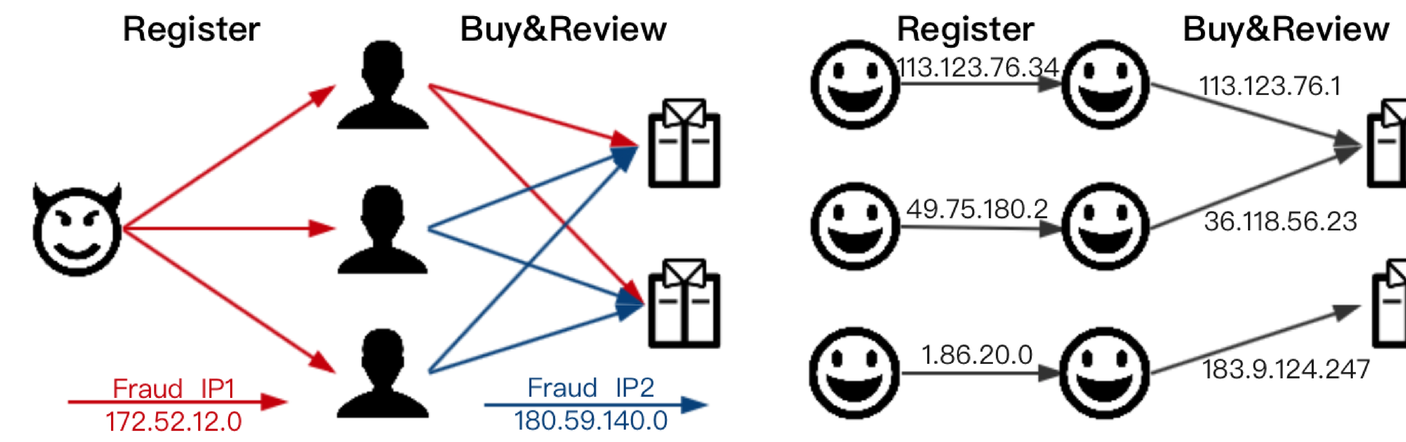
No Place to Hide: Catching Fraudulent Entities in Tensors

Yikun Ban[1], Xin Liu[2], Yitao Duan[3], Xue Liu[4], Wei Xu[2]

[1]Peking University [2]Tsinghua University [3]Fintech.ai [4]McGill University.

Introduction



Detecting Synchronized Behavior. To maximize profits, fraudsters reuse different resources (e.g., fake accounts, IP addresses, and device IDs) over multiple frauds.

Challenges.

- Search-based dense block detection methods are not resistant to noise.
- Tensor decomposition methods tend to miss small fraud groups.
- Semi-supervised fraud detection methods rely on labels difficult to obtain.

