Time Adaptive Sketches (Ada-Sketches) for Summarizing Data Streams

Anshumali Shrivastava (Rice), Arnd Christian König (Microsoft) and Misha Bilenko (Microsoft)

anshumali@rice.edu

30th June 2016
Motivation: Counting over Temporal Data Streams

Counting over temporal data streams is a common problem.

E.g.: Given query, user and context, predict probability of click.

**Given:**

- **Basic Features:** userid, query, timestamp, ad-url, ad content, more categorical features like browser type, etc.
- **Target:** (1/0) Click/Non-Click
Motivation: Counting over Temporal Data Streams

Counting over temporal data streams is a common problem.

E.g.: Given query, user and context, predict probability of click.

Given:

- **Basic Features**: userid, query, **timestamp**, ad-url, ad content, more categorical features like browser type, etc.

- **Target**: (1/0) Click/Non-Click

**Most Information in Count Features:**

- A significant burst of clicks on some “XYZ” category (or combinations) in past few minutes.

- Features based on frequencies of clicks/non-clicks conditional on categories, over time, is generally adopted in practice.
Motivation: Counting over Temporal Data Streams

Counting over temporal data streams is a common problem.

E.g.: Given query, user and context, predict probability of click.

Given:

- **Basic Features**: userid, query, timestamp, ad-url, ad content, more categorical features like browser type, etc.
- **Target**: (1/0) Click/Non-Click

**Most Information in Count Features:**

- A significant burst of clicks on some “XYZ” category (or combinations) in past few minutes.
- Features based on frequencies of clicks/non-clicks conditional on categories, over time, is generally adopted in practice.

**Scale: Needle in haystack.**

- Billions of users and tens of millions of ads.
- Combinations blow up. Just \((ad \times query) \sim 10^{10+}\)
- 50ms response time, update statistics with observations.
The Classical (Non-Adaptive) Approximate Counting:

**Setting:**
- We are given a huge number of items (co-variates) $i \in I$ to track over time $t \in \{1, 2, ..., T\}$. $T$ can be large as well.
- We only see increments $(i, t, v)$, the increment $v$ to item $i$ at time $t$.

**Goal:** In limited space (hopefully $O(\log |I| \times T)$), we want to
- **Point Queries:** Estimate the counts (increments) of item $i$ at time $t$.
- **Range Queries:** Estimate the counts (increments) of item $i$ during the given range $[t_1, t_2]$.

**Classical Sketching:** Count-Min Sketch (CMS), Lossy Counting, etc.
Basic Idea behind Sketching.

Randomly assign items to a small number of counters.
- It works! AMS 85, Moody 89, Charikar 99, MuthuKrishnana 02, etc.
- If no collisions, counts exact.

Handling Time:
- Treat each pair \((i, t)\) (item, time) as different item.
- Hash pairs \((i, t)\), instead of just items.
- Time only increases the number of items to \(|I| \times T\).
What happens during Collision?

The Good

We typically care about heavy hitters.
What happens during Collision?

The Good

The

Irrelevant

We typically care about heavy hitters.
What happens during Collision?

The Good

The Irrelevant

The Unlucky

We typically care about heavy hitters.
Maximizing Luck: Count-Min Sketch (CMS)

**Idea:**
- We always overestimate, if unlucky, by a lot.
- Repeat independently $d$ times and take minimum of all overestimates.
- Unless unlucky all $d$ times, it will work. ($d = \log \frac{1}{\delta}$, $w = \frac{1}{\epsilon}$)

**Theoretical Guarantee**
- $c \leq \hat{c} \leq c + \epsilon M^T$ with probability $1 - \delta$, where $M^T$ is sum of all counts in the stream.
- Space $O(\log |I| \times T)$
New Requirement: Time Adaptability

In Practice:
- Recent trends are more important.
- A burst in the number of clicks in the past few minutes more informative than similar burst last month.

