta}_n = N_{2,n}/N_{1,n}$ is reliable, we consider the following procedure. If our observations are i.i.d., then our estimate has mean β_n and variance $\beta_n(1-\beta_n)/N_{1,n}$. Ideally, we want the ratio of the mean to the standard deviation

$$\sqrt{\frac{N_{1,n}\beta_n}{(1-\beta_n)}}$$

to be large to ensure reliable estimates. Since we do not know the true β_n, we replace it with $\hat{\beta}_n$ in the above metric, and choose only those users for which the estimated metric is more than $\sqrt{3}$. (We chose this value for our dataset since higher values lead to very few users of interest.) In addition, we ignore users with activity period of less than six months. The users not chosen by this procedure are treated to have zero response probability. As a consequence, we rule out a number of followers with very few responses (indicating a low response probability, which is difficult to estimate reliably, and also has little impact on performance). But we also capture a few relatively low-activity users with high response probabilities. After this shortlisting we get a list of 86 significant users, which we use to report our results. In Figure 1, we show the histogram of the response probabilities estimated from data.

4 Scheduling Strategies: MDP Framework and Simulations

In this section, we develop different scheduling strategies. We start by formulating the general scheduling problem. Then, we consider the specific cases of SIRP and NRUI respectively. In the end, we summarize our results and discuss the practical implications.

We recall that under assumption A1)-A4), the process $\{\boldsymbol{S}(t)\}$ is Markov with observable states. Our aim in this section is to find scheduling strategies that minimize the cost function

$$C(\lambda) = \lambda E[\bar{S}] - E[\bar{U}],$$

where $\lambda > 0$ is a Lagrange multiplier. The constant λ allows us to tradeoff the mean reach $E[\bar{U}]$ and the mean irritation level $E[\bar{S}]$. At one extreme, $\lambda = 0$ leads to the flooding with maximum reach (U_{max}) and also maximum irritation (I_{max}), while the other extreme of $\lambda \to \infty$ leads to no transmissions with zero reach and irritation. For our dataset, working with only the $N = 86$ significant users, we get,

$$U_{max} = 77,588.21, \quad I_{max} = 1,869,859.75.$$

In practice, λ can be chosen to obtain a desired number of total transmissions.

In the Appendix, we show that the cost can be written in the standard form of an MDP and the optimal control $\{M(t)\}$ can be computed by the dynamic programming (DP) algorithm. (For an introduction to MDP and DP, see (Bertsekas 1995).) The DP algorithm works backwards: first we compute the optimal action (as a function of the current state) at the terminal step T, then at $T-1$, and so on. The best action at time t (given the optimal actions for time greater than t) is obtained by minimizing the expected future cost, and the corresponding optimal cost at time t is referred to as the value function. Since the campaign ends at time T, the value function at time $T+1$ is $J_{T+1}(\boldsymbol{s}) = \lambda \boldsymbol{d}^t \boldsymbol{s}$. We show in the Appendix that this entails the following optimal choice of $M(T)$:

$$M(T) = 1 \text{ if and only if} \\ \sum_{n=1}^{N} d_n a_n(T)\big(\lambda - (1+\lambda)\beta_{n,s_n}\big) < 0. \quad (4)$$

Having found the optimal value at time T, the dynamic programming algorithm then proceeds to find the value function at time $T-1$, which also yields the optimal value of $M(T-1)$. Without any further assumption, there is no simplification, and hence we consider the special cases of SIRP and NRUI below.

State Independent Response Probability (SIRP)

Consider the case that $\beta_{n,s} = \beta_n$ for all n, s, that is, the state of a user does not affect his/her probability of response. Under this assumption, we see that $M(T) = 1$ if and only if

$$\frac{\sum_{n=1}^{N} d_n a_n(T)\beta_n}{\sum_{n=1}^{N} d_n a_n(T)} > \frac{\lambda}{1+\lambda} =: c.$$

Thus the optimal value of $M(T)$ does not depend on the state and the corresponding value function (see the Appendix) $J_T(\boldsymbol{s}) = \lambda \boldsymbol{d}^t \boldsymbol{s} + c'$, where

$$c' = \sum_{n=1}^{N} d_n a_n(T)\big(\lambda - (1+\lambda)\beta_n\big),$$

does not depend on the state. As a consequence, the optimization problem for finding $J_{T-1}(\boldsymbol{s})$ and the optimal value of $M(T-1)$ is same as the one we just solved. Repeating this argument we get that for the optimal strategy

$$M(t) = 1 \text{ if and only if } \frac{\sum_{n=1}^{N} d_n a_n(t)\beta_n}{\sum_{n=1}^{N} d_n a_n(t)} > c.$$

