Time Maps: A Tool for Visualizing Many Discrete Events Across Multiple Timescales

Max C. Watson
L-3 Global Solutions
McLean, USA
max.watson@l-3com.com
June 28, 2015

***Abstract*—Visualizing many events over long time periods poses a unique set of challenges. We show how two-dimensional plots displaying the timings between events can reveal both outliers and hidden structure. Adopted from the field of chaotic systems, these "time maps" allow users to identify features that can take place on timescales ranging from milliseconds to months, all within a single image. The exploratory value of time maps is demonstrated using examples from Twitter and online bot behavior.**

***Keywords*-time series; information visualization; return map;**

I. Introduction

Strikes of lightning, mouse-button clicks, the firing of neurons, all occur at distinct moments in time. The structure of discrete event data is deceptively simple−just a series of timestamps. However, it is difficult to display many events over long timelines without obscuring details on short timescales.

This problem is exemplified with histograms over time. Fig. 1 shows the number of events associated with a certain IP address over seven months. The height of each bar is equal to the number of events that occurred in each time bin. While this visualization does display important information about overall event rates, details regarding the timing of events within each bin are completely lost. One can of course view smaller portions of the timeline with a finer bin size. But two choices must then be made: what bin size should be used, and what portion of the histogram should be viewed? For long timelines, this amounts to an intractable array of options.

Several visualizations manage to display more details by stretching out the timeline. For example, in [1], [2], [3] the timeline is stacked linearly, like lines of text. Alternatively, the timeline can be shown as a spiral [4], [5], [6]. For comprehensive reviews of time series visualizations, see [7], [8]. While the stacked and spiral approaches can deliver some insights, information on timescales much shorter than the entire duration of the data can still remain hidden from the viewer. Displaying many events over long timelines can also contribute to information overload for the viewer, which serves to conceal patterns, rather than reveal them.

Various mathematical tools have been developed for analyzing discrete event data, also known as point processes.

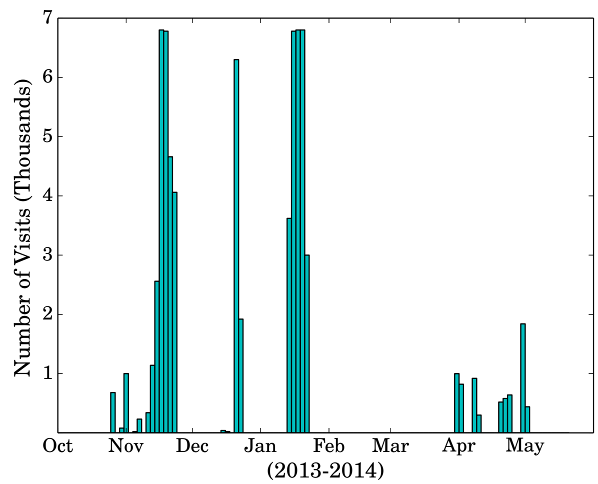

Figure 1. A histogram over time showing the activity of an internet bot operating from a specific IP address. The height of each bar is the total number of visits to various websites.

However, these tools yield curves which display average statistical properties [9], [10], [11], [12]. The raw data points are not shown.

In this paper, we present a visualization which displays the timings between events across multiple timescales within a single image. One of the first applications of this approach involved the analysis of water from a dripping faucet [13]. The specific type of plot has been referred to as a return map, or a return-time map. For simplicity, we will call the visualizations time maps. To our knowledge, time maps have only been applied within the field of chaotic systems. We will show how time maps can be used to study humans and computers. More specifically, we will demonstrate the exploratory power of time maps by visualizing the behavior of Twitter accounts and the activity of web-based entities, known as bots. In each case, the time map immediately reveals patterns on different timescales, with little or no zooming required.

II. Method

Time maps are simple to construct. As shown in Fig. 2, imagine each event as a dot along a time axis. Next, calculate the time difference between each event, denoted by t_1, t_2, t_3,

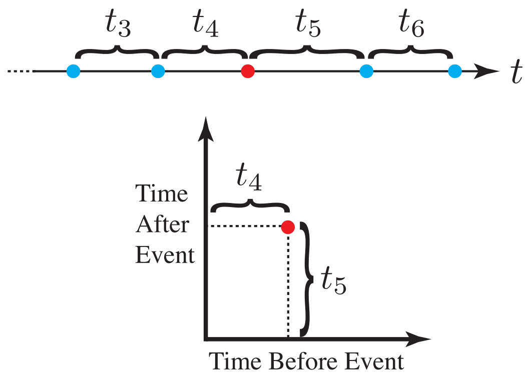

Figure 2. Top: events are shown as dots over time. The time between events is denoted by t_i. In a time map, the (x, y) coordinates of each event are given by (t_i, t_{i+1}). Bottom: the coordinates of the event colored in red.

and so on. The time map is a scatter plot, where the (x, y) coordinates are specified by each neighboring pair of time delays: (t_1, t_2) , (t_2, t_3) , (t_3, t_4) ... Every point drawn on a time map corresponds to an event within the dataset. In other words, an event's x-coordinate is the time delay between the event itself and the event preceding it. An event's y-coordinate is the time separation between the event itself and the event directly after it. The earliest (first) point in the dataset will not be displayed since the time delay before it is undefined. The same holds for the last point. Fig. 3 shows time maps for two simple sequences.