Methods

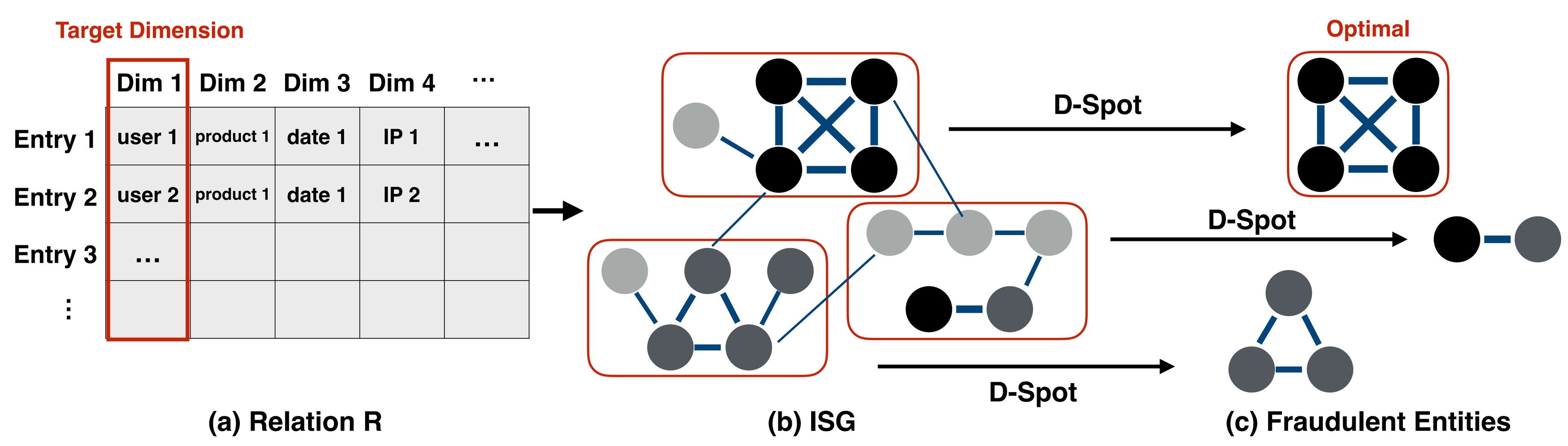


Figure 1: Workflow of ISG+D-Spot.

Input Tensor – Relation R.

- Entries $\{t_0, \cdots, t_{|X|}\}$. Each entry $t = (a_1, \cdots, a_N, x)$ has N dimensional features and an identifier x. In Figure 1, the entry is (user, product, date, IP,...,x).
- Target dimension U. For example, we want to detect fraud users in Figure 1.

Step1: Building ISG

Information Sharing Graph (ISG).

- The probability an entry has a at dimension A_k: $p^k(a) = \Pr(t[A_k] = a)$.
- The self information of the event that u_i and u_j share a at dimension A_k

$$I^k_{i,j}(a) = \log(\frac{1}{p^k(a)})^2.$$

- **Edge weight** $S_{i,j}$: the suspicious level between entities' sharing. We use the pairwise value sharing between u_i and u_j across all dimensions

$$S_{i,j} = \sum_{k=1}^{K} \sum_{a:\text{shared by } u_i,u_j} I^k_{i,j}(a).$$

- The self informaion of the event that u_i uses a at dimension A_k for m times

$$I^k_i(a) = \log(\frac{1}{p^k(a)})^m.$$

- **Node weight** S_i: the suspicious level of the node. We use the self-value sharing for u_i across all dimensions

$$S_i = \sum_{k=1}^{K} \sum_{a:\text{used by } u_i} I^k_i(a).$$

Subgraph Formed by Fraud Groups on ISG.

- Large node weights. For example, a fraud user reusing the IPs to review fraud products has a large node weight.
- Large edge weights. For example, two fraud users share the same IP and review the same unpopular product on Amazon. Because this sharing event has a low probability and high information, the edge weight is large.
- Large group size. Many users in one fraud.
- The edge density is close to 1.0. For example, all fraud users review the same products, so they are all connected.
- Small edge weights between legitimate entities and fraud entities. For example, fraud users review different products from normal users.

Step2: D-Spot

Graph Paritition. Delete low-weight edges and partition the ISG into multiple connected components $\mathcal{G}_1, \mathcal{G}_2, \ldots$. So the later computation could run in parallel.

Finding One Dense Subgraph from One Graph Partition.

- Input: one graph paritition $\mathcal{G}$ of ISG.
- Objective: finding a subgraph $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ on $\mathcal{G}$ that maximizes the suspiciousness density

$$\mathcal{F}_{\hat{\mathcal{G}}} = \frac{\sum_{(u_i,u_j)\in\hat{\mathcal{E}}} S_{i,j} + \sum_{u_i\in\hat{\mathcal{V}}} S_i}{|\hat{\mathcal{V}}|}.$$

- Algorithm: In each iteration, we delete a set of nodes that leads the density $\mathcal{F}$ decreases least from the current node set. Finally, the algorithm returns the node set that maximizes $\mathcal{F}$.