Thus the optimal strategy is a static strategy, that is, it does not depend on the state and can be determined offline. In particular, this means that we do not have to track the state evolution and need not follow different followers to collect information such as $A_n(t), F_n(t)$. The constant $c \in [0,1]$. For $c = 0$, we get the flooding strategy, while for $c = 1$, we get no transmissions. The relationship between the number of transmissions and c is not straightforward, and in practice, in order to control the total number of transmissions, we may implement the optimal strategy in an alternative fashion. Let t_0 be the total number of transmissions needed. Then we choose the slots with the highest t_0 values of the ratio

$$\text{ratio}(t) = \frac{\sum_{n=1}^{N} d_n a_n(t)\beta_n}{\sum_{n=1}^{N} d_n a_n(t)}.$$

We call this the **MaxRatio policy**. We note that the numerator is the mean reach at time t, while the denominator is the sum of the mean reach (say $\bar{U}(t)$) and mean irritation at time t (say $\bar{S}(t)$), that is,

$$\text{ratio}(t) = \frac{\bar{U}(t)}{\bar{U}(t) + \bar{S}(t)} = \frac{\frac{\bar{U}(t)}{\bar{S}(t)}}{\frac{\bar{U}(t)}{\bar{S}(t)} + 1}.$$

Thus equivalently, the optimal policy transmits in the time slots corresponding to top t_0 values of the mean reach to mean irritation ratio.

We next compare the optimal MaxRatio policy with the following two policies.

- **Maximum Activity Policy (MaxAct):** Transmit in the top t_0 slots having the highest activity $\sum_{n=1}^{N} d_n a_n(t)$. This is related to the common practice of using the most active periods.
- **Maximum Reach Policy (MaxReach):** Transmit in the top t_0 slots having the highest reach $\sum_{n=1}^{N} d_n a_n(t)\beta_n$.

In Figure 2, for MaxReach and MaxRatio, we show the normalized mean reach $E[\bar{U}]/U_{max}$ and the corresponding percentage reduction in irritation (w.r.t. MaxAct) for values of t_0 ranging from 0 to $T = 168$. We have used the parameter values estimated in Section 3. We see that MaxAct and MaxReach are very close. At a normalized reach of 0.25, MaxAct and MaxReach both have a normalized mean irritation of about 0.27, while the optimal policy has mean irritation level of about 0.22, which is a reduction of about

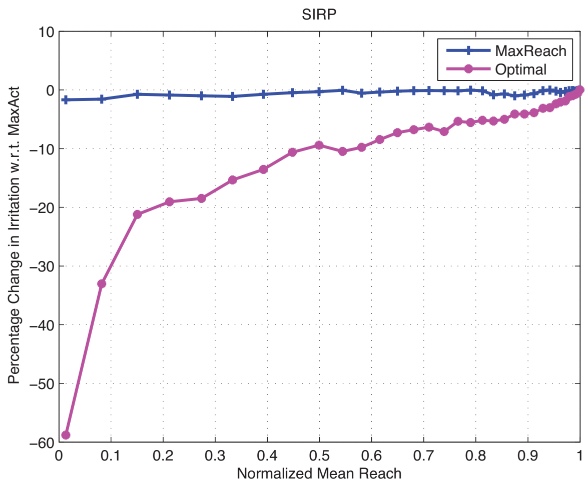

Figure 2: Percentage reduction in irritation (w.r.t. MaxAct) for the same reach by using MaxRatio and MaxReach under SIRP.

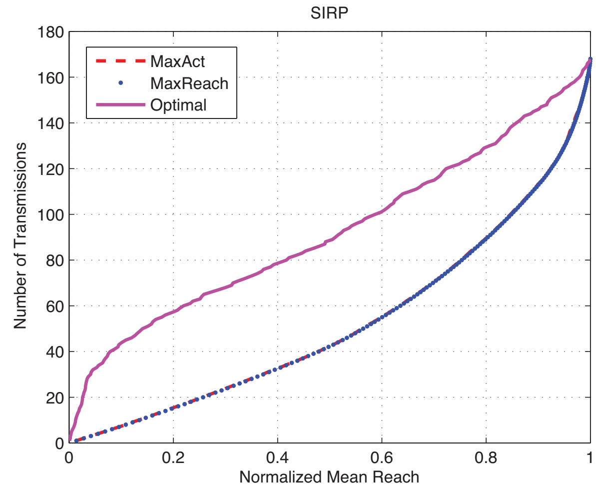

Figure 3: The number of transmissions needed to attain a desired reach for the different schemes under SIRP.