The time map paradigm provides key advantages. Since only the intervals between events are displayed, the timespan of the entire dataset is no longer an issue. When the separation time between events varies by orders of magnitude, the time map can be plotted on a logarithmic scale. Also, time maps are simple to create, and require no custom graphics.

Although time maps are extremely useful for analyzing discrete event data, they certainly do not replace existing visualization methods. After all, time maps remove information related to an event's absolute time of occurrence. Though the time of day can be incorporated into time maps (Fig. 9), the most powerful exploratory analysis will involve the combination of time maps with other visualization techniques.

Since we are accustomed to viewing time along a single axis, time maps are not intuitive to newcomers. Figure 4 is a heuristic diagram for interpreting time maps. It is divided into quadrants. Since points on a time map correspond to events, we will use "points" and "events" interchangeably. Roughly speaking, points in the lower-left and upper-right quadrants are evenly spaced in time between the events directly before and after them. The events in the lower-left occur in rapid succession, so the quadrant is labeled "fast and steady." The points in the upper-right occur at a slower rate. That region is therefore "slow and steady."

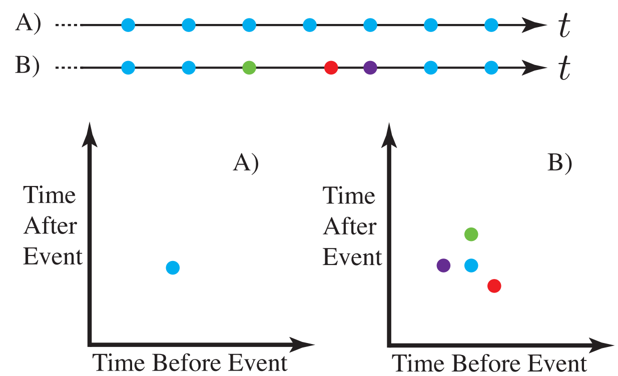

Figure 3. Top: two sequences of events. Bottom: their corresponding time maps. In sequence A, the events are evenly spaced. In the corresponding time map, all of the points are exactly on top of each other. Effectively one point in the time map represents the entire sequence. Series B is identical to A, except the timing of one event (colored red) has been changed. The resulting time map consists of effectively four points, making the change easy to notice. Unless a very small bin size is chosen, the histogram over time for each sequence would be identical, keeping the shift hidden.

Events in the upper-left quickly follow their preceding event, and a longer time elapses until the next event. The upper-left quadrant is therefore labeled as "slowing down." The lower-right quadrant is akin to "speeding up," since a long delay is followed by an accelerated pair of events. In a sense, viewing a time map for the first time is like reading a map of a new city. While unfamiliar at first, it eventually becomes second nature.

"Heated" versions of time maps are useful when dense clusters of points are present. The most straightforward method for generating a heat map involves the creation of a two-dimensional histogram based on the position of the

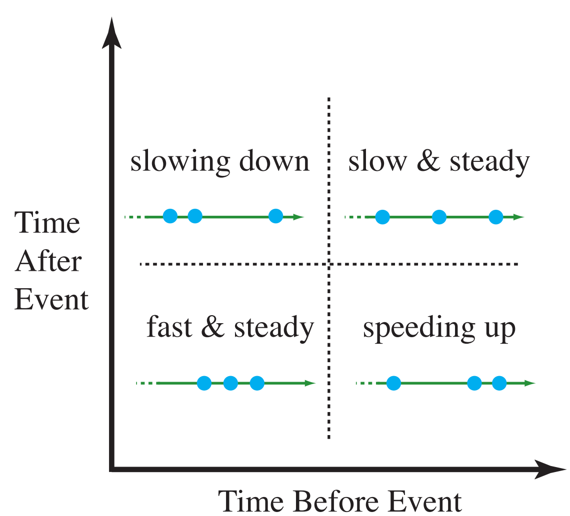

Figure 4. Time maps can be understood with the help of quadrants. Each quadrant is labeled, along with a diagram of events in time. The quadrants refer to the coordinates of the middle point within each diagram.

points in the time map. Computationally, the histogram $\mathbf{H}$ is represented by a two-dimensional matrix, where each entry corresponds to a grid-square of the time map. The value of each entry is equal to the number of events within that grid-square ($\mathbf{H}$ can easily be calculated in parallel if necessary). The most straightforward heat map is simply a color-coding of the entries in $\mathbf{H}$. A minor drawback of this approach is that the heat map appears pixelated. We avoid this effect by using a finely spaced histogram and applying a Gaussian filter to $\mathbf{H}$, which smoothens the image. The Gaussian filter can be applied using a single line of code in SciPy. For more details, see the supplementary Python file.

