Trending topics using the Twitter-API

Marco Pock-Steen Fraile
Christian Palmhøj Nielsen
IT University of Copenhagen

December 11, 2012

Abstract

We have designed an algorithm for identifying trending topics in twitter..

1 Introduction

We wish to make use of the Twitter streaming API to implement our own version of trending topics using the Misra-Gries algorithm, and for the trending topics we would track how many unique tweets have been made per topic, using one of the algorithms for counting distinct elements. We could also count number of tweets for trending topics to compare actual tweets with unique tweets, but this is not algorithmically a challenge.

Challenges would include finding a suitable k for the Misra-Gries that is a good trade-off between memory space and making sure we take into account the tweets inbetween the actual trending topics that might "flood" them out of the array.

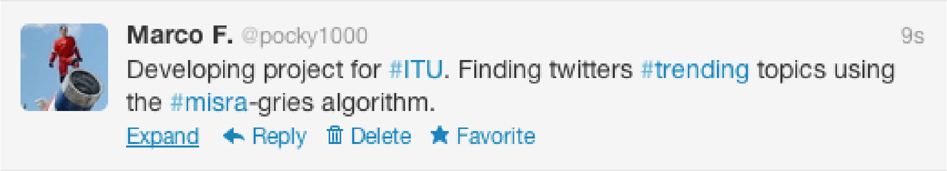
For the counting algorithm we similarly need to find a good hashing algorithm for tweets, and based on this find a good number for kth-minimum distances that gives us a realistic count for unique tweets.

Outline The remainder of this article is organized as follows. Section 3 gives account of previous work. Our new and exciting results are described in Section 4. Finally, Section 5 gives the conclusions.

2 Twitter topics

Twitter is a real-time information network that connects you to the latest stories, ideas, opinions and news about what you find interesting.

At the heart of Twitter are small bursts of information called Tweets. Each Tweet is 140 characters long. An example of a tweet can be seen below.



Marco F. @pocky1000 9s

Developing project for #ITU. Finding twitters #trending topics using the #misra-gries algorithm.

Users of twitter can see the tweets of people they follow, and have the options to either reply to a tweet, or retweet it. On retweet the user basically shares the same message to their own followers, giving credit to the creator of the message. The topic of a tweet is identified by the presence of a hashtag # in front of a word. In the example above the topics are #ITU,#trending and #Misra. Tweets can be tagged by writing a hashtag "#" followed by a topic or keyword. Twitter users this as a way to categorize messages. A trending topic is defined by a topic that has a high frequency, in other words many people are "tweeting" about the same topic at some point in time. A trending topic can either be many unique tweets about the same topic, e.g. if people are tweeting to promote some beneficial cause, or it can be the same message retweeted by many users. Twitter currently have over 500 million users, so the amount of tweets produced at any moment is equally big, some specifics. We use the public Twitter API to create a stream of which corresponds to a sample of all tweets.

3 Streaming algorithms

Twitter and other social networks are structured to accomodate personal communication across large networks of friends, and a such produce enormous amounts of data. The open availability of this data through developer

APIs makes it's an interesting source for useful real-time information extraction using streaming algorithms. [3] This section introduces the concepts of algorithms that can compute some function of a massively long input stream σ such as all public available tweets. In our model this is formalized as a sequence $\sigma = \langle a_1, a_2, ..., a_m \rangle$, where the elements of the sequence (called *tweets*) are drawn from the universe $[n] =: \{1, 2, ..., n\}$. Note the two size parameters: the stream length, m, and the universe size, n. Our goal will be to process the input stream using a small amount of space s, i.e., to use s bits of random-access working memory. Since m and n are to be thought of "huge" we want to make s much smaller than these. Ideally we want to achieve $s = \mathrm{O}(\log m + \log n)$, because this is the amount of space needed to store a constant number of elements from the stream and a constant number of counters that can count up to the length of the stream. [2]

3.1 Finding frequent items

We want to find the the frequency of certain terms in each *tweet* in our input stream $\sigma = \langle a_1, a_2, ..., a_m \rangle$, and define a frequency vector $f = (f_1, ..., f_n)$, where $f_1 + ... + f_n = m$.

The Misra-Gries Algorithm solves the problem of estimating the frequencies f_j [2]

A similar scoring method is suggested where a topic is given a score based on the frequency over a period of time. That means that if a topic generally has a high frequency over a long period of time, it will not necessarily be considered trending if a topic with a lower frequency gained it over a shorter period of time. The standard score is defined as $z = \frac{x-\mu}{\sigma}$

3.2 Finding distinct values

K-minimum values (KMV) is is a probalistic distinct value counting algorithm, that is intuitive and easy to implement [1]. Suppose we have a good hash function that return evenly distributed values in the hash space $[0-1]$, then you could estimate the number of distinct values you have seen by knowing the average spacing between values in the hash space. The main challenge is to find a good hash function, and to select the number of minimum k values on which to approximate the average spacing. If the hash values were indeed evenly distributed, we could keep just keep track of the minimum value, a get a good estimate of distinct values. However taking only one value opens up to a lot of variance and would rely heavily upon the"goodness" of the hash function. In order to improve this Bar-Yossef[4] suggests keeping the k-smallest values, to give a more realistic estimate.

Other examples of the usages of data stream algorithms are described in this chapter.

3.3 Misra-Gries Algorithm

The algorithm first initializes a dictionary with k number of values. The keys in the dictionary are elements seen in the stream, and the value are counters associated with the elements. Then there is a process function that is executed each time we see a new element. If a new element is already in the dictionary, it's value will be increased by 1, otherwise if the number of elements in A is less than k, the element will be inserted an its value set to 1. If the length of A is equal to k, all values are decreased by 1, and removed if the value is equal to 0. [2].Finally we return the key value pairs with the highest frequencies.

Algorithm 1 Misra-Gries Algorithm

We use Misra-Gries to find frequent *topics* in our data stream of tweets using a one-pass algorithm.

```
$A \leftarrow InitializeArray$
function PROCESS($j$)
    if $j \in keys(A)$ then
        $A[j] \leftarrow A[j] + 1$
    else if $|keys(A)| < k - 1$ then
        $A[j] \leftarrow 1$
    else
        for $l \in keys(A)$ do
            $A[l] \leftarrow A[l] - 1$
            if $A[l] = 0$ then remove $l$ from A
            end if
        end for
    end if
end function
```

3.4 Analysis of algorithm

3.5 Rising trends

Varying k to change the amount of time the trending topics cover.

3.6 Data structure

When a new tweet is registered the algorithm has to determine if the contained topic has already been seen. We need a data structure that provides fast lookup, such as a dictionary with key-value pair as topic-frequency. We also need to find a data structure that allows for decrementing all values by 1 if the topic is not already created. Lastly we will have to find a way to randomly remove a topic if the data structure exceeds k at the creation of a new topic.

3.7 Data Set

4 Results

In this section we describe the results.

5 Conclusions

We worked hard, and achieved very little.

References

[1] Sketch of the day: K-minimum values.

[2] A. Chakrabarti. Cs85: Data stream algorithms. 2009.

[3] B. D. Zachary Miller and W. Hu. Gender prediction on twitter using stream algorithms with n-gram character features. 2012.

[4] R. K. D. S. Ziv Bar-Yossef, T. S. Jayram and L. Trevisan. Counting distinct elements in a data stream. 2002.