Expectation: Time Adaptive Counting.
- Classical sketches do not take temporal effect into consideration.
- **Smart Tradeoff:** Given the same space, trade errors of recent counts with that of older ones.
- Like our memory, forget slowly.
Existing Solution: Hokusai

\[ t = T \ (A^T) \]

\[ t = T-1 \ (A^{T-1}) \]

\[ t = T-2 \ (A^{T-2}) \]

\[ t = T-3 \ (A^{T-3}) \]

\[ t = T-4 \ (A^{T-4}) \]

\[ t = T-5 \ (A^{T-5}) \]

\[ t = T-6 \ (A^{T-6}) \]

**Idea:** Disproportionate allocation over time.

- Accuracy of CMS dependent on memory allocated.
- More space for recent sketches and less for older.
- Keep a CMS sketch for every time. Shrink sketch size on fly.

**Clever Idea:** Exploit Rollover.

\(^1\)Matusevych, Smola and Ahmad 2012
## Existing Solution: Hokusai

$$t = T \ (A^T)$$

$$t = T-1 \ (A^{T-1})$$

$$t = T-2 \ (A^{T-2})$$

$$t = T-3 \ (A^{T-3})$$

$$t = T-4 \ (A^{T-4})$$

$$t = T-5 \ (A^{T-5})$$

$$t = T-6 \ (A^{T-6})$$

### Idea:
Disproportionate allocation over time.

- Accuracy of CMS dependent on memory allocated.
- More space for recent sketches and less for older.
- Keep a CMS sketch for every time. Shrink sketch size on fly.

**Clever Idea:** Exploit Rollover.

---

1Matusevych, Smola and Ahmad 2012
Existing Solution: Hokusai

\[ t = T \ (A^T) \]
\[ t = T-1 \ (A^{T-1}) \]
\[ t = T-2 \ (A^{T-2}) \]
\[ t = T-3 \ (A^{T-3}) \]
\[ t = T-4 \ (A^{T-4}) \]
\[ t = T-5 \ (A^{T-5}) \]
\[ t = T-6 \ (A^{T-6}) \]

**Idea:** Disproportionate allocation over time.
- Accuracy of CMS dependent on memory allocated.
- More space for recent sketches and less for older.
- Keep a CMS sketch for every time. Shrink sketch size on fly.

**Clever Idea:** Exploit Rollover.

---

Matusevych, Smola and Ahmad 2012

1Matusevych, Smola and Ahmad 2012
Existing Solution: Hokusai

Idea: Disproportionate allocation over time.

- Accuracy of CMS dependent on memory allocated.
- More space for recent sketches and less for older.
- Keep a CMS sketch for every time. Shrink sketch size on fly.

Clever Idea: Exploit Rollover.

1Matusevych, Smola and Ahmad 2012
Problems with Hokusai

Issues:

- Discontinuity. If time $t$ is empty, we still have to shrink sketch size for older times.

- $O(T)$ memory. One for each $t$.

- Shrinking overhead. Shrink log $t$ sketches for every transition from $t$ to $t+1$.

- No flexibility.
Detour: Dolby Noise Reduction (1960s)

High Level View

- In digital recording, the music signal compete with tape hiss (background noise).
- If Signal to Noise (SNR) ratio is high, the recording is noise free.
- While recording the frequencies in the music is artificially inflated (Pre-Emphasis).
- During playback a reverse transformation is applied which cancels pre-emphasis. (De-Emphasis)
- Overall effect of noise is minimized.
Proposal: (Adaptive) Ada-Sketches

Analogy with Dolby Noise Reduction:

- Sketches preserves heavier counts more accurately.
- Artificially inflate recent counts (Pre-emphasis).
- Inflated counts will be preserved with less error.
- Deflate by exact same amount during estimation. (De-emphasis)
Proposal: (Adaptive) Ada-Sketches

Analogy with Dolby Noise Reduction:
- Sketches preserves heavier counts more accurately.
- Artificially inflate recent counts (Pre-emphasis).
- Inflated counts will be preserved with less error.
- Deflate by exact same amount during estimation. (De-emphasis)

![Diagram showing data stream, pre-emphasis, sketch, query, de-emphasis, and result]

Proposal
- Let $f(t)$ be any (pre-defined) monotonically increasing sequence. ($f(t)$ can be chosen wisely)
- Multiply the count of $(i, t)$ with $f(t)$ and then add to the sketch.
- While querying $(i, t)$, get the estimate and divide by $f(t)$
Why it works?

Observation

- If no collision then exact.
- During collision, errors or recent counts decrease due to greater de-emphasis.
Why it works?

