When is it Biased? Assessing the Representativeness of Twitter's Streaming API

Fred Morstatter
Arizona State University
699 S. Mill Ave
Tempe, AZ, 85281
fred.morstatter@asu.edu

Jürgen Pfeffer
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA, 15213
jpfeffer@cs.cmu.edu

Huan Liu
Arizona State University
699 S. Mill Ave
Tempe, AZ, 85281
huan.liu@asu.edu

ABSTRACT

Twitter shares a free 1% sample of its tweets through the "Streaming API". Recently, research has pointed to evidence of bias in this source. The methodologies proposed in previous work rely on the restrictive and expensive Firehose to find the bias in the Streaming API data. We tackle the problem of finding sample bias without costly and restrictive Firehose data. We propose a solution that focuses on using an open data source to find bias in the Streaming API.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

Keywords

Data Sampling, Sampling Bias, Twitter Analysis, Big Data

1. INTRODUCTION

Twitter's policy for data sharing is very open, providing a free "Streaming API"[1] that returns tweets matching a query. One drawback of the Streaming API is that it only returns at most 1% of the tweets on Twitter at a given moment. Once the volume of the query surpasses 1% of all of the tweets on Twitter, the response is sampled. The way in which Twitter samples the data is unpublished. Recent research [3] has shown that there is strong evidence of bias in the data returned from the Streaming API, namely in the top hashtags found in the text.

In this work we seek to find bias in the Streaming API automatically. We define "bias" as sample bias. We say that a hashtag is "biased" if the relative trend is statistically significantly over-represented or underrepresented in contrast to its true trend on Twitter. In particular, we are looking for particular time periods of bias in the Streaming API. Based on Twitter's documentation, the sample size is determined by the volume of a query at a point in time. We identify time periods where the Streaming API is biased.

[1] `https://dev.twitter.com/docs/streaming-apis`

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2576952.

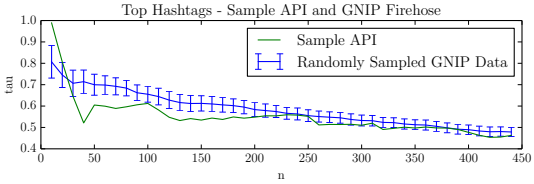


Figure 1: Rank correlation of Sample API and Gnip Firehose. Relationship between n - number of top hashtags, and τ_β - the correlation coefficient for different levels of coverage.

2. DISCOVERING BIAS IN THE STREAMING API WITHOUT THE FIREHOSE

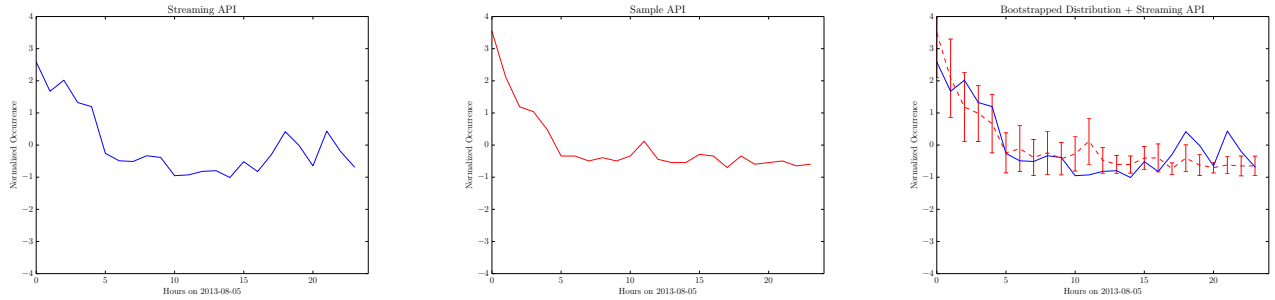
With work showing evidence that the Streaming API is biased, researchers must be able to tell whether their Streaming API sample is biased. Vetting their dataset with the Firehose is prohibitive due to the cost and restrictive nature of the feed. We propose a methodology that can give an indication of bias for a particular hashtag using publicly-available data sources.

We investigate whether another open data source, Twitter's Sample API, can be used to find bias in the Streaming API. We show that using the Sample API, one can detect bias in the Streaming API without the need of the prohibitive Firehose. We focus on alternative methods to help the user understand *when* their data diverges from the true activity on Twitter.

2.1 Vetting the Randomness of the Sample API

Given the evidence of bias that was observed in the Streaming API, we must proceed with caution before using the Sample API as a surrogate gold standard. We collect all of the tweets available through the Sample API on 2013-08-30 from 17:00 - 21:00 UTC, and post-filter them by the keyword "syria". Simultaneously, we collect all of the tweets matching the keyword "syria" from the Gnip Twitter, another outlet for Firehose data. Gnip also provides 100% of the publicly-available Tweets on Twitter.

