Fast, Accurate and Flexible Algorithms for Dense Subtensor Mining

KIJUNG SHIN, BRYAN HOOI and CHRISTOS FALOUTSOS, Carnegie Mellon University

Given a large-scale and high-order tensor, how can we detect dense subtensors in it? Can we spot them in near-linear time but with quality guarantees? Extensive previous work has shown that dense subtensors, as well as dense subgraphs, indicate anomalous or fraudulent behavior (e.g., lockstep behavior in social networks). However, available algorithms for detecting dense subtensors are not satisfactory in terms of speed, accuracy, and flexibility. In this work, we propose two algorithms, called M-ZOOM and M-BIZ, for fast and accurate dense-subtensor detection with various density measures. M-ZOOM gives a lower bound on the density of detected subtensors, while M-BIZ guarantees the local optimality of detected subtensors. M-ZOOM and M-BIZ can be combined, giving the following advantages: (1) **Scalable**: scale near-linearly with all aspects of tensors and are up to **114× faster** than state-of-the-art methods with similar accuracy, (2) **Provably accurate**: provide a guarantee on the lowest density and local optimality of the subtensors they find, (3) **Flexible**: support multi-subtensor detection and size bounds as well as diverse density measures, and (4) **Effective**: successfully detected edit wars and bot activities in Wikipedia, and spotted network attacks from a TCP dump with near-perfect accuracy (**AUC=0.98**).

CCS Concepts: •**Information systems** → **Data mining;** •**Computing methodologies** → *Anomaly detection;*

Additional Key Words and Phrases: Tensor, Dense Subtensor, Anomaly Detection, Fraud Detection

ACM Reference Format:
Kijung Shin, Bryan Hooi, and Christos Faloutsos, 2017. Fast, Accurate and Flexible Algorithms for Dense Subtensor Mining *ACM Trans. Knowl. Discov. Data.* V, N, Article A (January YYYY), 30 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

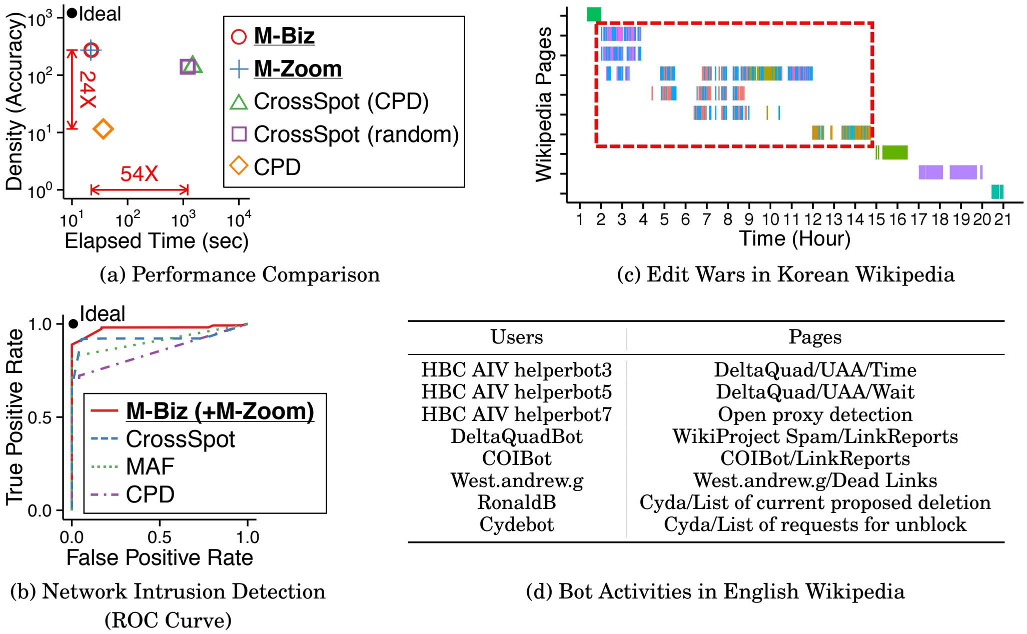
Imagine that you manage a social review site (e.g., Yelp) and have the records of which accounts wrote reviews for which restaurants. How do you detect suspicious lockstep behavior: for example, a set of accounts which give fake reviews to the same set of restaurants? What about the case where additional information is present, such as the timestamp of each review, or the keywords in each review?

Such problems of detecting suspicious lockstep behavior have been extensively studied from the perspective of dense-subgraph detection. Intuitively, in the above example, highly synchronized behavior induces dense subgraphs in the bipartite review graph of accounts and restaurants. Indeed, methods which detect dense subgraphs have been successfully used to spot fraud in settings ranging from social networks [Beutel et al. 2013; Hooi et al. 2017; Jiang et al. 2014a; Jiang et al. 2014b; Shin et al. 2016a], auctions [Pandit et al. 2007], and search engines [Gibson et al. 2005].

This material is based upon work supported by the National Science Foundation under Grant No. CNS-1314632 and IIS-1408924. Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. K. Shin was supported by KFAS Scholarship.

Author's addresses: K. Shin, B. Hooi and C. Faloutsos, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213; emails: kijungs@cs.cmu.edu, bhooi@andrew.cmu.edu, christos@cs.cmu.edu.

DOI: http://dx.doi.org/10.1145/0000000.0000000



(a) Performance Comparison

(c) Edit Wars in Korean Wikipedia

(b) Network Intrusion Detection (ROC Curve)

Users	Pages
HBC AIV helperbot3	DeltaQuad/UAA/Time
HBC AIV helperbot5	DeltaQuad/UAA/Wait
HBC AIV helperbot7	Open proxy detection
DeltaQuadBot	WikiProject Spam/LinkReports
COIBot	COIBot/LinkReports
West.andrew.g	West.andrew.g/Dead Links
RonaldB	Cyda/List of current proposed deletion
Cydebot	Cyda/List of requests for unblock

(d) Bot Activities in English Wikipedia

Fig. 1: **M-Zoom and M-Biz are fast, accurate, and effective.** (a) Our methods were 54× faster with denser subtensors than CrossSpot in Korean Wikipedia Dataset. Moreover, our methods found 24× denser subtensors than CPD. (b) Our methods identified network attacks with near-perfect accuracy (AUC=0.98). (c) Our methods spotted edit wars, during which many users (distinguished by colors) edited the same set of pages hundreds of times within several hours. (d) Our methods spotted bots, and pages edited hundreds of thousands of times by the bots.

Additional information helps identify suspicious lockstep behavior more accurately. In the above example, the fact that reviews forming a dense subgraph were also written at about the same time, with the same keywords and number of stars, makes the reviews even more suspicious. A natural and effective way to incorporate such extra information is to model data as a tensor and find dense subtensors in it [Jiang et al. 2015; Maruhashi et al. 2011].

However, neither existing methods for detecting dense subtensors nor simple extensions of graph-based methods are satisfactory in terms of speed, accuracy, and flexibility. Especially, the types of fraud detectable by each of the methods are limited since, explicitly or implicitly, each method is based on only one density metric, which decides how dense and thus suspicious each subtensor is.

