

EVENT DETECTION IN SOCIAL STREAMS

GROUP 9

BOGA SRINIVAS

ASHWANTH KUMAR GUNDETI

CHINTHA SAI SREENIVAS

NACHIKET CHOUHAN

DHEERAJ RACHA

NOVEMBER 10, 2019

- Motivation
- Problem Statement
- Proposed Methodology
- Experiments and Results

MOTIVATION

Social streaming sites contain a multitude of incoming news. Users on these sites have specific topic interests.

Hence, the feed of the users should be personalized to contain news of their interests. This can be accomplished by breaking the incoming stream into disjoint events.

PROBLEM STATEMENT

PROBLEM STATEMENT

- The problem of clustering and event detection in social streams in which each text message is associated with at least a pair of participants in the social network.
- Given an incoming stream S , the task of event detection is to assign this stream to one of ' n ' disjoint clusters, each corresponding to an event.

Challenges in event detection in social streams:

- The ability to use both the content and the structure (graphical) of the interactions for event detection.
- The ability to use temporal information in the event detection process.
- The ability to process the data in a single pass
- The efficient representation of data in memory through hashing, sampling, or latent factors.

PRELIMINARIES

How to represent data?

The structure of social stream can be denoted by the graph $G = (N, A)$. The node set being N and the edge set A .

Social Stream

A social stream is a continuous and temporal sequence of objects $S_1 \dots S_r \dots$, such that each object S_i corresponds to a content-based interaction between social entities, and contains explicit content information and linkage information between entities.

The object S_i is represented by the tuple (q_i, R_i, T_i) .

- The object S_i contains a text document T_i which corresponds to the content of the interaction of an entity in the social network with one or more other entities.
- The object S_i contains the origination node $q_i \in N$ which is the sender of the message T_i to other nodes.
- The object S_i contains a set of one or more receiver nodes $R_i \subseteq N$, which correspond to all recipients of the message T_i from node q_i . Thus, the message T_i is sent from the origination node q_i to each node $r \in R_i$. It is assumed that each edge (q_i, r) belongs to the set A .

Social Stream Clustering

A social stream $S_1 \dots S_r \dots$ is continuously partitioned into k current clusters $C_1 \dots C_k$, such that:

- Each object S_i belongs to at most one of the current clusters C_r .
- The objects are assigned to the different clusters with the use of a similarity function which captures both the content of the interchanged messages, and the dynamic social network structure implied by the different messages.

Fractional Cluster Presence

The fractional cluster presence for cluster C_i in the time period (t_1, t_2) is the fraction of records from the social stream arriving during time period (t_1, t_2) , which belong to the cluster C_i . This fractional presence is denoted by $F(t_1, t_2, C_i)$.

It is possible that existing cluster may relate closely to sudden burst of objects on a particular topic. This sudden burst is characterized by change in fractional presence of data points in clusters.

Such events are defined as Evolution events.

What all events can occur?

Evolution Event

An evolution event over horizon H at current time t_c is said to have occurred at threshold α for cluster C_i , if the ratio of the relative presence of points in cluster C_i over the horizon $(t_c - H, t_c)$ to that before time $t_c - H$ is greater than the threshold α .

Mathematically:

$$\frac{F(t_c - H, t_c, C_i)}{F(t(C_i), t_c - H, C_i)} \geq \alpha$$

It is also assumed that

$$t_c - 2.H \geq t(C_i)$$

This assumption is made to ensure that evolution happens in stable way.

The above ratio is evolution ratio *(which can be used to analyse temporal behaviour of an event)*

Novel Event

The arrival of data point S_i is said to be a novel event if it is placed as a single point within a newly created cluster C_i .

These kind of events occur when the incoming stream doesn't match with any of the current clusters.

Then, we'll replace the most stale cluster with a singleton cluster containing only the object S_i and corresponding cluster summary statistics.

Cluster Summary $\psi_i(C_i)$

- Node summary, set of nodes $V_i = \{j_{i1}, j_{i2} \dots j_{is_i}\}$ along with the frequencies $n_i = v_{i1}, v_{i2} \dots v_{is_i}$. This node set is assumed to contain s_i nodes.
- Content summary, set of word identifiers $W_i = \{l_{i1}, l_{i2} \dots l_{iu_i}\}$ and corresponding word frequencies $\Phi_i = \phi_{i1}, \phi_{i2} \dots \phi_{is_i}$. W_i is assumed to contain u_i words.