Assuming the distribution of delay times has statistically converged, time maps also offer a probabilistic interpretation. In general, the probability of a specific situation is the number of times the situation occurred, divided by the total number of events. With time maps, a "situation" is an event with time difference t_b before it, and time difference t_a after it. We can obtain probabilities by dividing each entry of $\mathbf{H}$ by the total number of events, yielding a probability associated with each grid-square of the time map. The probability $p\left(t_b, t_a\right)$ corresponding to a grid-square with coordinates $\left(t_b, t_a\right)$ is the probability of an event with time t_b before it and time t_a after it. In other words, $p\left(t_b, t_a\right)$ is the probability of a time difference t_b followed by a time difference t_a [1]. So time maps are both a visualization tool and a probabilistic quantity. If a region of a time map has a dense cluster of points, events in that cluster have a high likelihood of occurring. Similarly, a sparse region of points has a very low probability of occurring in the dataset. See appendix two for a brief technical discussion of conditional probability and statistical dependence.

[1] More formally, $p\left(t_b, t_a\right)$ is the probability that the time before an event is between t_b and $t_b + t_{bin}$, and the time after that event is between t_a and $t_a + t_{bin}$, where t_{bin} is the bin size of $\mathbf{H}$.

III. Examples

In this section, we demonstrate how time maps can provide immediate insights into the behavior of bots and the activity of Twitter accounts. Our intent is to establish the unique value of time maps in terms of exploratory visualization, not to perform detailed analyses.

A. *Internet Bot*

Internet bots are software applications that execute automated tasks online. Bots are unwittingly installed on millions of personal computers when users click on certain links. Using data from a company that provides monitoring services for various websites, we used time maps to visualize online visits from a specific IP address that exhibited high rates of activity. Over 70,000 visits were recorded from October 2013 to May 2014. Such behavior is a simple indication that the IP is controlled by a bot. Fig. 1 displays a histogram of the visits over that time period.

The time map in Fig. 5 is shown with logarithmic axes, allowing us to view the data across multiple timescales. There are four striking features: a symmetry about the diagonal, a dense square of fast and steady events in the lower left, two bars of points lying in the speeding up and slowing down regions, and the sparse points beyond them.

An overall symmetry about the diagonal is present in all of the data within this paper. Note however that the symmetry

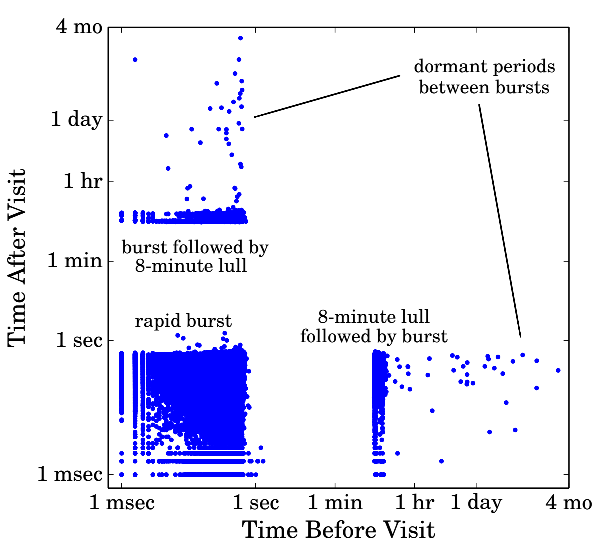

Figure 5. A time map displaying the activity of an internet bot (see also Fig. 1) visiting various websites. The axes are scaled logarithmically.