Hence, in this work, we propose M-Zoom (Multidimensional Zoom) [1] and M-Biz (Multidimensional Bi-directional Zoom), which are general and flexible algorithms for detecting dense subtensors. They allow for a broad class of density metrics, in addition to having the following strengths:

- **Scalable:** Our methods are up to **114× faster** than state-of-the-art methods with similar accuracy (Figure 5) thanks to their near-linear scalability with all aspects of tensors (Figure 7).

[1] The preliminary version of M-Zoom appeared in [Shin et al. 2016b].

Table I: **M-ZOOM and M-BIZ are flexible.** Comparison between M-ZOOM, M-BIZ, and other methods for dense-subtensor detection. ✓represents 'supported'.

		FRAUDAR [Hooi et al. 2017]	Densest Subgraph [Khuller and Saha 2009]	GreedyOQC [Tsourakakis et al. 2013]	LocalOQC [Tsourakakis et al. 2013]	CROSSSPOT [Jiang et al. 2015]	MAF [Maruhashi et al. 2011]	CPD [Kolda and Bader 2009]	**M-ZOOM**	**M-BIZ**	**M-BIZ + M-ZOOM**
Data	Graph	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Tensor					✓	✓	✓	✓	✓	✓
Density Measure	Average Mass (ρ_{ari})	✓	✓	✓	✓				✓	✓	✓
	Average Mass (ρ_{geo})		✓						✓	✓	✓
	Entry Surplus			✓	✓				✓	✓	✓
	Suspiciousness					✓			✓	✓	✓
Accuracy Guarantee	Approximation Ratio	✓	✓	✓					✓		✓
	Local Optimality				✓	✓				✓	✓
Features	Multiple Subtensors					✓	✓	✓	✓	✓	✓
	Size Bounds		✓						✓	✓	✓

- **Provably accurate:** Our methods provide a guarantee on the lowest density and local optimality of subtensors they find (Theorems 3 and 5). In addition, they show high accuracy similar to state-of-the-art methods, in real-world datasets (Figure 1(a)).
- **Flexible:** Our methods work successfully with high-order tensors and support various density measures, multi-subtensor detection, and size bounds (Table I).
- **Effective:** Our methods successfully detected edit wars and bot activities in Wikipedia (Figures 1(c) and 1(d)), and detected network attacks with near-perfect accuracy (**AUC=0.98**) from a TCP dump (Figure 1(b)).

Reproducibility: The source code and data used in the paper are available at **http://www.cs.cmu.edu/~kijungs/codes/mzoom/**.

In Section 2, we present preliminaries and problem definitions. Our proposed methods, M-ZOOM and M-BIZ, are described in Section 3 and Section 4. In Section 5, we present experimental results. After discussing related work in Section 6, we draw conclusions in Section 7. In Appendix, we give detailed proofs and additional experiments.

2. NOTATIONS AND DEFINITIONS

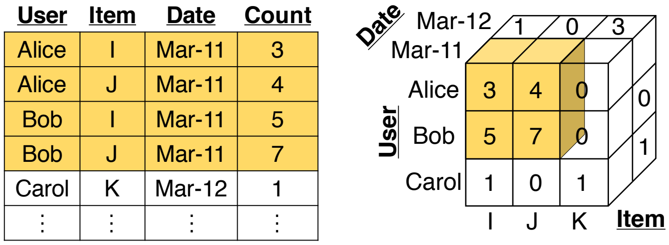
In this section, we introduce definitions and notations used in the paper. We also discuss density measures and give the formal definitions of our problems.

2.1. Definitions and Notations

Table II lists the symbols frequently used in the paper. We use $[x] = \{1, 2..., x\}$ for brevity. Let $\mathfrak{R}(A_1, A_2, ..., A_N, X)$ be a relation with N dimension attributes A_1, A_2, ..., A_N, and a nonnegative measure attribute X (see Example 1 for a running example and its pictorial description). For each tuple t in $\mathfrak{R}$ and for each dimension $n \in [N]$, the value of A_n in t is denoted by $t[A_n]$, and the value of X in t is denoted by $t[X]$. For each dimension $n \in [N]$, we use $\mathfrak{R}_n = \{t[A_n] : t \in \mathfrak{R}\}$ to denote the set of distinct values of A_n in $\mathfrak{R}$. The relation $\mathfrak{R}$ can be represented as an N-way tensor. In the tensor, each n-th

Table II: Table of symbols.

Symbol	Definition
$\mathfrak{R}(A_1, A_2, ..., A_N, X)$	a relation with N dimension attributes and a measure attribute
N	number of dimension attributes in a relation
A_n	n-th dimension attribute in $\mathfrak{R}$
X	measure attribute in $\mathfrak{R}$
$t[A_n]$ (or $t[X]$)	value of attribute A_n (or X) in tuple t
$\mathfrak{B}$	a subtensor in $\mathfrak{R}$
$\rho(\mathfrak{B}, \mathfrak{R})$	density of subtensor $\mathfrak{B}$ in $\mathfrak{R}$
$\mathfrak{R}_n$ (or $\mathfrak{B}_n$)	set of distinct values of A_n in $\mathfrak{R}$ (or $\mathfrak{B}$)
$\mathfrak{R}(a, n)$	set of tuples with attribute $A_n = a$ in $\mathfrak{R}$
$M_{\mathfrak{R}}$ (or $mass(\mathfrak{R})$)	mass of $\mathfrak{R}$
$S_{\mathfrak{R}}$ (or $size(\mathfrak{R})$)	size of $\mathfrak{R}$
$V_{\mathfrak{R}}$ (or $volume(\mathfrak{R})$)	volume of $\mathfrak{R}$
k	number of subtensors we aim to find
$\partial_{\mathfrak{B},\mathfrak{R}}$	boundary of subtensor $\mathfrak{B}$ in $\mathfrak{R}$ (defined in Section 4.1)
$\partial_{\mathfrak{B},\mathfrak{R}}(a, n)$	set of tuples with attribute $A_n = a$ in $\partial_{\mathfrak{B},\mathfrak{R}}$ (defined in Section 4.1)
$[x]$	$\{1, 2..., x\}$



User	Item	Date	Count
Alice	I	Mar-11	3
Alice	J	Mar-11	4
Bob	I	Mar-11	5
Bob	J	Mar-11	7
Carol	K	Mar-12	1
⋮	⋮	⋮	⋮

(a) Relation $\mathfrak{R}$ (b) Tensor Representation of $\mathfrak{R}$

Fig. 2: Pictorial description of Example 1. (a) Relation $\mathfrak{R}$ (*Purchase*) where the colored tuples compose subtensor $\mathfrak{B}$. (b) Tensor representation of $\mathfrak{R}$ where $\mathfrak{B}$ forms a subtensor.

