Applications of Count-Min Sketch

Zixuan Chen, Leiqi Ye

University of Michigan

December 10, 2022

Overview

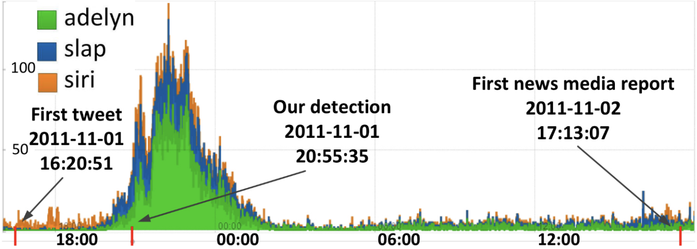
1. TopicSketch: Real-Time Bursty Topic Detection from Twitter

2. Extreme Classification in Log Memory using Count-Min Sketch: A Case Study of Amazon Search with 50M Products

3. Processor Security: Detecting Microarchitectural Attacks via Count-Min Sketches

1. TopicSketch: Real-Time Bursty Topic Detection from Twitter

In a real-time stream of tweets, we want to identify bursty topics: which gain their popularity recently.



First tweet
2011-11-01
16:20:51

Our detection
2011-11-01
20:55:35

First news media report
2011-11-02
17:13:07

Figure 1. The tweet volume of each of the top three keywords of the topic: "adelyn", "slap" and "siri".

1. TopicSketch: formulation of the problem

- There are N words in total.
- $D(t) = \{\text{tweets generated up to time } t\}$
- For a tweet d, Let $d(i) =$ times that word i appears in tweet d. Let $|d|$=number of words in d;
- A topic T_k: a fixed distribution $p_k \in \mathbb{R}^N$ over words.
- $\lambda_k(t)$: P(a tweet about $T_k$is generated in $[t, t + dt)$).

Assume that at time t, there are at most K topics with positive $\lambda_k(t)$.

Our objective:

- Identify topic T_k with large $\lambda'_k(t)$, and its key words (words with high probability in T_k).

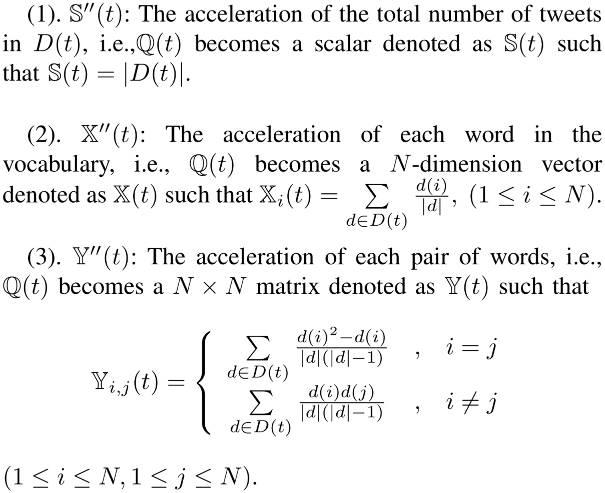
1. TopicSketch: formulation of the problem

Challenges: how to maintain proper amount of statics?

- What information is needed?
- How to maintain as few data as possible?

1. TopicSketch: calculate p_k, $\lambda'_k(t)$ based on the following

Input: $D(t)$ Store: $S''(t), X''(t), Y''(t)$



(1). $\mathbb{S}''(t)$: The acceleration of the total number of tweets in $D(t)$, i.e.,$\mathbb{Q}(t)$ becomes a scalar denoted as $\mathbb{S}(t)$ such that $\mathbb{S}(t) = |D(t)|$.

(2). $\mathbb{X}''(t)$: The acceleration of each word in the vocabulary, i.e., $\mathbb{Q}(t)$ becomes a N-dimension vector denoted as $\mathbb{X}(t)$ such that $\mathbb{X}_i(t) = \sum_{d \in D(t)} \frac{d(i)}{|d|}$, $(1 \leq i \leq N)$.