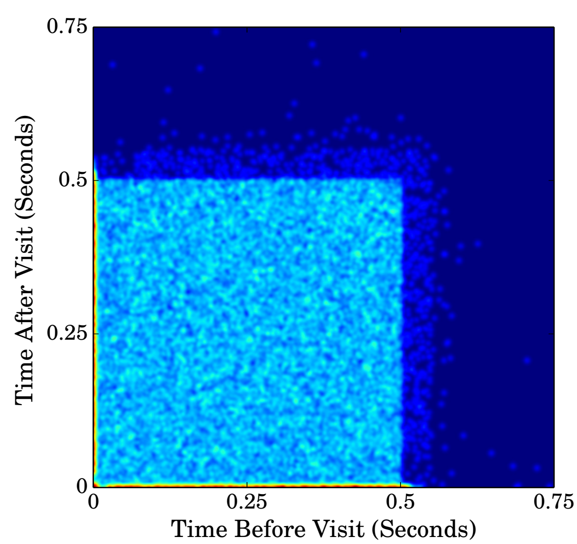

Figure 6. A heat map of the lower-left region of Fig. 5, shown on a linear scale. This block represents the bursts of activity shown in Fig. 7. For the most part, it appears that the time between each visit is a random number between zero and 0.5 seconds drawn from a uniform probability distribution.

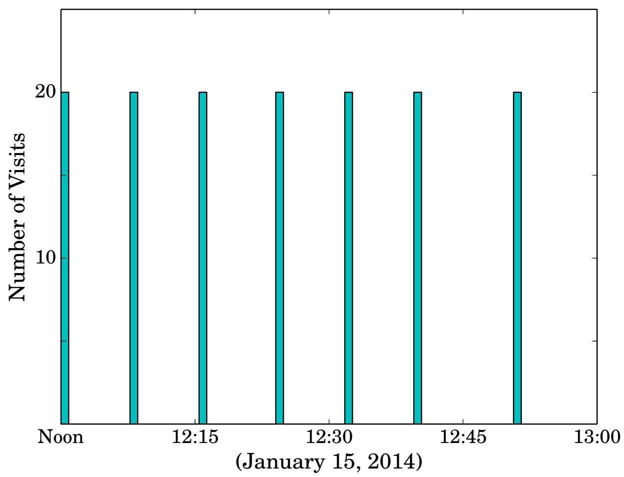

Figure 7. A zoomed-in histogram of the website visits shown in Fig. 1 on the morning of January 15, 2014. In this period, the internet bot visits 20 websites, followed by roughly 8-minute lulls.

is not exact. For discussion of this effect and its implications, see appendix one.

The regions of sparse points in Fig. 5 correspond to long periods preceding and following jumps in activity. Only the longest gaps can be seen in Fig. 1. The rest are concealed by binning effects. When plotted on a linear scale (Fig. 6), we see that the fast and steady block forms a square. (since the smallest unit of measurement is milliseconds, the logarithmic scaling creates line-like patterns for small delay times). While the locations of the points within the square do not have an ordered pattern, the density of points is very uniform. This indicates that the time between each visit is a random number drawn from a uniform probability distribution. The speeding up and slowing down bars in Fig. 5 show that there are 8-minute lulls between abrupt drops/jumps in activity. This is confirmed by re-scaling the time histogram using a much smaller bin size. By examining a small portion of the histogram at a zoomed-in level (Fig. 7), we see that the visits arrive in bursts of 20 hits each. The activity within each burst corresponds to the square shown in Fig. 6. Within each burst, the bot visits a list of 20 different websites. Though the bursting behavior is evident in the fine-grained histogram, the histogram over the entire dataset (Fig. 1) provided absolutely no evidence of the 8-minute timescale. It was the time map that immediately revealed this feature. Unlike the histograms, the time map required no zooming, since it can display all timescales in a single image.

B. *Tweets from @BarackObama*

The Twitter API (application programming interface) allows you to gather the 3,200 most recent tweets written by a specific user. We downloaded the timeline from @BarackObama, which is mostly managed by the president's staff. The tweets extend from October 2013 to April 2015. A heated time map and a time map are shown in Figs. 8 and 9. The time map is color-coded by time of day. Four main clusters are present. As in the previous example, there are speeding up and slowing down bars. These mark the beginning and ending of each work day (in terms of tweets).

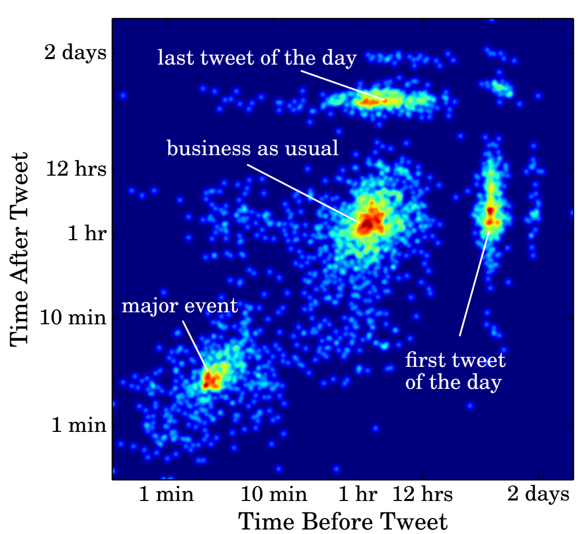

Figure 8. A heated time map for tweets written by @BarackObama. There are two distinct modes of behavior. During major events, a tweet is written every few minutes. On other days, the tweet rate is about once per hour. See also Fig. 9.