mode has length $|\mathfrak{R}_n|$, and each cell has the value of attribute X, if the corresponding tuple exists, and 0 otherwise. Let $\mathfrak{B}_n$ be a subset of $\mathfrak{R}_n$. Then, we define a *subtensor* $\mathfrak{B}(A_1, A_2, ..., A_N, X) = \{t \in \mathfrak{R} : \forall n \in [N], t[A_n] \in \mathfrak{B}_n\}$, i.e., the set of tuples where each dimension attribute A_n has a value in $\mathfrak{B}_n$. $\mathfrak{B}$ is called 'subtensor' because it forms a subtensor where each n-th mode has length $|\mathfrak{B}_n|$ in the tensor representation of $\mathfrak{R}$. For each relation $\mathfrak{R}$, the set of tuples in $\mathfrak{R}$ with attribute $A_n = a$ is denoted by $\mathfrak{R}(a, n) = \{t \in \mathfrak{R} : t[A_n] = a\}$. We also define the mass of $\mathfrak{R}$ as $M_{\mathfrak{R}} = mass(\mathfrak{R}) = \sum_{t \in \mathfrak{R}} t[X]$ (i.e., the sum of the values of attribute X in $\mathfrak{R}$), the size of $\mathfrak{R}$ as $S_{\mathfrak{R}} = size(\mathfrak{R}) = \sum_{n=1}^{N} |\mathfrak{R}_n|$, and the volume of $\mathfrak{R}$ as $V_{\mathfrak{R}} = volume(\mathfrak{R}) = \prod_{n=1}^{N} |\mathfrak{R}_n|$.

Example 1 (Purchase History). *Let $\mathfrak{R} = Purchase(user, item, date, count)$ depicted in Figure 2a. Each tuple (u, i, d, c) in $\mathfrak{R}$ indicates that user u purchased c units of item i on date d. The first three attributes $A_1 = user$, $A_2 = item$, and $A_3 = date$ are dimension attributes, and the other one $X = count$ is the measure attribute. Let $\mathfrak{B}_1 = \{$'Alice', 'Bob'$\}$, $\mathfrak{B}_2 = \{$'I','J'$\}$, and $\mathfrak{B}_3 = \{$Mar-11$\}$. Then, $\mathfrak{B}$ is the set of tuples regarding the purchases by 'Alice' or 'Bob' on 'I' or 'J' on Mar-11, and its mass $M_{\mathfrak{B}} = 19$, which is the total units sold by such purchases. Likewise, $M_{\mathfrak{B}(user, \text{'Alice'})} = mass(\mathfrak{B}(user, \text{'Alice'})) = 7$,* *which is the total units of 'I' or 'J' purchased exactly by 'Alice' on Mar-11. In the tensor representation, $\mathcal{B}$ composes a subtensor in $\mathcal{R}$, as depicted in Figure 2b.*

2.2. Density Measures

In this paper, we consider four specific density measures although our methods are not restricted to them. Among them, *arithmetic average mass* (Definition 1) and *geometric average mass* (Definition 2) are multi-dimensional extensions of average degree measures, which have been widely used for dense-subgraph detection. The merits of each average degree measure are discussed in [Charikar 2000; Kannan and Vinay 1999], and extensive research based on them is discussed in Section 6.

Definition 1 (Arithmetic Average Mass [Charikar 2000])**.** *The arithmetic average mass of a subtensor $\mathcal{B}$ of a relation $\mathcal{R}$ is defined as $\rho_{ari}(\mathcal{B},\mathcal{R}) := M_{\mathcal{B}}/(S_{\mathcal{B}}/N)$.*

Definition 2 (Geometric Average Mass [Charikar 2000])**.** *The geometric average mass of a subtensor $\mathcal{B}$ of a relation $\mathcal{R}$ is defined as $\rho_{geo}(\mathcal{B},\mathcal{R}) := M_{\mathcal{B}}/V_{\mathcal{B}}^{(1/N)}$.*

Another density measure is *entry surplus* (Definition 3), defined as the observed mass of $\mathcal{B}$ subtracted by α times the mass expected under the assumption that the value of each cell (in the tensor representation) in $\mathcal{R}$ is i.i.d. Entry surplus is a multi-dimensional extension of *edge surplus*, defined in graphs [Tsourakakis et al. 2013]. Subtensors with high entry surplus are configurable by adjusting α. With high α values, relatively small compact subtensors have higher entry surplus than large sparse subtensors, while the opposite happens with small α values. We show this trend experimentally in Appendix C.2.

Definition 3 (Entry Surplus [Tsourakakis et al. 2013])**.** *The entry surplus of a subtensor $\mathcal{B}$ of a relation $\mathcal{R}$ is defined as $\rho_{es(\alpha)}(\mathcal{B},\mathcal{R}) := M_{\mathcal{B}} - \alpha M_{\mathcal{R}}(V_{\mathcal{B}}/V_{\mathcal{R}})$, where α is a constant.*

The other density measure is *suspiciousness* (Definition 4), defined as the negative log likelihood that a subtensor with the same volume of $\mathcal{B}$ has mass $M_{\mathcal{B}}$ under the assumption that the value on each cell (in the tensor representation) of $\mathcal{R}$ is i.i.d. from a Poisson Distribution. This proved useful in fraud detection [Jiang et al. 2015].

Definition 4 (Suspiciousness [Jiang et al. 2015])**.** *The suspiciousness of a subtensor $\mathcal{B}$ of a relation $\mathcal{R}$ is defined as $\rho_{susp}(\mathcal{B},\mathcal{R}) := M_{\mathcal{B}}(\log(M_{\mathcal{B}}/M_{\mathcal{R}}) - 1) + M_{\mathcal{R}}V_{\mathcal{B}}/V_{\mathcal{R}} - M_{\mathcal{B}}\log(V_{\mathcal{B}}/V_{\mathcal{R}})$.*

Our methods, however, are not restricted to the four measures mentioned above. Our methods, which search for dense subtensors, allow for any density measure ρ that satisfies Axiom 1, which any reasonable density measure should satisfy.

Axiom 1 (Density Axiom)**.** *If two subtensors of a relation have the same cardinality for every dimension attribute, the subtensor with higher or equal mass is at least as dense as the other one. Formally,*

$$|\mathcal{B}_n| = |\mathcal{B}'_n|, \forall n \in [N] \textit{ and } M_{\mathcal{B}} \geq M_{\mathcal{B}'} \Rightarrow \rho(\mathcal{B},\mathcal{R}) \geq \rho(\mathcal{B}',\mathcal{R}).$$

2.3. Problem Definition

We formally define the problem of detecting the k densest subtensors in a tensor.

