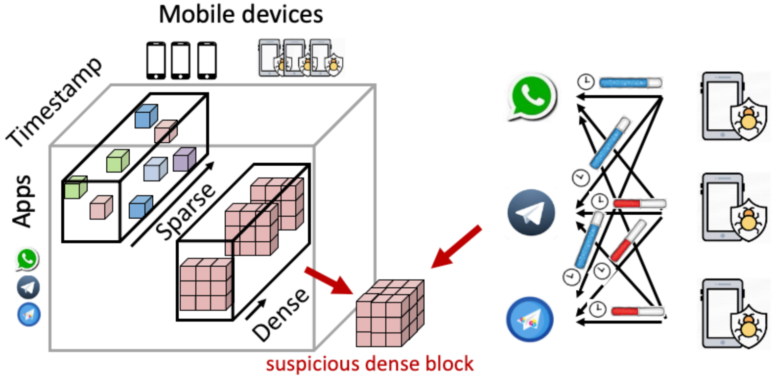
Summary

- **Goal:** to detect and update dense subtensors in streaming tensors
- **Previous Work**
  - showed that dense subtensors signal anomalies or fraud
  - batch algorithms for fast and accurate dense subtensor detection
- **Algorithm**: incremental algorithm for detecting the densest subtensor
- **Result:**
  - **fast:** up to 320x faster than best streaming methods,
  - **robust:** splicing theory to do incremental splices for dense block,
  - **accurate:** successfully detect anomalies in real-world tensors, including App rank boosting fraud, and rating manipulations.

Motivation

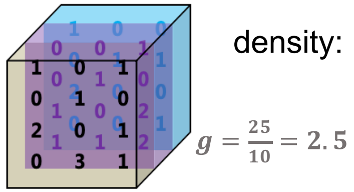
- **Synchronized behavior in App data:** rank boosting fraud results in dense subtensors



How can we detect dense subtensors incrementally in streaming tensors ?

Basic Concepts

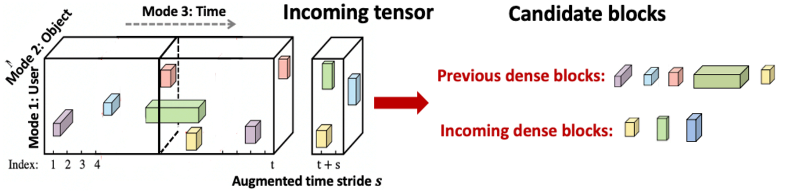
a 3-mode tensor



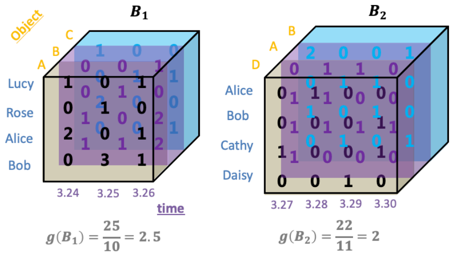
density: $g(\cdot) = \frac{M(\cdot)}{S(\cdot)}$ the sum of entries / number of attributes

Proposed Algorithm: AugSplicing

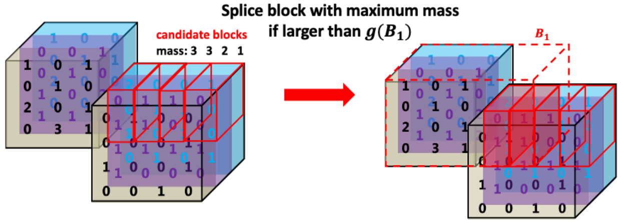
- **Goal**: to incrementally update dense subtensors while the input tensor changes
- **Overall algorithm:** iteratively choose two blocks from candidate blocks and splice these two blocks until reaches the splicing threshold, and output top k dense blocks.



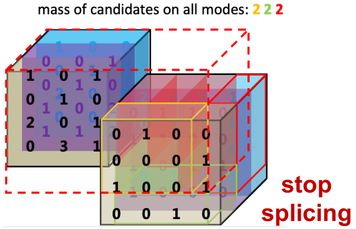
- **Procedure of splicing two blocks:**
  - splice B_2 into B_1 if $g(B_1) \geq g(B_2)$ to make $g(B_1)$ increase

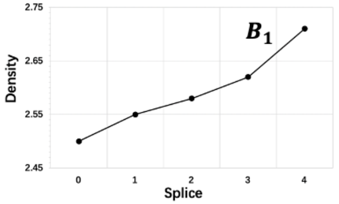


(1) splice on non-overlapped modes, i.e. *time* mode.
  - ✓ choose one attribute from time mode and overlapped attributes from other modes to generate candidate blocks.



(2) randomly choose a mode to splice until no large-mass blocks





Splicing Theorem

Theorem 1 (Splicing Condition). *Given two blocks $\mathcal{B}_1$, $\mathcal{B}_2$ with $g(\mathcal{B}_1) \geq g(\mathcal{B}_2)$, $\exists \mathcal{E} \subseteq \mathcal{B}_2$ such that $g(\mathcal{B}_1 \cup \mathcal{E}) > g(\mathcal{B}_1)$ if and only if*

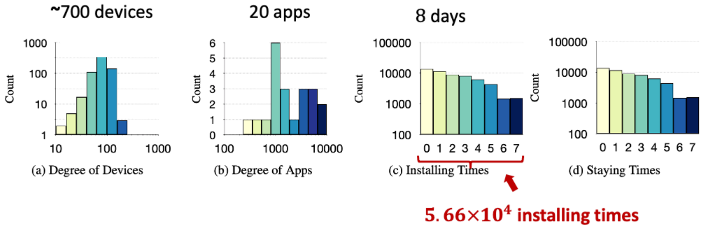
$$M(\mathcal{E}) > \sum_{n=1}^{N} r_n \cdot g(\mathcal{B}_1) = Q \cdot g(\mathcal{B}_1), \qquad (1)$$

the number of new attributes $\mathcal{E}$ brings into B_1

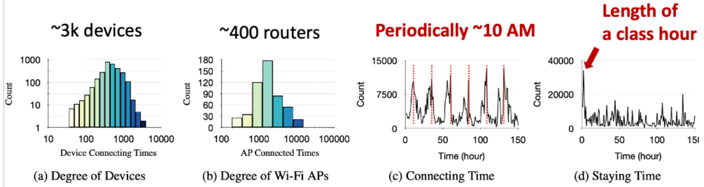
Experimental Results

- **Q1 Effectiveness**: what does AugSplicing detect in real-world tensors?

(1) App installation and uninstallation data



(2) Wi-fi router connection and disconnection data



- **Q2 Speed:** How fast is AugSplicing compared to baselines?
- **Q3 Accuracy:** How accurately does AugSplicing update a dense subtensor?