Scalable and Near Real-Time Burst Detection from eCommerce Queries

Nish Parikh, Neel Sundaresan

ACM SIGKDD '08

Presenter: Luo Yiming

Outline

- Context in which the problem is bored
- Infinite-state automaton

  Bursty and Hierarchical Structure in Streams--ACM SIGKDD'02
- Main contribution of this work
- Former related work

Main Idea—Bursty and Hierarchical Structure in Streams

- Extract meaningful structure from document stream
- Burst of activity: certain features rising sharply in frequency as the topic emerges
- A formal approach for modeling such "bursts"
  - An infinite-state automaton
  - Bursts appear as state transitions
  - A nested representation of the set of bursts that imposes a hierarchical structure on the overall stream.

A Weighted Automation Model: One State Model

- Generating model:

$$f(x) = \alpha e^{-\alpha x}$$

  - x : the gap in time of two consecutive messages
  - Expectation: α^{-1}
  - α : rate of message arrivals
- Why this model?

A Weighted Automaton Model: Two State Model

- Two states automaton A: q_0, q_1

$$f_0(x) = \alpha_0 e^{-\alpha_0 x} \quad f_1(x) = \alpha_1 e^{-\alpha_1 x}$$

- A changes state with probability p, remaining in its current state with probability 1-p, independently of previous emissions and state changes.
- A begins in state q_0. Before each message is emitted, A changes state with probability p. A message is then emitted, and the gap in time until the next message is determined by the distribution associated with A's current state.

A Weighted Automaton Model: Two State Model

- Based on a set of messages to estimate a state sequence
  - Maximum likelihood

$$\mathbf{x} = (x_1, x_2, \cdots x_n) \qquad \mathbf{q} = (q_{i_1}, q_{i_2}, \cdots q_{i_n})$$

- n inter-arrival gaps:
- A state sequence:
- b denotes the number of state transitions in the sequence $\mathbf{q}$

$$\Pr[\mathbf{q} \mid \mathbf{x}] = \frac{\Pr[\mathbf{q}] f_{\mathbf{q}}(\mathbf{x})}{\sum_{\mathbf{q}'} \Pr[\mathbf{q}'] f_{\mathbf{q}'}(\mathbf{x})} = \frac{1}{Z}\left(\frac{p}{1-p}\right)^b (1-p)^n \prod_{t=1}^{n} f_{i_t}(x_t)$$

A Weighted Automaton Model: Two State Model

- Finding a state sequence q maximizing previous probability is equivalent to finding one that minimizes

$$-\ln \Pr[\mathbf{q}\,|\,\mathbf{x}] = b\ln\left(\frac{p}{1-p}\right) + \left(\sum_{t=1}^{n} -\ln f_{i_t}(x_t)\right) - n\ln(1-p) + \ln Z$$

- Equivalent to minimize the following *cost function*:

$$c(\mathbf{q}\,|\,\mathbf{x}) = b\ln\left(\frac{p}{1-p}\right) + \left(\sum_{t=1}^{n} -\ln f_{i_t}(x_t)\right)$$

Experiment related

- Dataset: 5 months of queries from eBay.com in 2007 (75+ TB of data).
- Assumption and pre-definition:

i) The number of queries uniform distribute over time of one day;

ii) Max number of segments of query arrivals per day is scaled to 48;

iii) Each arrival is represented by a UNIX

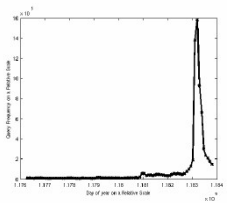

Figure 2 shows the daily frequency of the query 'iphone' per day during the same time period shown in Figure 1. One can see the peak towards the end. Y axis shows relative query frequency.

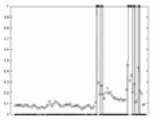

Figure 4 Temporal Pattern in Dotted Line and Optimal State Sequence in Solid Line for Query 'paris hilton'. We see various peaks (spikes) and corresponding automaton jumps indicating multiple burst periods.

$\alpha_0 = average\ rate\ of\ arrival\ for\ query$

$\alpha_1 = 2.5\alpha_0 \quad C = -\ln(0.38)$

Incremental Burst Detection

- Based on the rate of change of percentage volume for a query
- Vs. change of absolute volume—Noiseless;
- Object to batched arrival of new queries– avoid recalculate the entire state sequence when new batch arrives.

Incremental Burst Detection

- $R = \sum_{t=1}^{n} r_t$, $D = \sum_{t=1}^{n} d_t$

  let the t^{th} batch contain r_t instances of Q out of a total of d_t queries, and n is the total number of batches.
- $q_0 : p_0 = R/D$; $q_1 : p_1 = s\ p_0$
- Cost= $\sigma(i, r_t, d_t) = -\ln\left[d_t C_{r_t} p_i^{r_t}(1-p_i)^{d_t - r_t}\right]$, when t^{th} batch comes to state q_i
- Given a time t, the cost of being in state 0

$$C_0(t) = -\ln\left[d_t C_{r_t} p_0^{r_t}(1-p_0)^{(d_t - r_t)}\right] + \min((C_0(t-1) + \tau(0,0)), (C_1(t-1) + \tau(1,0)))$$

and the cost of being in state 1

$$C_1(t) = -\ln\left[d_t C_{r_t} p_1^{r_t}(1-p_1)^{(d_t - r_t)}\right] + \min((C_0(t-1) + \tau(0,1)), (C_1(t-1) + \tau(1,1)))$$

Burst Classification

- Method based: Wavelet transforms
- 4 classes:

  i) Matterhorns; ii) Cuestas;

  iii) Dogtooths; iiii) Hogback.

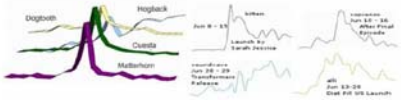

Figure 6 Labeling of Classes. Classes are named based upon the representative shapes of their centroids. X axis represents the time axis (day of the year), with burst period at the center. Y axis shows the relative normalized query frequencies, which gives an indication of the differences in amplitudes between burst and non burst periods for each of the 4 classes. 'bitten' is a Matterhorn, "sopranos" is a Cuesta, "alli" is a Dogtooth and "soundwave" is a Hogback kind of burst.

Sorting and Ranking

- Concentration based ranking

  -Duration of burst (D);

  -Mass (Popularity) of Burst (M);

  -Arrival Rate for Burst (A);

  -Span Ratio (SR);

  -Momentum of Burst (Mo): Mo = (M . A);

  -Concentration of Burst (Xc): $Xc \notin SR^{0.1}$)

Sorting and Ranking

- Distance Based Ranking $\frac{1}{D(I;S)}$

Table 2 Bursts ranked using Distance Based Ranking. For the bursty waveforms X axis indicates time and Y axis indicates relative query frequency

Bursty Query	Bursty Waveform	Burst Intensity *B*
wallet (Rank 1)		1.011
paris hilton (Rank 5)		0.929
treo 755p (Rank 341)		0.049

Performance Compare

Table 4 Table indicating calculated values of α and β for five different days using method discussed above

Sample Number	$\alpha = \frac{n(Gs \cap R)}{n(Gs)}$	$\beta = \frac{n(Ga \cap R)}{n(R)}$
1	0.33	0.27
2	0.27	0.31
3	0.40	0.37
4	0.38	0.31
5	0.16	0.26
Average	0.3	0.3

Implementation

Figure 10 Screenshot of an application using top burst of day to create a mash-up and attract online users based on curiosity

Thank You!