Problem 1 (Top-k Densest Subtensors)**. (1) Given:** *a relation $\mathcal{R}$, the number of subtensors k, and a density measure ρ,* **(2) Find:** *k distinct subtensors of $\mathcal{R}$ with the highest densities in terms of ρ.*

We also consider a variant of Problem 1 that incorporates lower and upper bounds on the size of the detected subtensors. This is particularly useful if the unrestricted densest subtensors are not meaningful due to being too small (e.g. a single tuple) or too large (e.g. the entire tensor). For example, consider dense subtensors in a TCP dump, which indicate suspicious TCP connections. If the unrestricted densest subtensors are too large, it is costly for security experts who monitor the network to investigate all the suspicious connections in the large subtensors. In such a case, upper bounding the size of dense subtensors can direct the attention of the experts to a smaller number of connections that are more suspicious. On the other hand, if the unrestricted densest subtensors are small, after investigating the connections in the small subtensors, experts can further investigate larger but sparser subtensors obtained by lower bounding the size of dense subtensors.

Problem 2 (Top-k Densest Subtensors with Size Bounds)**. (1) Given:** *a relation $\mathcal{R}$, the number of subtensors k, a density measure ρ, a lower size bound S_{min}, and an upper size bound S_{max},* **(2) Find:** *k distinct subtensors of $\mathcal{R}$ with the highest densities in terms of ρ* **(3) Among:** *subtensors whose sizes are at least S_{min} and at most S_{max}.*

Even when we restrict our attention to a special case (N=2, k=1, ρ=ρ_{ari}, S_{min}=S_{max}), exactly solving Problems 1 and 2 takes $O(S_{\mathcal{R}}^6)$ time [Goldberg 1984] and is NP-hard [Andersen and Chellapilla 2009], resp., which are infeasible for large datasets. Thus, in this work, we focus on approximate algorithms that (a) have near-linear scalability with all aspects of $\mathcal{R}$, (b) provide accuracy guarantees at least for some density measures, and (c) produce meaningful results in real-world tensors, as explained in detail in Sections 3 and 4.

3. PROPOSED METHOD: M-ZOOM (MULTIDIMENSIONAL ZOOM)

In this section, we propose M-ZOOM (Multidimensional Zoom), a fast, accurate, and flexible method for finding dense subtensors. We present the details of M-ZOOM in Section 3.1 and discuss its efficient implementation in Section 3.2. Then, we analyze its time complexity, space complexity, and accuracy guarantees in Section 3.3.

3.1. Algorithm

Algorithm 1 describes the outline of M-ZOOM. M-ZOOM first copies the given relation $\mathcal{R}$ and assigns it to $\mathcal{R}^{ori}$ (line 1). Then, M-ZOOM finds k dense subtensors one by one from $\mathcal{R}$ (line 4). After finding each subtensor from $\mathcal{R}$, M-ZOOM removes the tuples in the subtensor from $\mathcal{R}$ so that the same subtensor is not found repeatedly (line 5). Due to these changes in $\mathcal{R}$, a subtensor found in $\mathcal{R}$ is not necessarily a subtensor of the original relation $\mathcal{R}^{ori}$. Thus, instead of returning the subtensors found in $\mathcal{R}$, M-ZOOM returns the subtensors of $\mathcal{R}^{ori}$ consisting of the same attribute values with the found subtensors (lines 6-7). This also enables M-ZOOM to find overlapped subtensors, i.e., a tuple can be included in two or more subtensors.

Algorithm 2 and Figure 3 describe how M-ZOOM finds a single dense subtensor from the given relation $\mathcal{R}$. The subtensor $\mathcal{B}$ is initialized to $\mathcal{R}$ (lines 1-2). From $\mathcal{B}$, M-ZOOM removes attribute values one by one in a greedy way until no attribute value is left (line 4). Specifically, M-ZOOM finds a dimension $n^- \in [N]$ and a value $a^- \in \mathcal{B}_n$ which are $n \in [N]$ and $a \in \mathcal{B}_n$ maximizing $\rho(\mathcal{B} - \mathcal{B}(a,n), \mathcal{R})$ (i.e., density when tuples with $A_n = a$ are removed from $\mathcal{B}$) (line 7). Then, the attribute value a^- and the tuples with $A_{n^-} = a^-$ are removed from $\mathcal{B}_{n^-}$ and $\mathcal{B}$, respectively (lines 8-9). Before removing each attribute value, M-ZOOM adds the current $\mathcal{B}$ to the snapshot list if $\mathcal{B}$ satisfies the size bounds (i.e., $S_{min} \leq S_{\mathcal{B}} \leq S_{max}$) (lines 5-6). The final step of finding a subtensor is to return the subtensor densest among those in the snapshot list (line 10).

```
Algorithm 1: Common Outline of M-ZOOM and M-BIZ
Input  : relation: $\mathcal{R}$, number of subtensors: $k$, density measure: $\rho$,
         lower size bound: $S_{min}$, upper size bound: $S_{max}$
Output: $k$ dense subtensors
$\mathcal{R}^{ori} \leftarrow copy(\mathcal{R})$
$results \leftarrow \emptyset$
for $i \leftarrow 1..k$ do
    $\mathcal{B} \leftarrow find\_single\_subtensor(\mathcal{R}, \rho, S_{min}, S_{max})$        ▷ see Algorithm 2
    $\mathcal{R} \leftarrow \mathcal{R} - \mathcal{B}$
    $\mathcal{B}^{ori} \leftarrow \{t \in \mathcal{R}^{ori} : \forall n \in [N], t[A_n] \in \mathcal{B}_n\}$
    $results \leftarrow results \cup \{\mathcal{B}^{ori}\}$
return $results$
```