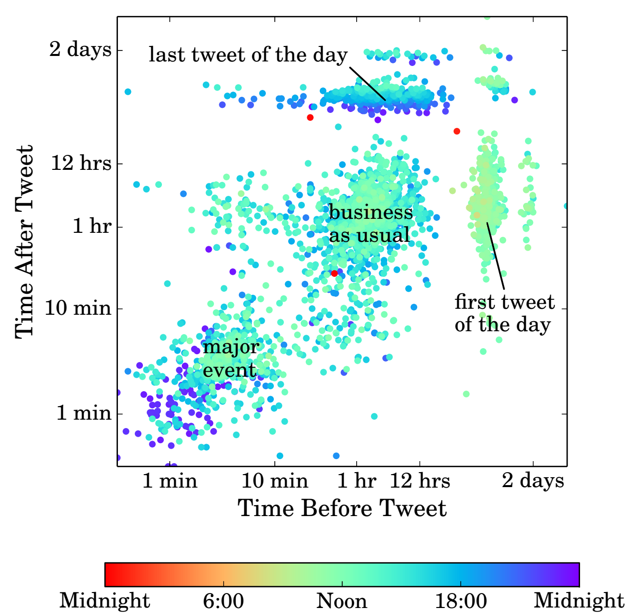

Figure 9. Tweets written by @BarackObama. The points are color-coded by time of day. Note the wide range in color for the last tweet of the day. The red dots correspond to a State of the Union address and the dawning of the Affordable Care Act. See also Fig. 8

The speeding up bar represents the first tweet of the day. It exhibits very little color variation, meaning these tweets occur very close to noon. The slowing down bar corresponds to the last tweet of the day. The greater range in color indicates that these tweets occur over a wider range of times. The two clusters along the diagonal represent tweets composed during the workday. Interestingly, the clusters are quite distinct. The tweets in the larger cluster are written at a rate of roughly one per hour, signifying "business as usual." The tweets from the fast and steady cluster correspond to major speeches and events. Many of the dark purple points in this cluster were tweeted on the night of the 2015 State of the Union address.

In addition to clusters, some tweets are also outliers. For example, in Fig. 9, three red points were written in the very early morning. Of these, two were tweeted after the 2014 State of the Union address. The third, written on the dawn 2014, celebrates the onset of the Affordable Care Act.

C. Twitterbot

The final, richest example is the Twitterbot @oliviataters. Twitterbots are programs that can tweet, follow users, and perform all of the same tasks available to humans on Twitter. @oliviataters not only writes unprovoked, "normal" tweets. It also replies to tweets written by other users. Using the Twitter API, we obtained a timeline from February to April 2015. A heated time map is shown in Fig. 10. Compact speeding up and slowing down clusters are present. They indicate that there are lulls lasting roughly two hours, followed by tweets separated by about 15 minutes. The red region in the center of the plot corresponds to activity within these active periods. The highly compact nature of this

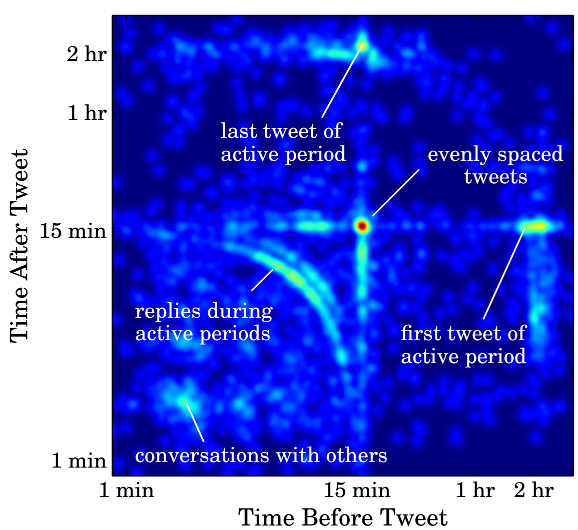

Figure 10. A heated time map for tweets written by @oliviataters, a Twitterbot. The clusters correspond to various aspects of the bot's behavior. See also Fig. 12.

