BURSTY EVENT DETECTION THROUGHOUT HISTORIES

Debjyoti Paul, Yanqing Peng, Feifei Li

35th IEEE International Conference on Data Engineering
ICDE 2019 Research Track
OVERVIEW

- Twitter trends

- Real-time trending (bursty) event detection
  - Tells people what’s happening
  - Help people react to important uprising events in their early stages while they are still developing
  - Well studied problem

- Historical Bursty Events:
  - Not a well studied problem but relevant for data scientists.
BURSTINESS

Intuition: Examples of bursty and non-bursty events

- Earthquake: discussed frequently in a time range
- Weather: discussed frequently all the time

Insight: *Bursty* = Surge in incoming rate

Definition: The burstiness of event \( e \) at time \( t \) is

\[
B_e(t) = bf_e(t) - bf_e(t - \tau)
\]

where \( bf_e(t) \) is the incoming rate of event \( e \) within time range \([t - \tau, t)\)
(a) Incoming rate. (b) Burstiness.
HISTORICAL BURSTY EVENTS

- Interesting problem:
  How to query and analyze bursty events from past efficiently?

- Query Examples:
  1. What are the bursty events in the first week of October in 2016?
  2. Is “Anthem Protest” a bursty event in second week of September in 2017?

- Understand and analyze bursty events by going back and forth in time.
Store timeline curves of all events in the history.

Cost: \#events * \#timestamps

Infeasible!!!
Given a temporal stream of events, design an approach to store the stream with compact space, and answer the following queries with theoretical bounded error:

1. Bursty Point Query: How bursty is this event at this time?
   - Query the burstiness value for event $e$ at time $t$

2. Bursty Time Range Query: In which time does this event become bursty?
   - Query the timestamps that the burstiness value of event $e$ is above threshold $\theta$

3. Bursty Event Query: What events are bursty at this time?
   - Query the events that has burstiness value above threshold $\theta$ at time $t$

Focus on Bursty Point Queries, then extend to other queries.
A single event stream represented as a staircase curve.
PBE-1 APPROXIMATION: BUFFERED SOLUTION

- Original data $F(t)$: frequency staircase curve
- Compress data $F^*(t)$: a staircase curve that under the original staircase
  - "Distance" between $F^*(t)$ to $F(t)$ is defined by the area of $F - F^*(t)$
  - Lemma: The corners of the optimal staircase must contain only the corners of $F(t)$
- Select a subset of staircase corner points to form a sub-staircase
  - Dynamic Programming
PBE-2 APPROXIMATION: ONLINE SOLUTION

- Piecewise Linear Approximation
- Use multiple segments to represent the original staircase

(a) Timestamped frequency ranges $A$. 
(b) A PLA $L$ for $A$.

Figure 3: An example of PBE-2.
PBE-2 APPROXIMATION: ONLINE SOLUTION

- Piecewise Linear Approximation
- Use multiple segments to represent the original staircase

Figure 3: An example of PBE-2.
MULTIPLE EVENT STREAM

- Count-Min (CM) Sketch
  - The count-min sketch (CM sketch) is a probabilistic data structure that serves as a frequency table of events in a stream of data
- Combining CM with PBEs
**OTHER TYPES OF QUERIES**

- **Bursty time range query**
  - Check only the corner points

- **Bursty event query**
  - Log N number of CM-PBE where N is number of events.

![Diagram of cumulative count of event mentions over time](image)

Figure 6: Binary decomposition of the event id space.
OTHER TYPES OF QUERIES

- Bursty time range query
  - Check only the corner points

- Bursty event query
  - Log N number of CM-PBE where N is number of events.

Figure 6: Binary decomposition of the event id space.
EXPERIMENT DATASETS

- **OlympicRio**: 50M tweets in August 2016 about Olympic Games Rio with 864 events.
  - Swimming and Soccer

- **USPolitics**: 286M tweets from June 2016 to November 2016 on US politics with 1689 events. Randomly sampled to make it as large as OlympicRio.

Figure 7: Two events in olympicrio. $\tau = 86, 400$ seconds (1 day).
PBE-1 (offline):
- Tradeoff: Error vs Space + Time
- Long construction time (~1min)
- Small space cost
- Low error

PBE-2 (online):
- Tradeoff: Error vs Space
- Short construction time (~10ms)
- Small space cost
- Relatively high error when compared with PBE-1
SINGLE EVENT STREAM

- 300x Space save compared with baseline
- Low error for both approaches, PBE-1 (offline) performs better.

Figure 10: PBE: single event stream.
Multiple Events Stream

- 100x Space save compared with baseline
- 12 GB raw data to 80 MB meta data.
- Low error for both approaches, PBE-1 (offline) performs better.

Figure 11: CM-PBE: Space vs accuracy.
CONCLUSION

- We have unleashed the potential of Bursty Event Detection for past events.
- Existing work focus on Real-time bursty detection, doesn’t discuss on efficient storage for retrieval.
- We propose a framework to answer historical bursty event queries with small space.
  - Single event stream
    - Offline Dynamic Programming: Optimal but requires buffering
    - Online Piecewise Linear Approximation: Fast and no-buffering, but with higher error.
  - Multiple events stream: A variant of Count-Min Sketch
- Supported queries
  - Point query
  - Bursty time range query
  - Bursty event query
REFERENCES

REFERENCES

Figure 1: An example of burst where $\tau = 1$. 
PBE1: OFFLINE OPTIMAL SOLUTION

- Input: $P$, the set of corner points in the original staircase
- Input: $\eta$, the number of points in the output
- Output: $P^*$, a subset of the input points with size $\eta$
- Use Dynamic Programming to calculate optimal $P^*$.
- $\Delta^*(i, j)$: The optimal solution when choosing $i$ points from the first $j$ points in $P$

$$\Delta^*(i, j) = \min \begin{cases} \min_{x \in [i-1, j-1]} \Delta^*(i-1, x) - \delta(j, F^*(i-1, x)) \quad \text{Choose the j-th point} \\ \min_{x \in [i, j-1]} \Delta^*(i, x) \quad \text{Not choose the j-th point} \end{cases}$$

- Buffering in online case
  - Buffer $\eta$ points, run DP, concatenate optimal staircases