18.5%. If we allow a large number of transmissions, then, as expected, the gain of the optimal policy is not much. For a smaller number of transmissions, the gain of the optimal MaxRatio policy is higher, but the normalized reach is very small. As shown in Figure 3, the optimal MaxRatio policy needs more number of transmissions for attaining the same mean reach. For example, to obtain a mean reach of 0.25, the optimal policy uses about 60 transmissions, while MaxAct/MacReach needs only about 20 transmissions. A higher number of transmissions is expected because the ratio-based policy picks those slots in which substantial reach as well as relatively less irritation is expected. In Table 1, we list the top 5 time slots (GMT time) as per MaxReach policy for the root node, and in Table 2, we list the top 5 slots for MaxReach. It is interesting to note the contrast between these strategies. While MaxReach prefers weekdays, the ratio based policy picks GMT Sunday 9:00 hours as the top slot and GMT Sunday 8:00 hours as the third preferred slot. A closer look reveals that there are three users with time zones Beijing, Quito, and Beijing respectively, that have high activity probabilities (0.19, 0.19, 0.11 respectively) on Sunday at 9am GMT and their number of followers is 139, 1323, and 215 respectively. Since most of the other users are in USA and are inactive at this time, the ratio based policy picks this slot, which results in low irritation but high reach. Such a choice has two aspects.

- The MaxRatio policy chooses different slots than MaxReach, which is correlated with the typical practice of using high tweeting activity periods. Thus compared to competitors of the root node, this different strategy may make the tweets stand out.
- However, it should be noted that since we only consider the number of followers d_n for computing the reach, and not their activities during the relevant period, some of the slots proposed by the ratio based policy may need closer scrutiny. For example, if a user exhibits high activity in an uncommon time slot for her time zone, and if several of her followers are in the same time-zone, then the effective reach of this user is quite small, even though the number of followers may be quite high. This goal can be attained by modifying the weights d_n to capture such activity/location aspects, but in this paper, we do not pursue this further. (**Remark:** A similar comment applies to MaxReach as well, but since a typical slot chosen by MaxReach has lot of active users, it is less sensitive to the presence of nodes with high reach but poor effective reach due their uncommon behavior.)

Time Slot	Day	Time
154	Sun	9:00
80	Thu	7:00
153	Sun	8:00
7	Mon	6:00
104	Fri	7:00

Table 1: Top 5 slots for MaxRatio

No Response Under Irritation (NRUI)

Consider the case $\beta_{n,0} = \beta_n$ and $\beta_{n,s} = 0$ for $s > 0$, that is, if the state is 1 or more, then the user does not forward

Time Slot	Day	Time
63	Wed	14:00
64	Wed	15:00
87	Thu	14:00
112	Fri	15:00
39	Tue	14:00

Table 2: Top 5 slots for MaxReach

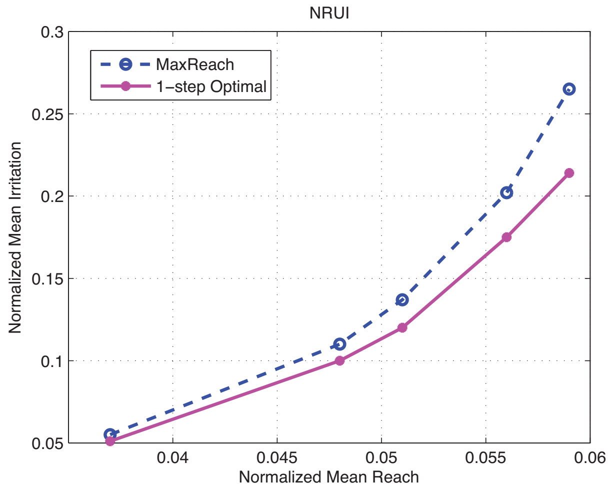

Figure 4: Comparison of the 1-step optimal policy under NRUI assumption with MaxReach policy.