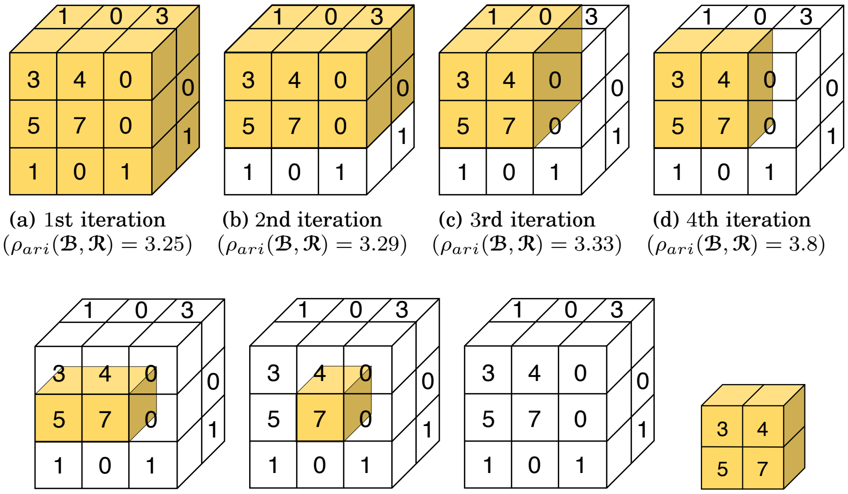
```
Algorithm 2: find_single_subtensor in M-ZOOM
Input  : relation: $\mathcal{R}$, density measure: $\rho$, lower size bound: $S_{min}$, upper size bound: $S_{max}$
Output: a dense subtensor
1  $\mathcal{B} \leftarrow copy(\mathcal{R})$
2  $\mathcal{B}_n \leftarrow copy(\mathcal{R}_n), \forall n \in [N]$
3  $snapshots \leftarrow \emptyset$
4  while $\exists n \in [N]$ s.t. $\mathcal{B}_n \neq \emptyset$ do
5      if $\mathcal{B}$ is in size bounds (i.e., $S_{min} \leq S_{\mathcal{B}} \leq S_{max}$) then
6          $snapshots \leftarrow snapshots \cup \{\mathcal{B}\}$
7      $(n^-, a^-) \leftarrow n \in [N]$ and $a \in \mathcal{B}_n$ maximizing $\rho(\mathcal{B} - \mathcal{B}(a, n), \mathcal{R})$        ▷ see Algorithm 3
8      $\mathcal{B} \leftarrow \mathcal{B} - \mathcal{B}(a^-, n^-)$
9      $\mathcal{B}_{n^-} \leftarrow \mathcal{B}_{n^-} - \{a^-\}$
10 return $\mathcal{B} \in snapshots$ with maximum $\rho(\mathcal{B}, \mathcal{R})$
```

```
Algorithm 3: Greedy Selection Using Min-Heaps in M-ZOOM
Input  : current subtensor: $\mathcal{B}$, density measure: $\rho$, min-heaps: $\{H_n^{min}\}_{n=1}^{N}$
Output: a dimension and an attribute value to be removed
for each dimension $n \in [N]$ do
    $a_n^- \leftarrow$ attribute value with minimum key in $H_n^{min}$        ▷ key= $M_{\mathcal{B}(a,n)}$
$n^- \leftarrow n \in [N]$ maximizing $\rho(\mathcal{B} - \mathcal{B}(a_n^-, n), \mathcal{R})$
$a^- \leftarrow a_{n^-}^-$
delete $a^-$ from $H_{n^-}^{min}$
for each tuple $t \in \mathcal{B}(a^-, n^-)$ do
    for each dimension $n \in [N] - \{n^-\}$ do
        decrease the key of $t[A_n]$ in $H_n^{min}$ by $t[X]$        ▷ key= $M_{\mathcal{B}(t[A_n],n)}$
return $(n^-, a^-)$
```

3.2. Efficient Implementation of M-ZOOM

In this section, we discuss an efficient implementation of M-ZOOM focusing on greedy attribute-value selection and densest-subtensor selection.

Attribute-Value Selection Using Min-Heaps. Finding a dimension $n \in [N]$ and a value $a \in \mathcal{B}_n$ that maximize $\rho(\mathcal{B} - \mathcal{B}(a, n), \mathcal{R})$ (line 7 of Algorithm 2) can be computationally expensive if all possible attribute values (i.e., $\{(n, a) : n \in [N], a \in \mathcal{B}_n\}$) should



(a) 1st iteration ($\rho_{ari}(\mathcal{B},\mathcal{R}) = 3.25$)
(b) 2nd iteration ($\rho_{ari}(\mathcal{B},\mathcal{R}) = 3.29$)
(c) 3rd iteration ($\rho_{ari}(\mathcal{B},\mathcal{R}) = 3.33$)
(d) 4th iteration ($\rho_{ari}(\mathcal{B},\mathcal{R}) = 3.8$)
(e) 5th iteration ($\rho_{ari}(\mathcal{B},\mathcal{R}) = 3$)
(f) 6th iteration ($\rho_{ari}(\mathcal{B},\mathcal{R}) = 2.33$)
(g) 7-9th iterations ($\rho_{ari}(\mathcal{B},\mathcal{R}) = 0$)
(h) Output of Algorithm 2

Fig. 3: Pictorial description of Algorithm 2 with tensor $\mathcal{R}$ in Example 1. All the invisible entries of $\mathcal{R}$ are zeros. We assume $\rho = \rho_{ari}$, $S_{min} = 0$, and $S_{max} = S_{\mathcal{R}}$. The colored region in each of (a)-(g) shows subtensor $\mathcal{B}$ added to *snapshots* in each iteration. Note that in each iteration, an attribute value (i.e., a slice in the tensor presentation) is removed from $\mathcal{B}$ so that $\rho_{ari}(\mathcal{B},\mathcal{R})$ is maximized. (h) shows the output of Algorithm 2, the subtensor with maximum $\rho_{ari}(\mathcal{B},\mathcal{R})$ among those in *snapshots*.

be considered. However, due to Axiom 1, which is assumed to be satisfied by considered density measures, the number of candidates is reduced to N if $M_{\mathcal{B}(a,n)}$ is known for each dimension $n \in [N]$ and each attribute value $a \in \mathcal{B}_n$. Lemma 1 formalizes this.

Lemma 1. *If we remove a value of attribute A_n from $\mathcal{B}_n$, removing $a \in \mathcal{B}_n$ with minimum $M_{\mathcal{B}(a,n)}$ results in the highest density. Formally, for each $n \in [N]$,*

$$M_{\mathcal{B}(a',n)} \leq M_{\mathcal{B}(a,n)}, \forall a \in \mathcal{B}_n \Rightarrow \rho(\mathcal{B} - \mathcal{B}(a',n), \mathcal{R}) \geq \rho(\mathcal{B} - \mathcal{B}(a,n), \mathcal{R}), \forall a \in \mathcal{B}_n.$$

Proof. Let $\mathcal{B}' = \mathcal{B} - \mathcal{B}(a',n)$ and $\mathcal{B}'' = \mathcal{B} - \mathcal{B}(a,n)$. Then, $|\mathcal{B}'_n| = |\mathcal{B}''_n|, \forall n \in [N]$. In addition, $M_{\mathcal{B}'} \geq M_{\mathcal{B}''}$ since $M_{\mathcal{B}'} = M_{\mathcal{B}} - M_{\mathcal{B}(a',n)} \geq M_{\mathcal{B}} - M_{\mathcal{B}(a,n)} = M_{\mathcal{B}''}$. Hence, by Axiom 1, $\rho(\mathcal{B} - \mathcal{B}(a',n), \mathcal{R}) \geq \rho(\mathcal{B} - \mathcal{B}(a,n), \mathcal{R})$. ■

By Lemma 1, if we let a_n^- be $a \in \mathcal{B}_n$ with minimum $M_{\mathcal{B}(a,n)}$, we only have to consider the dimension and value pairs in $\{(n, a_n^-) : n \in [N]\}$ instead of $\{(n, a) : n \in [N], a \in \mathcal{B}_n\}$ to find the attribute value maximizing density when it is removed. To exploit this, our implementation of M-ZOOM maintains a min-heap for each attribute A_n where the key of each value $a \in \mathcal{B}_n$ is $M_{\mathcal{B}(a,n)}$. This key is updated, which takes $O(1)$ if Fibonacci Heaps are used as min-heaps, whenever the tuples with the corresponding attribute value are removed. Algorithm 3 describes in detail how to find the attribute value to be removed based on these min-heaps, and how to update keys in them. Since Algorithm 3 considers all promising dimension and value pairs (i.e., $\{(n, a_n^-)\}_{n=1}^{N}$), it guarantees to find the value that maximizes density when it is removed.