We compare the ranked list of top hashtags in both sets. We first plot Kendall's τ_β score of the Sample API against the Firehose. Kendall's τ_β calculates the number of concordant and discordant pairs between two ranked lists. This score gives us a sense of whether the frequency of hashtags



(a) Streaming API Results. Trendline for "#believemovie" over one day.

(b) Sample API Results. Trendline for "#believemovie" over one day.

(c) Bootstrapped μ and ± 3 standard deviations overlaid with Streaming API.

Figure 2: Figures outlining different steps of the process for finding bias in the Streaming API.

coming through the Sample API is similar to the Firehose. We plot the average and standard deviation of 100 perfectly randomly sampled datasets of the same size as the Sample API against the Firehose in Figure 1.

While the Sample API closely resembles the random samples, we see τ_β values occasionally fall outside of one standard deviation. We perform a statistical test to ensure that the Sample API data and Firehose data are not independent. We calculate two-sided significance level from the Kendall τ statistic, which tests the following hypothesis: H_0 – The top k hashtags in the Firehose data and the top k hashtags in the Sample API data are independent, $\tau_\beta = 0$. We vary k from 10 to 450 in increments of 10. In all cases we reject H_0 with a 95% confidence level.

2.2 Finding Bias in the Trend of a Hashtag

With only one unbiased view from the Sample API, it is difficult to understand what we should expect from our Streaming API data. When the results match there is clearly no problem. When there is a difference, how do we know if the relative error is significant or if it is just a small deviation from a random sample? To better understand the Sample API's response, we bootstrap [1] the Sample API to obtain a confidence interval for the relative activity for the hashtag.

We begin by normalizing both sources by computing their standard score, which is calculated as: $\frac{t_i - \mu_T}{\sigma_T}$, where μ_T and σ_T are the mean and standard deviation of all of the time periods in the time series, respectively, and t_i is an individual time series point. This ensures that the distribution of points from both time series is $\mathcal{N}(0, 1)$. We create 100 bootstrapped samples for each hashtag from the Sample API. We then extract the time series data from each sample and normalize them. This gives us a distribution of readings for each time period in the dataset. Next, we compare this distribution to the normalized time series from the Streaming API to detect the bias. We take the sample mean and sample standard deviation of this distribution at each point t_i as μ_i^b and σ_i^b. We say that any Streaming API value at time t_i that is outside of $\pm 3\sigma_{t_i}$ is biased.

We enumerate the process for "#believemovie" on August 5th, 2013 in Figure 2. The process begins with the time series data for this hashtag from both the Streaming and Sample APIs, shown in Figures 2(a) and 2(b), respectively. To obtain a confidence interval on the difference of the two sources, we create 100 bootstrapped samples. Finally, taking the mean and 3 standard deviations of the bootstrapped samples, we obtain the confidence intervals seen in Figure 2(c). A spike that occurs between hours 10 and 11 is underrepresented in the Streaming data. Also, the spikes that appear after hour 16 are both over-represented in the Streaming API. Due to the nature of our bootstrapping method, these observations are all statistically significant at the 99.7% confidence interval.

3. DISCUSSION AND FUTURE WORK

In this work we ask how to find bias in the Streaming API without the need for costly Firehose data. We test the representativity of another freely-available Twitter data source, the Sample API. We find that overall the tweets that come through the Sample API are a representative sample of the true activity on Twitter. Finally, we propose a solution to harness this data source to find time periods where the Streaming API is likely biased.

Future work is to correct the bias directly. A simple solution is to purchase the data from resellers during times when the Streaming API is biased. Another is to cross-reference users from Twitter with other social media outlets to gain additional insight. Another direction is to improve the performance of our method by exploring alternative bootstrapping methods that may be more suitable for sparse data, such as [2]. We seek to find bias in sparse data scenarios, and adapt these methods to the speed and ephemerality of Twitter data.

4. ACKNOWLEDGMENTS

This work is sponsored, in part, by Office of Naval Research grants N000141010091 and N000141110527. The full version of this paper can be found here: `http://arxiv.org/abs/1401.7909`.

5. REFERENCES

[1] B. Efron. *The Jackknife, The Bootstrap and Other Resampling Plans*. Society for Industrial and Applied Mathematics, 1987.

[2] P. D. Kirk and M. P. Stumpf. Gaussian process regression bootstrapping: exploring the effects of uncertainty in time course data. *Bioinformatics*, 25(10):1300–1306, 2009.

[3] F. Morstatter, J. Pfeffer, H. Liu, and K. Carley. Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose. In *ICWSM*, 2013.