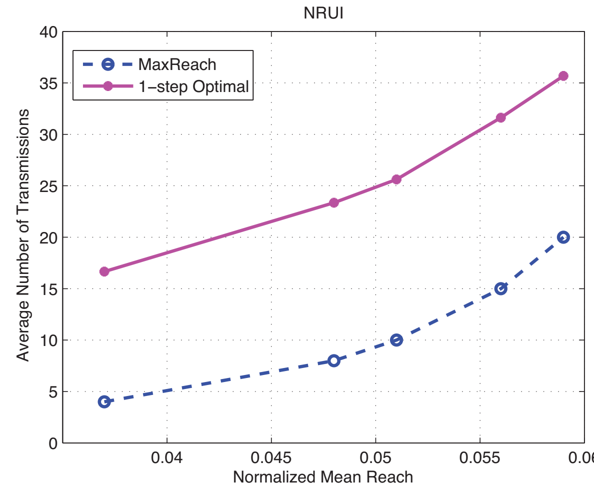

Figure 5: Comparison of the number of transmissions to attain a given reach level for the 1-step optimal policy under NRUI and MaxReach.

the tweet. In this case, from (4), we see that the optimal $M(T) = 1$ if and only if

$$\frac{\sum_{n:S_n(T)=0} d_n a_n(T)\beta_n}{\sum_{n=1}^{N} d_n a_n(T)} > c.$$

Due to the dependence on the set $\{n : S_n(T) = 0\}$ the explicit calculation of $J_T(\boldsymbol{s})$ is difficult. Moreover, even for a small number of users the problem becomes computationally intractable due to the large state space. Hence, we resort to a suboptimal algorithm:

$$M(t) = 1 \text{ iff } \frac{\sum_{n:S_n(t)=0} d_n a_n(t)\beta_n}{\sum_{n=1}^{N} d_n a_n(t)} > c.$$

This algorithm attempts to make the best decision under the assumption that the next step is the last step. Hence we refer to it as the 1-step optimal algorithm. We can compare this algorithm with MaxAct, MaxReach, and the optimal algorithm under SIRP. We find that the performance of MaxAct and the optimal algorithm under SIRP are slightly worse than MaxReach under the NRUI setting. Hence in Figures 4 and 5, we show a comparison of the 1-step optimal algorithm with only the MaxReach policy. We note that for both the algorithms, the irritation increases very rapidly as the reach increases. This is a consequence of the fact that once a node is irritated, no one can be reached through it again, and this forces us to work with very low normalized mean reach. The 1-step optimal policy, though not optimal, does give about 15-20% reduction in mean irritation at normalized mean reach levels above 0.05. In this regime, the average number of transmissions under the 1-step policy is roughly 1.5 to 2 times more than the MaxReach policy with the same mean reach. To implement the 1-step optimal policy, we need to monitor the followers and collect information about $A_n(t), F_n(t)$ so that the state can be tracked. In contrast, MaxReach can be designed offline.

5 Conclusion

The different strategies we have considered have different value depending on the perspective and we summarize our findings so far from different angles below.

1. **Implementation Complexity:** MaxReach, MaxAct, and MaxRatio (the optimal policy under SIRP) are offline while the 1-step optimal policy needs knowledge of the $A_n(t), F_n(t)$'s and hence has higher complexity.

2. **Performance Under SIRP:** For the dataset considered, MaxReach and MaxAct yield almost identical performance while the optimal MaxRatio policy can have up to 15-20% reduction in the mean irritation level in the moderate mean reach regime.

3. **Performance Under NRUI:** For the dataset considered, MaxAct and MaxRatio (which is the optimal policy under SIRP) are somewhat inferior to MaxReach, while the one-step optimal policy can yield improvement up to 15-20% at normalized mean reach level of around 0.05. All schemes have substantially less reach than the SIRP case.

4. **Robustness:** From the dataset available, it is not possible to determine if SIRP or NRUI (or if something in between) is true. But the MaxReach algorithm seems to be at most 15-20% from the best known scheme under either of the case. In this sense, it is *robust*.

5. **Non-observable Reward:** Our cost function gives significant importance to response by the followers in the form of sending out a related tweet. As the dataset reveals, only a tiny portion of all the users retweet and yield reward/cost. However, sending tweets to the other vast majority of users may also be assigned a reward/cost, even though observing such impact is not possible. If we agree that there is also a reward for simply reaching out to these users, then MaxReach/MaxAct gain more significance.

$$J_T(s) = \min_{m\in\{0,1\}} E\left[J_{T+1}(\boldsymbol{S}(T+1)) - \sum_{n=1}^{N} d_n F_n(T) A_n(T) M(T) \Big| \boldsymbol{S}(T) = \boldsymbol{s}, M(t) = m\right]$$
$$= \min_{m\in\{0,1\}} \left\{\lambda \boldsymbol{d}^t E[\boldsymbol{S}(T+1)|\boldsymbol{S}(T) = \boldsymbol{s}, M(t) = m] - m\sum_{n=1}^{N} d_n \beta_{n,s_n} a_n(T)\right\}.$$

Figure 6: The DP equation for the terminal step.