(3). $\mathbb{Y}''(t)$: The acceleration of each pair of words, i.e., $\mathbb{Q}(t)$ becomes a $N \times N$ matrix denoted as $\mathbb{Y}(t)$ such that

$$\mathbb{Y}_{i,j}(t) = \begin{cases} \sum\limits_{d \in D(t)} \frac{d(i)^2 - d(i)}{|d|(|d|-1)} & , \quad i = j \\ \sum\limits_{d \in D(t)} \frac{d(i)d(j)}{|d|(|d|-1)} & , \quad i \neq j \end{cases}$$

$(1 \leq i \leq N, 1 \leq j \leq N)$.

TopicSketch: calculate p_k, $\lambda'_k(t)$ based on S, X, Y

They are easy to update whenever a new tweet d comes.
To solve $\alpha_k(t), p_k$ for $k = 1, 2, ..., K$ can be reduced to solve an optimization problem related to vectors $S''(t), X''(t)$ and matrix $Y''(t)$.

1. TopicSketch: Procedure

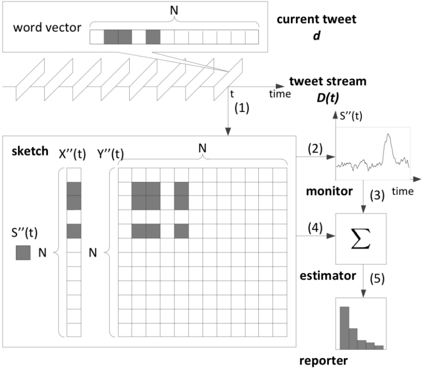


Figure 2. TopicSketch Framework Overview

The monitor tracks the sketch using simple calculation. Only when an update leads to a difference larger than a pre-determined threshold will it notify the estimator to infer the bursty topics.

1. TopicSketch: tackle the challenge using count-min sketch

Challenge: Even if we only consider words appears in the past 15 minutes, N is too large.

- It takes $O(N^2)$ space to store the matrix $Y''(t)$.
- The optimization problem takes $O(N \cdot K)$.

Solution: count-min sketch!

- Map N words to B buckets. View buckets as "words".
- Uniform and independent hash functions $\mathcal{H}_1, \mathcal{H}_2, ..., \mathcal{H}_H : [1...N] \to [1...B]$
- Distribution of the bucket j in topic T_k under the $h-th$ hash function: $p^h_{k,j} = \sum_{i:\mathcal{H}_h(i)=j} p_{k,i},\ j = 1, 2, ..., B.$
- Distribution of the word i in topic T_k: $min_{1 \leq h \leq H}\, p^h_{k,\mathcal{H}_h(i)}$
- Only need to return key words: words with distribution > 0.02

1. TopicSketch: evaluation

- 32479134 tweets
- 8470180 distinct words in total
- 10000 to 20000 words in the past 15 minutes
- $B = 300$, $H = 5$

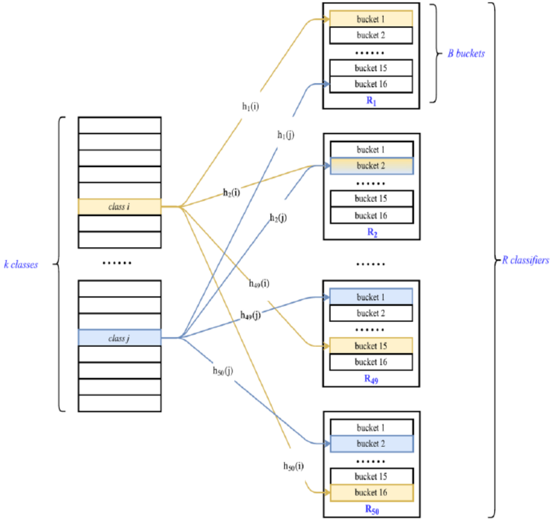
2. Amazon Search: Background

- Popular NLP models predict the best word, given the full context observed so far. All those context will lead to different status. For a large dataset, the vocabulary size can quickly run into billions.

2. Amazon Search: Core problem

The core problem of this application is: For a certain item, it will have different probability of different status. In some example, there are over billions status.

2. Amazon Search: Count-Min Sketch



Processor Security: Background

- SCA (side-channel analysis) exploits unintended information leakage by analyzing timing information, power consumption, thermal footprint, or electromagnetic emanation of computing systems while executing security primitives to extract information about the processed data and then use them to infer sensitive information.