Finding Multiple Dense Subgraphs in Parallel. In each graph partition $\mathcal{G}_1, \mathcal{G}_2, \ldots$, find a single dense subgraph. Then we delete it and find the next dense subgraph.

Advantages

- Two-approximation guarantee of D-Spot.
- By removing a set of nodes at once, we reduce the number of iterations.
- Low computation complexity. $O(|\mathcal{V}|^2 + |\mathcal{E}|)$ for each graph partition.
- Robustness to noisy features.

Evaluation

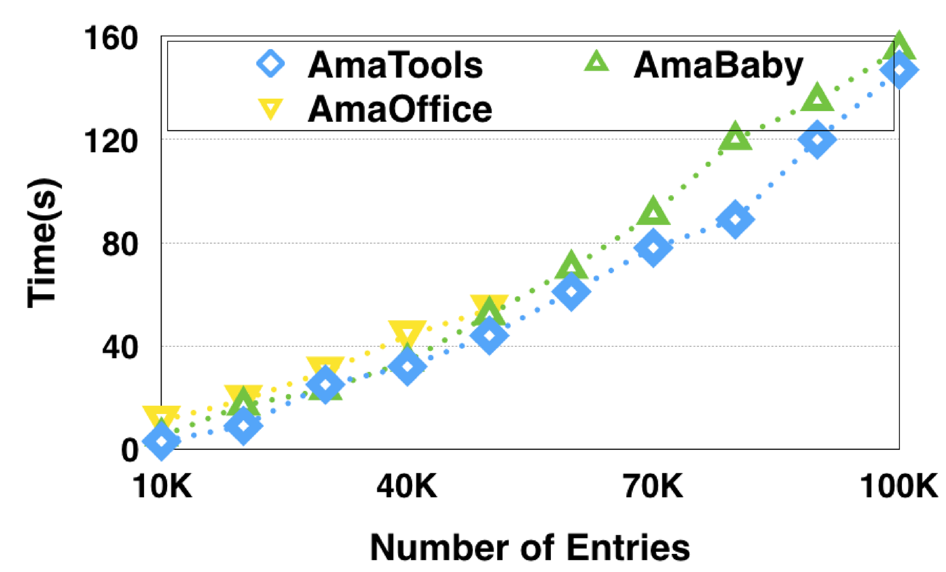
Accurate Fraud User Detection. Three restaurant review datasets from Yelp.

AUC	YelpChi	YelpNYU	YelpZip
	67K Entries	359K Entries	1.14M Entries
M-Zoom[1]	0.9831	0.9451	0.9426
M-Biz[1]	0.9831	0.9345	0.9403
D-Cube[2]	0.9810	0.9223	0.9376
ISG+D-Spot	0.9875	0.9546	0.9529

Robustness to Noisy Features. Registration information of 16,154 normal users and 9,961 fraud users. 'C' = 'crucial feature' and 'N' = 'noisy feature'.

AUC	1C	2C	2C+1N	2C+2N	2C+3N
M-Zoom[1]	0.7676	0.8880	0.8827	0.8744	0.8439
M-Biz[1]	0.7677	0.8842	0.8827	0.8744	0.8439
D-Cube[2]	0.7522	0.9201	0.8586	0.8312	0.7987
ISG+D-Spot	0.7699	0.9946	0.9935	0.9917	0.9859

High Scalability. Near-linear time with respect to the number of entries on three Amazon review datasets.



	Edge Density
AmaOffice	0.0534
AmaBaby	0.0211
AmaTools	0.0171

References

[1] Kijung Shin, Bryan Hooi, and Christo Faloutsos.
Fast, accurate, and flexible algorithms for dense subtensor mining. tkdd, 12(3), 2018.

[2] Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos.
D-cube: Dense-block detection in terabyte-scale tensors. wsdm, 2017.

The World Wide Web Conference, May 13-17, 2019, San Francisco, USA