Observation

- If no collision then exact.
- During collision, errors or recent counts decrease due to greater de-emphasis.
Why it works?

Observation

- If no collision then exact.
- During collision, errors or recent counts decrease due to greater de-emphasis.
Advantages

- No Discontinuity. If time $t$ is empty, no addition, no extra collisions, no extra errors.
- $O(\log |I| \times T)$ memory just like CMS.
- No shrinking overhead. Minimum modification to CMS. (Strict Generalization)

Provable Time Adaptive Guarantees

**Theorem**

For $w = \lceil \frac{e}{\epsilon} \rceil$ and $d = \log \frac{1}{\delta}$, given any $(i, t)$ we have

$$c_i^t \leq \hat{c}_i^t \leq c_i^t + \epsilon \beta^t \sqrt{M_2^T}$$

with probability $1 - \delta$. Here $\beta^t = \sqrt{\sum_{t'=0}^{T} (f(t'))^2} / f(t)$ is the time adaptive factor monotonically decreasing with $t$. 
Works with any Sketching Algorithm

- Adaptive Count Sketches, Adaptive Lossy Counting etc.
- Provable Time Adaptive Guarantees for all of them.
More..

Works with any Sketching Algorithm
- Adaptive Count Sketches, Adaptive Lossy Counting etc.
- Provable Time Adaptive Guarantees for all of them.

Flexibility in Choice of $f(t)$
- Any monotonic $f(t)$ works. Can be tailored
- Upper bound dependent on $\beta_t = \sqrt{\frac{\sum_{t'=0}^{T} (f(t'))^2}{f(t)}}$.
- Fine control over the error distributions.
Experiments: Accuracy for a given Memory

Figure: Mean and Standard deviation of errors for $w = 2^{18}$. 
Scalability: Throughput

**Table:** Time in sec to summarize AOL dataset

<table>
<thead>
<tr>
<th></th>
<th>$2^{20}$</th>
<th>$2^{22}$</th>
<th>$2^{25}$</th>
<th>$2^{27}$</th>
<th>$2^{30}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMS</td>
<td>44.62</td>
<td>44.80</td>
<td>48.40</td>
<td>50.81</td>
<td>52.67</td>
</tr>
<tr>
<td>Hoku</td>
<td>68.46</td>
<td>94.07</td>
<td>360.23</td>
<td>1206.71</td>
<td>9244.17</td>
</tr>
<tr>
<td>ACMS (lin)</td>
<td>44.57</td>
<td>44.62</td>
<td>49.95</td>
<td>52.21</td>
<td>52.87</td>
</tr>
<tr>
<td>ACMS (exp)</td>
<td>68.32</td>
<td>73.96</td>
<td>76.23</td>
<td>82.73</td>
<td>76.82</td>
</tr>
</tbody>
</table>

**Table:** Time in sec to summarize Criteo Dataset

<table>
<thead>
<tr>
<th></th>
<th>$2^{20}$</th>
<th>$2^{22}$</th>
<th>$2^{25}$</th>
<th>$2^{27}$</th>
<th>$2^{30}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMS</td>
<td>40.79</td>
<td>42.29</td>
<td>45.81</td>
<td>45.92</td>
<td>46.17</td>
</tr>
<tr>
<td>Hoku</td>
<td>55.19</td>
<td>90.32</td>
<td>335.04</td>
<td>1134.07</td>
<td>8522.12</td>
</tr>
<tr>
<td>ACMS (lin)</td>
<td>39.07</td>
<td>42.00</td>
<td>44.54</td>
<td>45.32</td>
<td>46.24</td>
</tr>
<tr>
<td>ACMS (exp)</td>
<td>69.21</td>
<td>69.31</td>
<td>71.23</td>
<td>72.01</td>
<td>72.85</td>
</tr>
</tbody>
</table>
Summary

- Randomized counting algorithms are very powerful for massive data streams.
- Pre-emphasis and De-emphasis idea give more control over distributing errors more smartly.
- Adaptive Sketches are strict generalization of the existing sketching algorithms, which unify the tradeoffs between memory, accuracy and time adaptability in one framework.
- We see significant improvements both in accuracy and speed by smartly balancing this tradeoff.
- The final algorithm is principled with provable guarantees, and it can be easily tailored to incorporate a variety of application dependent constraints.