The overall summary is $\psi_i(C_i) = (V_i, n_i, W_i, \Phi_i)$

PROPOSED METHODOLOGY

We design an online partition-based clustering methodology, in which a set of clusters $C_1 \dots C_k$ are maintained together with their cluster summaries $\psi_1(C_1) \dots \psi_k(C_k)$.

The overall similarity $Sim(S_i, C_r)$ is computed as linear combination of Structural similarity $SimS(S_i, C_r)$ and content similarity $SimC(S_i, C_r)$ as follows:

$$Sim(S_i, C_r) = \lambda.SimS(S_i, C_r) + (1 - \lambda).SimC(S_i, C_r)$$

where λ is a balancing parameter specified by user in the range (0,1).

Structural Similarity-

The structural similarity $SimS(S_i, C_r)$ is computed as follows:
Define $B(S_i) = (b_1, b_2, \dots, b_{s_r})$ as the bit-vector representation of $R_i \cup \{q_i\}$, which has a bit for each node in V_r . The bit value is 1, if the corresponding node is included in $R_i \cup \{q_i\}$ and otherwise it is 0.

$$SimS(S_i, C_r) = \frac{\sum_{t=1}^{s_r} b_t \cdot v_{rt}}{\sqrt{\|R_i \cup \{q_i\}\| \cdot (\sum_{t=1}^{s_r} v_{rt})}}$$

Content Similarity-

$$SimC(S_i, C_r) = \sum_w f_{S_i}(w) * TF_IDF(w)$$

f_{S_i} is the frequency of word w in the stream S_i .

$TF_IDF(w)$ is calculated considering all the clusters and the stream S_i as documents.

- To maintain the cluster summary, in each iteration, the incoming stream object is assigned to it's closest cluster, unless closest similarity is significantly lower than all the previous values. This is needed in case a new event is to be detected.
- So to maintain the threshold, we'll keep track of the zeroth, first and second moments M_0 , M_1 and M_2 of the closest similarity values continuously.

$$\mu = \frac{M_1}{M_0} \quad \sigma = \sqrt{\frac{M_2}{M_0} - \left(\frac{M_1}{M_0}\right)^2}$$

- Then, Threshold = $\mu - 3.\sigma$

- Suppose stream S_i is found to be closest to cluster C_r then the corresponding cluster summary is updated as follows.
- That is $Sim(S_i, C_r) > \mu - 3.\sigma$

Updating Node Summary

- The frequency of nodes in S_i which are present in V_r are incremented by 1.
- For the nodes in S_i which are not present in V_r , add the corresponding nodes to the node summary of C_r with unit node frequencies.

Updating Word Summary

- A similar approach is followed in maintaining the word summaries.
- The frequency of words in S_i which are present in V_r are incremented by the frequency of the corresponding word in S_i .
- For the words in S_i which are not present in V_r , add the corresponding words to the word summary of C_r with frequencies from the stream S_i .

- On the other hand, if the stream S_i is not similar with any of the ongoing events or previously formed clusters,
That is, $Sim(S_i, C_r) < \mu - 3.\sigma$
then a new singleton cluster with only one entity S_i is formed replacing the most stale cluster.
- Any ties, i.e. , any 2 clusters having same similarity values will have the stream object assigned to exactly one of them randomly.

We'll use a sketch based technique: 'Count-Min Sketch' to maintain the node statistics.

COUNT-MIN SKETCH

A Count-Min sketch with parameters (ϵ, δ) is represented by a two-dimensional array counts with width w and depth d

Given parameters (ϵ, δ) , set $w = \lceil \frac{e}{\epsilon} \rceil$ and $d = \lceil \ln \frac{1}{\delta} \rceil$
Each entry of the array is initially zero. Additionally, d hash functions

$$h_1 \dots h_d : \{1 \dots n\} \rightarrow \{1 \dots w\}$$

are chosen uniformly at random from a pairwise-independent family.