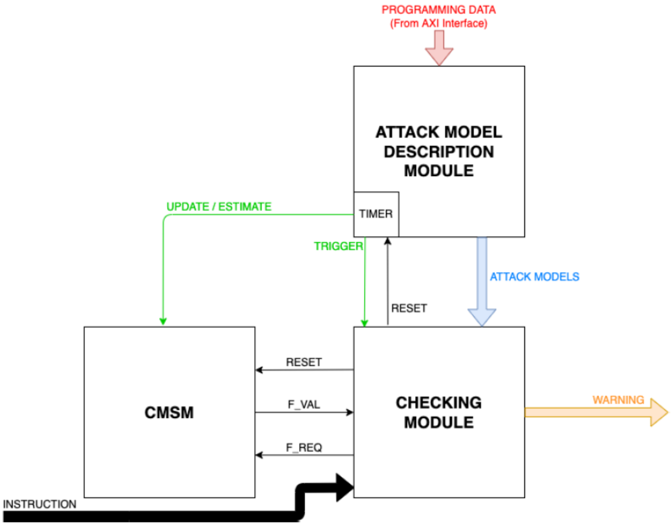
3. Processor Security: Background2

- As an example, a wellknown microarchitectural attack is Spectre , where the attacker takes advantage of speculative execution to break address space isolation without exploiting any software bug. By exploiting Spectre, an attacker allows his/her own program to access the memory (and, thus, also secrets) of other programs and the operating system.

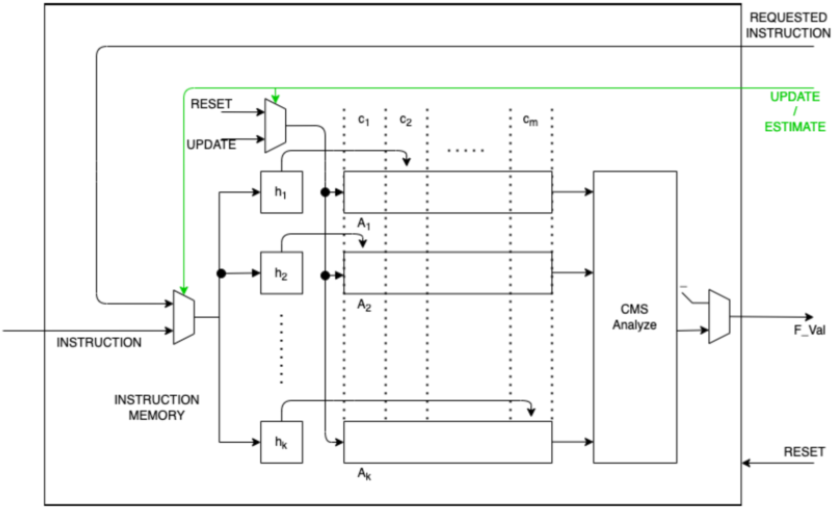
3. Processor Security: Solution & Core Problem

- The security checker (SC) observes and keeps track of all the instructions fetched by the microprocessor. Moreover, thanks to a CMS, the SC is able to estimate the occurrence frequency of a set of instruction sequences.
- But there is over 2^{64} different intructions.

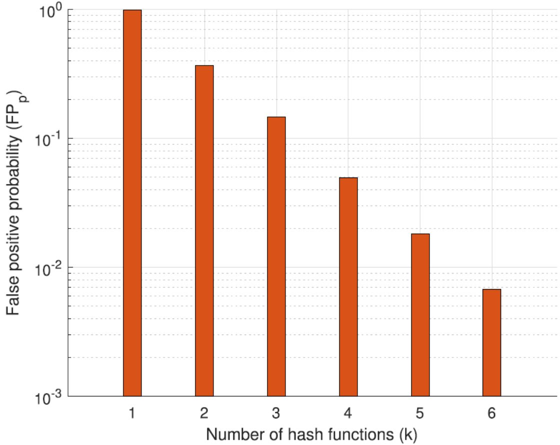
3. Processor Security: Security checker



3. Processor Security: Count-Min Sketch



3. Processor Security: Evaluation



Conclusion

- In the example, the Count-Min Sketch analysis a lots of problem which constrains the range of a[x] in 0-1. So, improvement on the subproblem of Count-Min Sketch will get a better solution in those problems we discussed above.