Main Conclusion: Given the above, we make following main conclusions for the root node under consideration.

- If we are not sure about the model for $\beta_{n,s}$, then MaxReach is attractive for the root node since at low complexity it yields decent performance under SIRP as well as NRUI.
- The loss of performance under NRUI is quite severe, and every effort should be made to compose novel tweets to ensure that we are closer to SIRP than NRUI. If we expect SIRP to be more representative of reality, then we should use the MaxRatio that is optimal under SIRP.
- Following the followers in order to obtain retweet feedback is not important under SIRP.

Future directions: We think our work is only a first look at the problem and lot more needs to be done. One important direction is to validate the different scheduling strategies on Twitter, and currently, we are working in this direction. Second, we have worked with only one root node. It is likely that different nodes have different characteristics and the conclusions for other root nodes may be different. Third, considering one more level of depth of the graph around the root node may help fine tune the performance metric. And last but not least, it is important to understand the dynamics of the networks if a substantial fraction of nodes start using such scheduling strategies.

6 Appendix

In this appendix, we present the details of the MDP formulation and the DP algorithm. (For an introduction to these concepts, please see (Bertsekas 1995).) The cost function can be expressed in the form

$$C(\lambda)$$
$$= E\left[\lambda\sum_{n=1}^{N} d_n S_n(T+1) - \sum_{t=1}^{T}\sum_{n=1}^{N} d_n F_n(t) A_n(t) M(t)\right]$$
$$=: E\left[\lambda \boldsymbol{d}^t \boldsymbol{S}(T+1) + \sum_{t=1}^{T} g(W_n(t), M(t))\right],$$

that is, we have additive cost of $E[g(W_n(t), M(t))]$ at time t and a terminal cost of $\lambda E[\boldsymbol{d}^t \boldsymbol{S}(T+1)]$. This shows that we have an MDP and the optimal control $\{\mu_t\}_{t=1}^{T}$ is given by the dynamic programming (DP) algorithm, which we analyze below. To find the value function at time T and the optimal control value $M(T)$, we have to solve the optimization problem specified in Figure 6. From (1) and assumptions A1)-A2), we get that,

$$\boldsymbol{d}^t E[\boldsymbol{S}(T+1)|\boldsymbol{S}(T) = \boldsymbol{s}, M(t) = m]$$
$$= \boldsymbol{d}^t \boldsymbol{s} + m\sum_{n=1}^{N} d_n a_n(T)(1-\beta_{n,s_n}).$$

Therefore

$$J_T(s)$$
$$= \lambda \boldsymbol{d}^t \boldsymbol{s} + \min_{m\in\{0,1\}} m\sum_{n=1}^{N} d_n a_n(T)\big(\lambda - (1+\lambda)\beta_{n,s_n}\big).$$

From this we get (4). The computation of the optimal control for $t < T$ needs more assumptions and is discussed in Section 4 in detail.

References

Berry, R., and Gallager, R. 2002. Communication over fading channels with delay constraints. *Information Theory, IEEE Transactions on* 48(5):1135 –1149.

Bertsekas, D. P. 1995. *Dynamic Programming and Optimal Control*. Athena Scientific.

Bomberger, E. E. 1966. A dynamic programming approach to a lot size scheduling problem. *Management Science* 12(11):pp. 778–784.

Boyd, D.; Golder, G.; and Lotan, G. 2010. Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter. *Proc. Intl. Conf. Systems Sciences* 1–10.

Durrett, R. 1996. *Probability: Theory and examples*. Belmont: Duxbury Press, second edition.

Sahni, S. K. 1976. Algorithms for scheduling independent tasks. *J. ACM* 23:116–127.

Socialoomph. 2010. http://www.socialoomph.com/.

Su, C.-J.; Tassiulas, L.; and Tsotras, V. J. 1999. Broadcast scheduling for information distribution. *Wirel. Netw.* 5:137–147.

Sysomos. 2010. Replies and retweets on twitter. White Paper.

Twitter. 2011. Twitter API. http://dev.twitter.com/start.