When an update (i_t, c_t) arrives, meaning that item a_{i_t} is updated by a quantity of c_t , then c_t is added to one count in each row; the counter is determined by h_j .

$$\text{count}[j, h_j(i_t)] \rightarrow \text{count}[j, h_j(i_t)] + c_t$$

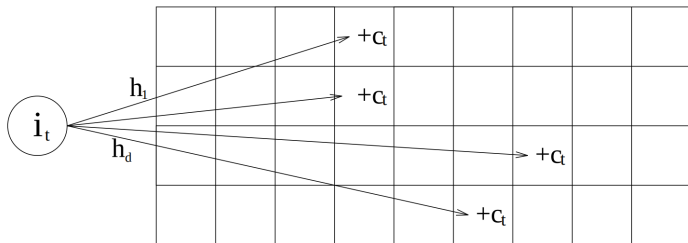


Figure: Each item i is mapped to one cell in each row of the array of counts: when an update of c_t to item i_t arrives, c_t is added to each of these cells

$$\hat{a}_i = \min_j \text{Count}[j, h_j(i)]$$

In order to estimate the count of an item, we determine the set of 'w' cells to which each of the 'w' hash-functions map, and compute the minimum value among all these cells.

PSEUDOCODE

Algorithm SocialStreamClustering (NumClusters: k, λ)

begin

Initialize clusters $C_1 \dots C_r$ to Null

Initialize $i, \mu, \sigma, M_0, M_1, M_2$ to 0

repeat

$i++$

Receive next social stream object S_i

for each cluster C_l Compute $Sim(S_i, C_l)$

Let r be the index of cluster C_r with largest similarity to S_i

if $Sim(S_i, C_r) < \mu - 3\sigma$

Remove most stale cluster and create a singleton cluster with S_i

else

add S_i to C_r and update statistics $\psi_r(C_r)$ of cluster C_r

Update M_0, M_1, M_2 additively

$$\mu = M_1/M_0, \quad \sigma = \sqrt{M_2/M_0 - \mu^2}$$

until(end of stream)

end

DATA COLLECTION

- Enron Dataset (Email Dataset)
- Twitter Dataset (Disaster Datasets from CrisisNLP)

On the previously listed datasets, we removed-

- URLs
- Punctuations, whitespaces, special characters
- stop words
- Non-English content
- performed stemming

Then sorted the dataset according to the Timestamps.

Enron data set:

Number of e-mails before pre-processing - 5,17,401

Number of e-mails after pre-processing - 1,53,002

Twitter data set:

Number of tweets before pre-processing - 3,29,779

Number of tweets after pre-processing - 2,48,134

EXPERIMENTAL RESULTS

Cluster Purity (CP):

- For each cluster, we find the tag with the highest presence and compute the fraction of cluster objects that belong to the tag(label).
- This value is averaged over the different clusters in a weighted way, where the weight was proportional to the number of objects(streams) in it.

$$CP = \frac{\sum_i n(C_i) \cdot f(C_i)}{\sum_i n(C_i)}$$

where $n(C_i)$ is the number of objects in the cluster C_i and $f(C_i)$ is the fraction of total objects in C_i that belong to the tag with the highest presence in the above cluster.

We define tags as follows:

Twitter: Hashtags are treated as tags

Enron: Nouns and unique words in subject of e-mail are treated as tags.

EXPERIMENTAL RESULTS

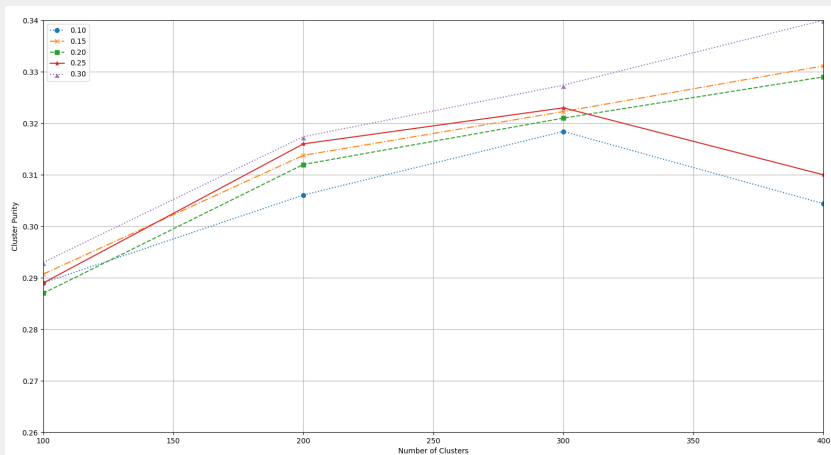


Figure: Average Cluster Purity

- Since emails contain more text, there should be bias towards content similarity. Therefore, we used λ in the range 0.1 to 0.3 for clustering.
- From the above plots, considering the range of values of λ (0.1 - 0.3) with increase in weightage to structural similarity, there is increase in cluster purity. Simultaneously λ should not be too high which might dilute the final similarity.
- Fixing the value of λ , with increase in the number of clusters there is an increase in cluster purity