Densest-Subtensor Selection Using Attribute-Value Ordering. As explained in Section 3.1, M-ZOOM returns the subtensor with maximum density among the snapshots of $\mathcal{B}$ (line 10 of Algorithm 2). Explicitly maintaining the list of snapshots, whose length is at most $S_{\mathcal{R}}$, requires $O(N|\mathcal{R}|S_{\mathcal{R}})$ computation and space for copying them. Even maintaining only the current best (i.e., the one with the highest density so far) leads to high computational cost if the current best keeps changing. Instead, our implementation maintains the order by which attribute values are removed as well as the iteration where the density was maximized, which requires only $O(S_{\mathcal{R}})$ space. From these and the original relation $\mathcal{R}$, our implementation restores the snapshot with maximum density in $O(N|\mathcal{R}| + S_{\mathcal{R}})$ time and returns it.

3.3. Complexity and Accuracy Analyses

The time and space complexities of M-ZOOM depend on the density measure used. We assume that one of the density measures in Section 2.2, which satisfy Axiom 1, is used.

Theorem 1 (Time Complexity of M-ZOOM)**.** *Let* $L = \max_{n\in[N]} |\mathcal{R}_n|$. *Then, if* $N = O(\log L)$*, the time complexity of Algorithm 1 is* $O(kN|\mathcal{R}|\log L)$.

Proof. See Appendix B.1. ■

As stated in Theorem 1, M-ZOOM scales linearly or sub-linearly with all aspects of relation $\mathcal{R}$ as well as k, the number of subtensors we aim to find. This result is also experimentally supported in Section 5.4.

Theorem 2 (Space Complexity of M-ZOOM)**.** *The space complexity of Algorithm 1 is* $O(kN|\mathcal{R}|)$.

Proof. See Appendix B.2. ■

M-ZOOM requires up to $kN|\mathcal{R}|$ space for storing k detected subtensors, as stated in Theorem 2. However, since detected subtensors are usually far smaller than $\mathcal{R}$, as seen in Tables V , IV and VI in Section 5, actual space usage can be much less than $kN|\mathcal{R}|$.

We show lower bounds on the densities of the subtensors found by M-ZOOM under the assumption that ρ_{ari} (Definition 1) is used as the density measure. Specifically, we show that Algorithm 2 without size bounds is guaranteed to find a subtensor with density at least $1/N$ of the density of the densest subtensor in the given relation (Theorem 3). This means that each n-th subtensor returned by Algorithm 1 has density at least $1/N$ of the density of the densest subtensor in $\mathcal{R} - \bigcup_{i=1}^{n-1}(i\text{-th subtensor})$.

Let $\mathcal{B}^{(r)}$ be the relation $\mathcal{B}$ at the beginning of the r-th iteration of Algorithm 2 with ρ_{ari} as the density measure, and $n^{(r)}$ and $a^{(r)}$ be n^- and a^- in the same iteration. That is, in the r-th iteration, value $a^{(r)} \in \mathcal{B}^{(r)}_{n^{(r)}}$ is removed from attribute $A_{n^{(r)}}$.

Lemma 2. *If a subtensor* $\mathcal{B}'$ *satisfying* $M_{\mathcal{B}'(a,n)} \geq c$ *for every* $n \in [N]$ *and every* $a \in \mathcal{B}'_n$ *exists in* $\mathcal{R}$*, there exists* r *satisfying* $M_{\mathcal{B}^{(r)}(a,n)} \geq c$ *for every* $n \in [N]$ *and* $a \in \mathcal{B}^{(r)}_n$.

Proof. See Appendix B.3. ■

Theorem 3 ($1/N$-Approximation Guarantee for Problem 1)**.** *Given a relation* $\mathcal{R}$*, let* $\mathcal{B}^*$ *be the subtensor* $\mathcal{B} \subset \mathcal{R}$ *with maximum* $\rho_{ari}(\mathcal{B}, \mathcal{R})$. *Let* $\mathcal{B}'$ *be the subtensor obtained by Algorithm 2 without size bounds (i.e.,* $S_{min} = 0$ *and* $S_{max} = S_{\mathcal{R}}$*). Then,* $\rho_{ari}(\mathcal{B}', \mathcal{R}) \geq \rho_{ari}(\mathcal{B}^*, \mathcal{R})/N$.

Proof. From the maximality of $\mathcal{B}^*$, $\forall n \in [N]$, $\forall a \in \mathcal{B}^*_n$, $M_{\mathcal{B}^*(a,n)} \geq M_{\mathcal{B}^*}/S_{\mathcal{B}^*}$ holds. Otherwise, a contradiction would result since for n and a with $M_{\mathcal{B}^*(a,n)} < M_{\mathcal{B}^*}/S_{\mathcal{B}^*}$,

$$\rho_{ari}(\mathcal{B}^* - \mathcal{B}^*(a,n), \mathcal{R}) = \frac{M_{\mathcal{B}^*} - M_{\mathcal{B}^*(a,n)}}{(S_{\mathcal{B}^*} - 1)/N} > \frac{M_{\mathcal{B}^*} - M_{\mathcal{B}^*}/S_{\mathcal{B}^*}}{(S_{\mathcal{B}^*} - 1)/N} = \rho_{ari}(\mathcal{B}^*, \mathcal{R}).$$

Consider $\mathcal{B}^{(r)}$ where $\forall n \in [N]$, $\forall a \in \mathcal{B}_n^{(r)}$, $M_{\mathcal{B}^{(r)}(a,n)} \geq M_{\mathcal{B}^*}/S_{\mathcal{B}^*}$. Such $\mathcal{B}^{(r)}$ exists by Lemma 2. Then, $M_{\mathcal{B}^{(r)}} \geq (S_{\mathcal{B}^{(r)}}/N)\ (M_{\mathcal{B}^*}/S_{\mathcal{B}^*}) = (S_{\mathcal{B}^{(r)}}/N)(\rho_{ari}(\mathcal{B}^*, \mathcal{R})/N)$ holds. Hence, $\rho_{ari}(\mathcal{B}', \mathcal{R}) \geq \rho_{ari}(\mathcal{B}^{(r)}, \mathcal{R}) = M_{\mathcal{B}^{(r)}}/(S_{\mathcal{B}^{(r)}}/N) \geq \rho_{ari}(\mathcal{B}^*, \mathcal{R})/N$ holds. ■

Theorem 3 can be extended to cases where only a lower bound exists. In these cases, the approximate factor is $1/(N+1)$, as stated in Theorem 4.