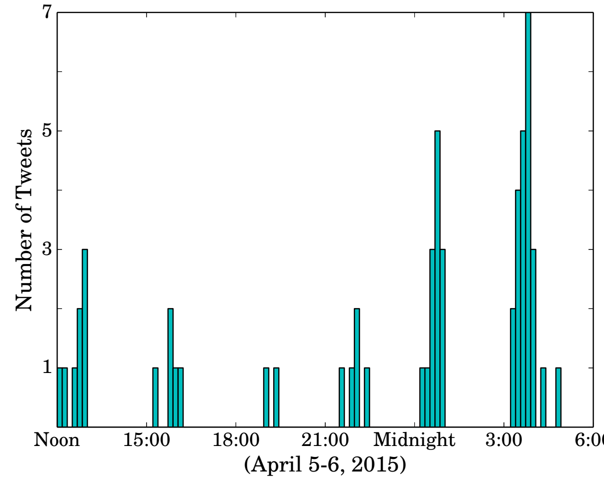

Figure 11. A histogram over time for tweets written by @oliviataters spanning April 5-6, 2015. Active periods of tweets are separated by two-hour lulls. This feature can also be seen in Fig. 10.

region, along with its color, implies that there are multiple events within the active periods, and that they are uniformly separated by 15-minute intervals. A zoomed-in histogram showing a sample 18-hour period (Fig. 11) is consistent with this picture. Again, the time map immediately revealed these features with no zooming required.

There is more to @oliviataters than lulls and active periods. In Fig. 12, we have slightly zoomed in and plotted

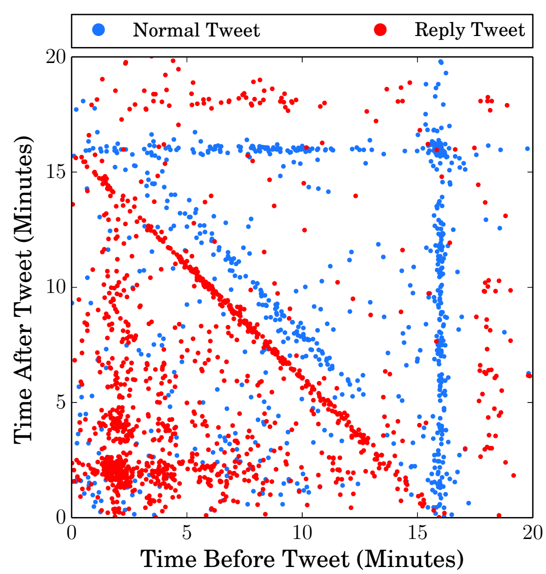

Figure 12. Tweets from @oliviataters, a Twitterbot. Compared to Fig. 10, we have slightly zoomed in on the lower-left quadrant and used linear axes. Red: tweets in reply to tweets written by other users. Blue: "normal" tweets, which are not replies. The cluster in the lower left corresponds to conversations with other users. The origin of the triangular shape is explained in the text and in Fig. 13.

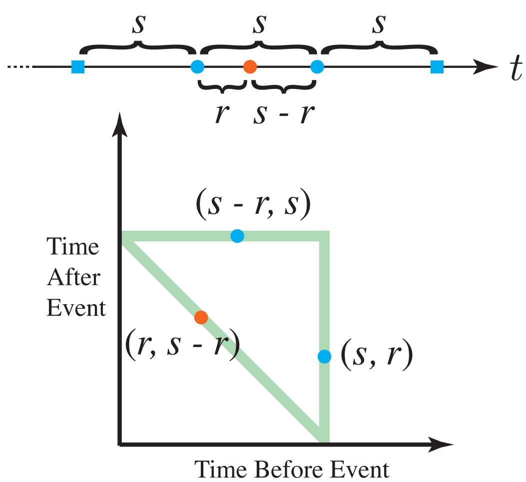

Figure 13. Top: four normal tweets (blue) separated by the time interval s, representing normal tweets written by @oliviataters. A reply tweet (red) occurs r seconds after the second normal tweet. It therefore occurs $s - r$ seconds before the third normal tweet. Bottom: the resulting time map. The green lines mark the locations of the points for different values of r. This mechanism accounts for the triangular shape found in Figs. 10 and 12. The tweets marked by squares in the top of the figure are not shown.