STREAM-WISE ANALYSIS OF A CLUSTER (EVOLUTION RATIO)

Setting:

Number of Clusters: 500

Text Only: Twitter Dataset

H = 20 mins

EVOLUTION RATIO

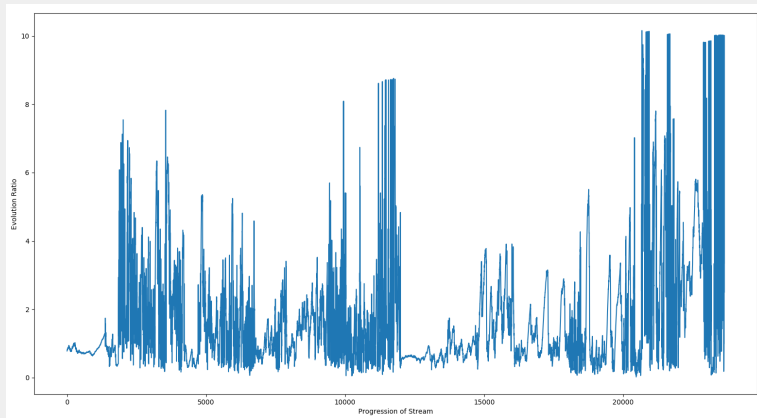


Figure: Evolution Ratio for first Cluster

Most frequent words: *Earthquake, WeatherChannel, Syndrome, Death*

EVOLUTION RATIO

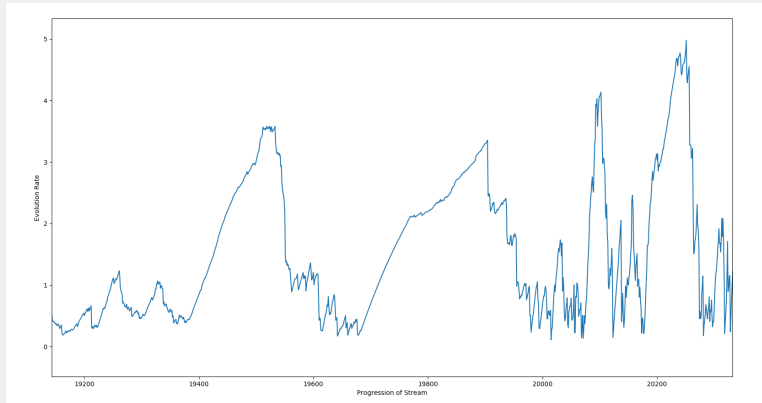


Figure: Evolution Ratio

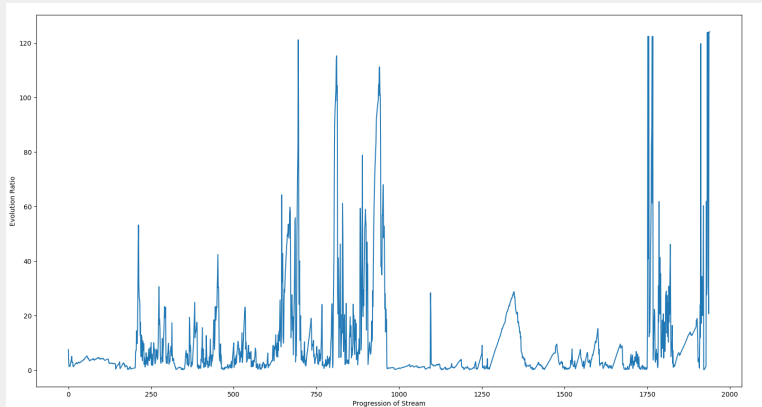


Figure: Evolution Ratio for 31st Cluster



Most frequent words: *Tremor, Landslide, Week, Day, Earthquake*

EVOLUTION RATIO(INFERENCE)

- The above plot helps to understand how each event is evolving with the progression of streams.
- Thus this tells us when an event is active.

THANK YOU

REFERENCES

-  KARTHIK SUBBIAN. CHARU C. AGGARWAL.
EVENT DETECTION IN SOCIAL STREAMS.
SIAM International Conference on Data, 2012.
-  GRAHAM CORMODE A. S. MUTHUKRISHNAN.
AN IMPROVED DATA STREAM SUMMARY: THE COUNT-MIN SKETCH AND ITS APPLICATIONS.
Journal of Algorithms, 2004.