Theorem 4 ($1/(N+1)$-Approximation Guarantee for Problem 2)**.** *Given a relation $\mathcal{R}$, let $\mathcal{B}^*$ be the subtensor $\mathcal{B} \subset \mathcal{R}$ with maximum $\rho_{ari}(\mathcal{B}, \mathcal{R})$ among the subtensors with size at least S_{min}. Let $\mathcal{B}'$ be the subtensor obtained by Algorithm 2 with the lower size bound S_{min} (i.e., $1 \leq S_{min} \leq S_{\mathcal{R}}$ and $S_{max} = S_{\mathcal{R}}$). Then, $\rho_{ari}(\mathcal{B}', \mathcal{R}) \geq \rho_{ari}(\mathcal{B}^*, \mathcal{R})/(N+1)$.*

Proof. See Appendix B.4. ■

4. PROPOSED METHOD: M-BIZ (MULTIDIMENSIONAL BI-DIRECTIONAL ZOOM)

In this section, we propose M-BIZ (Multidimensional Bi-directional Zoom), a dense-subtensor detection algorithm complementary to and combinable with M-ZOOM, described in the previous section. M-BIZ, which starts from a seed subtensor, adds or removes attribute values and the corresponding tuples in a greedy way until the subtensor reaches a local optimum, which is defined in Section 4.1. We present the details of M-BIZ in Section 4.2 and its efficient implementation in Section 4.3. Then, we analyze the time complexity and accuracy guarantees of M-BIZ in Section 4.4.

4.1. Local Optimality

We give the definition of *local optimality*, which is a property satisfied by the subtensors found by M-BIZ. For this purpose, we first define the *boundary* of a subtensor $\mathcal{B}$ in a tensor $\mathcal{R}$ as the set of tuples that do not belong to $\mathcal{B}$ but have $N-1$ attribute values composing $\mathcal{B}$. That is, for each tuple t on the boundary, there exists a unique dimension $n \in N$ such that $t[A_n] \notin \mathcal{B}_n$, but for every $i \neq n$, $t[A_i] \in \mathcal{B}_i$ holds. Definition 5 gives a formal definition of the boundary of $\mathcal{B}$ in $\mathcal{R}$, which is denoted by $\partial_{\mathcal{B},\mathcal{R}}$.

Definition 5 (Boundary of a Subtensor)**.** *The boundary of a subtensor $\mathcal{B}$ in a relation $\mathcal{R}$ is defined as*

$$\partial_{\mathcal{B},\mathcal{R}} := \{t \in \mathcal{R} - \mathcal{B} :\ \exists n \in [N]\ s.t.\ (t[A_n] \notin \mathcal{B}_n \textit{ and } \forall i \in [N] - \{n\}, t[A_i] \in \mathcal{B}_i)\}.$$

For boundary $\partial_{\mathcal{B},\mathcal{R}}$, we use $\partial_{\mathcal{B},\mathcal{R}}(a,n) = \{t \in \partial_{\mathcal{B},\mathcal{R}} : t[A_n] = a\}$ to denote the set of tuples in $\partial_{\mathcal{B},\mathcal{R}}$ that have value a for attribute A_n. Then, for each attribute value $a \in \mathcal{R}_n - \mathcal{B}_n$, $\partial_{\mathcal{B},\mathcal{R}}(a,n)$ is the set of tuples that are included in $\mathcal{B}$ if a is added to $\mathcal{B}_n$. Based on this concept, Definition 6 gives a formal definition of local optimality for the case without size bounds (i.e., $S_{min} = 0$ and $S_{max} = S_{\mathcal{R}}$). A subtensor $\mathcal{B}$ is locally optimal if adding or removing an attribute value does not increase the density of $\mathcal{B}$.

Definition 6 (Local Optimality for Problem 1)**.** *Given a relation $\mathcal{R}$, a subtensor $\mathcal{B} \subset \mathcal{R}$, and a density measure ρ, $\mathcal{B}$ is a local optimum in $\mathcal{R}$ in terms of ρ without size bounds if the following two conditions are met:*

(*1*) $\forall n \in [N]$, $\nexists a \in \mathcal{R}_n - \mathcal{B}_n$ *s.t.* $\rho(\mathcal{B} \cup \partial_{\mathcal{B},\mathcal{R}}(a,n), \mathcal{R}) > \rho(\mathcal{B}, \mathcal{R})$,
(*2*) $\forall n \in [N]$, $\nexists a \in \mathcal{B}_n$ *s.t.* $\rho(\mathcal{B} - \mathcal{B}(a,n), \mathcal{R}) > \rho(\mathcal{B}, \mathcal{R})$.

The concept of local optimality is naturally extended to cases with size bounds, as in Definition 7. The subtensor $\mathcal{B}$ satisfying the size bounds is locally optimal if adding or removing an attribute value does not increase the density of $\mathcal{B}$ or does not satisfy the size bounds.

Algorithm 4: *find_single_subtensor in* M-Biz

```
Input  : relation: $\mathcal{R}$, density measure: $\rho$, lower size bound: $S_{min}$, upper size bound: $S_{max}$
Output: a dense subtensor
1  initialize $\mathcal{B}$ and $\{\mathcal{B}_n\}_{n=1}^{N}$ so that $S_{min} \leq S_{\mathcal{B}} \leq S_{max}$ (either by Algorithm 2 or randomly)
2  $\rho' \leftarrow -\infty$
3  while $\rho(\mathcal{B}, \mathcal{R}) > \rho'$ do
4      $\rho'$, $\rho^+$, $\rho^- \leftarrow \rho(\mathcal{B}, \mathcal{R})$
5      if $S_{\mathcal{B}} \leq S_{max} - 1$ then                          ▷ $\mathcal{B}$ can grow within the size bound
6          $(n^+, a^+) \leftarrow n \in [N]$ and $a \in \mathcal{R}_n - \mathcal{B}_n$ maximizing $\rho(\mathcal{B} \cup \partial_{\mathcal{B},\mathcal{R}}(a, n), \mathcal{R})$
7          $\rho^+ \leftarrow \rho(\mathcal{B} \cup \partial_{\mathcal{B},\mathcal{R}}(a^+, n^+), \mathcal{R})$
8      if $S_{\mathcal{B}} \geq S_{min} + 1$ then                          ▷ $\mathcal{B}$ can shrink within the size bound
9          $(n^-, a^-) \leftarrow n \in [N]$ and $a \in \mathcal{B}_n$ maximizing $\rho(\mathcal{B} - \mathcal{B}(a, n), \mathcal{R})$
10         $\rho^- \leftarrow \rho(\mathcal{B} - \mathcal{B}(a^-, n^-), \mathcal{R})$
11     if $\rho^+ > \rho'$ and $\rho^+ \geq \rho^-$ then                 ▷ adding $a^+$ to $B_{n^+}$ increases $\rho(\mathcal{B}, \mathcal{R})$ most
12         $\mathcal{B}_{n^+} \leftarrow \mathcal{B}_{n^+} \cup \{a^+\}$
13         $\mathcal{B} \leftarrow \mathcal{B} \cup \partial_{\mathcal{B},\mathcal{R}}(a^+, n^+)$
14     else if $\rho^- > \rho'$ then                                ▷ removing $a^-$ from $B_{n^-}$ increases $\rho(\mathcal{B}, \mathcal{R})$ most
15         $\mathcal{B}_{n^-} \leftarrow \mathcal{B}_{n^-} - \{a^-\}$
16         $\mathcal{B} \leftarrow \mathcal{B} - \mathcal{B}(a^-, n^-)$
17 return $\mathcal{B}$
```