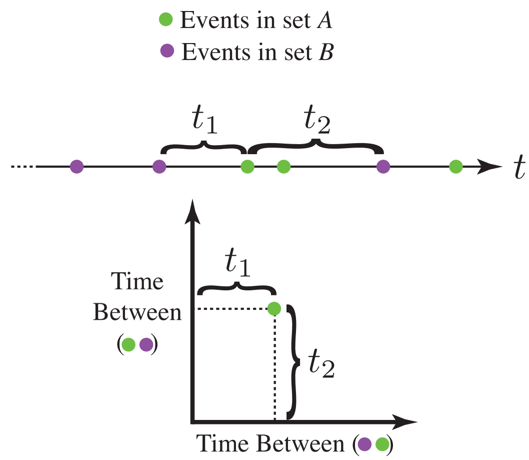

Figure 14. Time maps may also be built to visualize correlations between datasets. Top: a timeline with two overlapping datasets: A and B. Bottom: a time map showing how an event from set A would be plotted. A similar time map could be constructed for events in set B. Lines or clusters in the resulting plots would point to relationships between the datasets.

the time map on a linear scale. Tweets in reply to other users are shown in red. Normal tweets are shown in blue. By color-coding the tweets, we see that the cluster of points in the upper right, corresponding to tweets separated by 15-minute intervals, consists of mostly normal tweets. The cluster on the lower left represents replies within conversations between @oliviataters and other users. In these conversations, the Twitterbot exchanges multiple tweets with the same users.

The triangular shape in Fig. 12 tells an interesting story. For simplicity, we will focus on the diagonal line of reply tweets and the vertical/horizontal lines of normal tweets. We have already established that there are periods of activity where normal tweets are written every 15 minutes. The triangle represents normal tweets that are evenly spaced with each other, along with replies that occur in-between them. To understand this, see Fig. 13. A sequence of evenly spaced dots is shown, representing the normal tweets written during an active period. While more complicated scenarios can also occur, this situation contains the required ingredients that give rise to the triangular shape. The time separation between each normal tweet is denoted by s. A reply tweet occurs r seconds after one of the normal tweets. The resulting time map is shown in Fig. 13. Since r can have any value between zero and s, we also show the lines swept out by all possible values of r. The lines form a triangle! This mechanism not only accounts for the shape. It also explains why replies lie along the diagonal. To conclude, the triangle arises from a special combination of normal tweets and replies. The normal tweets are equally spaced with each other, while the replies are interspersed between them. Though the exact mechanism was not trivial, it manifested itself in terms of a simple shape within the time map that was easy to spot. This again demonstrates how time maps are a powerful tool for exploratory data visualization.

IV. Conclusion

In each of the examples, time maps exposed critical features. With the internet bot, we discovered bursts of activity every 8 minutes. This phenomenon was completely hidden in a histogram displaying the 7-month timespan of the data. Tweets written by @BarackObama exhibited two distinct modes of behavior: “business as usual” and “major event.” Color-coding the tweets by time of day also allowed us to spot daily trends. For the Twitterbot @oliviataters, a fast and steady cluster of replies revealed the presence of conversations with other users. A triangle pointed to replies inserted between normal tweets. The time maps also revealed outliers and smaller clusters that defied our general categories. For example, outliers in the tweets by @BarackObama coincided with the State of the Union speech and the onset of the Affordable Care Act. Though the above features took place on timescales ranging from milliseconds to months, they were revealed with little or no zooming at all.

Time maps can be modified to visualize correlations between datasets. For example, say there are two datasets with overlapping timestamps (Fig. 14). For each event in set A, its x-coordinate would be the time difference between the event itself and the most recent event from set B. The y-coordinate would be the time between the event itself and the next event in set B. Points from set B would be displayed in the same way. Lines or clusters in the resulting plot would be indicative of links between events in the different datasets.

Time maps can also be built to probe more complex effects. While the xy coordinates of our time maps were (t_i, t_{i+1}), one could build coordinates based on (t_i, t_{i+10}) to examine long-term correlations. Three-dimensional time maps may also reveal deeper relationships, where the xyz coordinates are given by (t_i, t_{i+1}, t_{i+2})[13].

Additional features can be added to time maps to provide even more information. If events have a geospatial component, the points can be colored based on their distance to the prior event. Mouse-interactivity would allow users to select specific points for further analysis. When a point is selected, additional details about the event could appear, along with a different visualization showing events that took place around the same time. For example, in our analysis of the internet bot, a zoomed-in time histogram was used to learn more about the speeding up and slowing down events in the time map. The xy coordinates of a selected event would dictate the zoom-in level for the histogram.

Picasso captured the entirety of an object by displaying it from multiple angles within a single painting. Similarly, time maps distill a dataset across multiple timescales within a single image. While many visualizations require a large amount of custom programming, time maps can be created with just a few lines of code. Given their easy implementation and ability to reveal hidden structure, time maps should become an invaluable tool for studying humans, bots, and everything in-between.

Acknowledgment

The author thanks Dina Eliezer, Joe Costantini, and Steve Moccio for insightful discussions.

Appendix 1: Symmetry in Time Maps

The high degree of symmetry about the diagonal (the line $x = y$) appears in all of the time maps we have examined. Though the symmetry is not perfect, it arises from the fact that the data has largely the same properties if the events transpired in reverse order. For example, imagine we heard a recording containing a beep each second, but every 10th beep was omitted. Playing the recording backwards would sound exactly the same. The time map for the forwards and backwards recordings would be identical, and both time maps would be symmetric about the diagonal. Likewise, the data in this paper would “sound” the same if the events were replayed in reverse order.

Discrete event data does exhibit this property in general. Imagine a sequence of beeps where the time between each beep decreased over the course of the recording. Playing the recording backwards would sound different, since the beeps would progressively get farther apart. The time maps for these two scenarios would not be identical. The time maps also would not be symmetric about their diagonal.

Appendix 2: Conditional Probability and Dependence

This appendix discusses conditional probability and statistical dependence applied to time maps, and requires familiarity with the basics of probability. Time maps are not only visualizations. They also represent the joint probability distribution $p\left(t_b, t_a\right)$ of a time difference t_b followed by a time difference t_a. We also introduce two other quantities: $p\left(t\right)$, the probability of a time difference t, and $p\left(t_a|t_b\right)$, the probability that a time difference is t_a given that the time difference before it is t_b. If two successive time differences are statistically independent,

$$p\left(t_b, t_a\right) = p\left(t_b\right) p\left(t_a\right) \tag{1}$$

for all values of t_b and t_a. With time maps, it is easier to check for independence using an equivalent form of (1):

$$p\left(t_a|t_b\right) = p\left(t_a\right) \tag{2}$$

which simply states that $p\left(t_a|t_b\right)$ does not depend on t_b.

In all of the examples in this paper, successive time differences are statistically dependent. This is because each example contains periods of rapid activity separated by individual lulls. To understand this, look at Fig. 5, showing the time map for the internet bot. In this case, $p\left(3\,\mathrm{months}|3\,\mathrm{months}\right) = 0$, but $p\left(3\,\mathrm{months}\right) \neq 0$. The same overall property holds for @BarackObama and @oliviataters. Given a large value of t_b, the probability of a large value of t_a is greatly reduced.

While successive time differences can be statistically dependent for a dataset as a whole, the times between events within active periods may still be independent. For example, Fig. 6 shows the portion of the time map associated with the internet bot’s rapid bursts. For those events, it is possible that each time difference is drawn from a uniform probability distribution. In other words, those time differences may be independent and identically distributed random variables.

References

[1] Milton H Shimabukuro, Edilson F Flores, Maria Cristina F de Oliveira, and Haim Levkowitz. Coordinated views to assist exploration of spatio-temporal data: A case study. In *Coordinated and Multiple Views in Exploratory Visualization, 2004. Proceedings. Second International Conference on*, pages 107–117. IEEE, 2004.

[2] Krist Wongsuphasawat and Ben Shneiderman. Finding comparable temporal categorical records: A similarity measure with an interactive visualization. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 27–34. IEEE, 2009.

[3] Fernanda B Viégas, Danah Boyd, David H Nguyen, Jeffrey Potter, and Judith Donath. Digital artifacts for remembering and storytelling: Posthistory and social network fragments. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, pages 10–pp. IEEE, 2004.

[4] Marc Weber, Marc Alexa, and Wolfgang Müller. Visualizing time-series on spirals. In *Information Visualization, IEEE Symposium on*, pages 7–7. IEEE Computer Society, 2001.

[5] Jinfeng Zhao, Pip Forer, and Andrew S Harvey. Activities, ringmaps and geovisualization of large human movement fields. *Information visualization*, 7(3-4):198–209, 2008.

[6] Takafumi Saito, Hiroko Nakamura Miyamura, Mitsuyoshi Yamamoto, Hiroki Saito, Yuka Hoshiya, and Takumi Kaseda. Two-tone pseudo coloring: Compact visualization for one-dimensional data. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 173–180. IEEE, 2005.

[7] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. *Visualization of time-oriented data*. Springer Science & Business Media, 2011.

[8] Ebad Banissi, Camilla Forsell, Jimmy Johansson, and Francis T Marchese. *Information Visualisation: Techniques, Usability and Evaluation*. Cambridge Scholars Publishing, 2014.

[9] David Roxbee Cox and Peter AW Lewis. The statistical analysis of series of events. *Monographs on Applied Probability and Statistics, London: Chapman and Hall, 1966*, 1, 1966.

[10] Abdel H El-Shaarawi and Walter W Piegorsch. *Encyclopedia of environmetrics*, volume 3. John Wiley & Sons, 2001.

[11] David R Brillinger. Fourier analysis of stationary processes. *Proceedings of the IEEE*, 62(12):1628–1643, 1974.

[12] MS Bartlett. The spectral analysis of point processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 264–296, 1963.

[13] Robert Shaw. *The dripping faucet as a model chaotic system*. Aerial Press, 1984.