Definition 7 (Local Optimality for Problem 2)**.** *Given a tensor $\mathcal{R}$, a subtensor $\mathcal{B} \subset \mathcal{R}$, a density measure ρ, a lower size bound S_{min}, and an upper size bound S_{max}, $\mathcal{B}$ is a local optimum in $\mathcal{R}$ in terms of ρ within size bound $[S_{min}, S_{max}]$ if the following three conditions are met:*

(*1*) $S_{min} \leq S_{\mathcal{B}} \leq S_{max}$,
(*2*) $\forall n \in [N]$, $\nexists a \in \mathcal{R}_n - \mathcal{B}_n$ *s.t.* $(\rho(\mathcal{B} \cup \partial_{\mathcal{B},\mathcal{R}}(a, n), \mathcal{R}) > \rho(\mathcal{B}, \mathcal{R})$ *and* $S_{\mathcal{B} \cup \partial_{\mathcal{B},\mathcal{R}}(a,n)} \leq S_{max})$,
(*3*) $\forall n \in [N]$, $\nexists a \in \mathcal{B}_n$ *s.t.* $(\rho(\mathcal{B} - \mathcal{B}(a, n), \mathcal{R}) > \rho(\mathcal{B}, \mathcal{R})$ *and* $S_{\mathcal{B} - \mathcal{B}(a,n)} \geq S_{min})$.

4.2. Algorithm

Having defined local optimality, we describe M-Biz, which finds locally optimal dense subtensors. The overall structure of M-Biz is the same with that of M-Zoom and is described in Algorithm 1. However, M-Biz and M-Zoom differ significantly in the way they find each dense subtensor.

Algorithm 4 and Figure 4 describe how M-Biz finds a single dense subtensor from the given relation $\mathcal{R}$. The subtensor $\mathcal{B}$ is initialized so that it satisfies the given size bounds, either by M-Zoom (i.e., Algorithm 2) or randomly (line 1). The effects of the initialization methods are investigated in Section 5.3. Starting from this $\mathcal{B}$, M-Biz grows or shrinks $\mathcal{B}$ until it reaches a local optimum (lines 3-16). In each iteration, M-Biz finds a dimension $n^+ \in [N]$ and a value $a^+ \in \mathcal{R}_n - \mathcal{B}_n$, which are $n \in [N]$ and $a \in \mathcal{R}_n - \mathcal{B}_n$ maximizing $\rho(\mathcal{B} \cup \partial_{\mathcal{B},\mathcal{R}}(a, n), \mathcal{R})$ (i.e., density when the tuples with $A_n = a$ on the boundary are added to $\mathcal{B}$) if $\mathcal{B}$ can grow satisfying the size bounds (lines 5-7). Likewise, M-Biz finds a dimension $n^- \in [N]$ and a value $a^- \in \mathcal{B}_n$, which are $n \in [N]$ and $a \in \mathcal{B}_n$ maximizing $\rho(\mathcal{B} - \mathcal{B}(a, n), \mathcal{R})$ (i.e., density when the tuples with $A_n = a$ are removed from $\mathcal{B}$) if $\mathcal{B}$ can shrink satisfying the size bounds (lines 8-10). If growing $\mathcal{B}$ increases the density more than shrinking it, the attribute value a^+ and the tuples with $A_{n^+} = a^+$ on the boundary are added to $\mathcal{B}_{n^+}$ and $\mathcal{B}$, respectively (lines 11-13). Likewise, if shrinking $\mathcal{B}$ increases the density more than growing it, the attribute

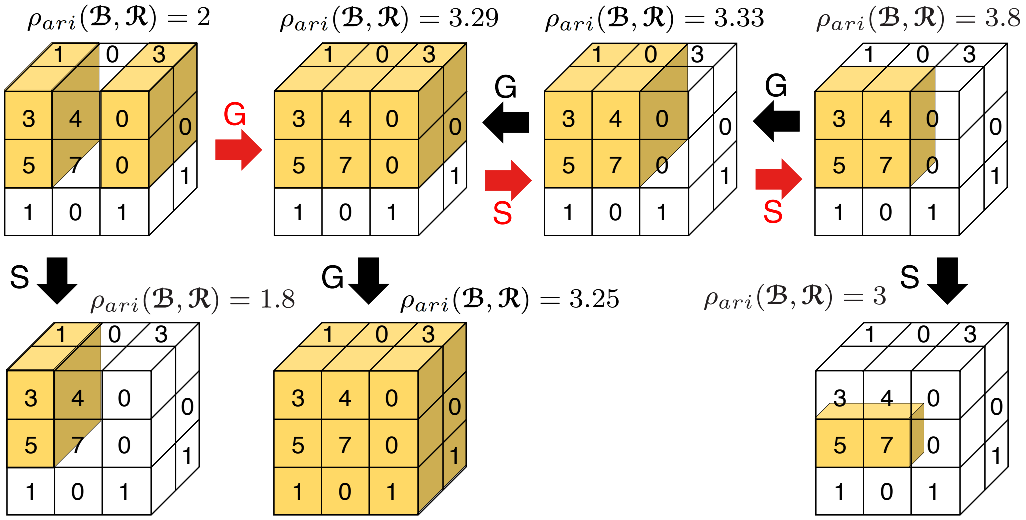

$\rho_{ari}(\mathfrak{B},\mathfrak{R}) = 2$
$\rho_{ari}(\mathfrak{B},\mathfrak{R}) = 3.29$
$\rho_{ari}(\mathfrak{B},\mathfrak{R}) = 3.33$
$\rho_{ari}(\mathfrak{B},\mathfrak{R}) = 3.8$
G
G
G
S
S
S
G
$\rho_{ari}(\mathfrak{B},\mathfrak{R}) = 1.8$
$\rho_{ari}(\mathfrak{B},\mathfrak{R}) = 3.25$
$\rho_{ari}(\mathfrak{B},\mathfrak{R}) = 3$
S
1 0 3
3 4 0
5 7 0
1 0 1
0
1

Fig. 4: Pictorial description of Algorithm 4 with tensor $\mathfrak{R}$ in Example 1. G: grow while maximizing density, S: shrink while maximizing density. All the invisible entries of $\mathfrak{R}$ are zeros. We assume $\rho = \rho_{ari}$, $S_{min} = 0$, and $S_{max} = S_{\mathfrak{R}}$. If subtensor $\mathfrak{B}$ is initialized to the colored region of the top-left tensor, it changes following the red arrows (i.e., G $\rightarrow$ S $\rightarrow$ S). Notice that, in each stage, $\mathfrak{B}$ either grows or shrinks so that $\rho_{ari}(\mathfrak{B},\mathfrak{R})$ is maximized. If $\mathfrak{B}$ reaches the colored the region of the top-right tensor, it is returned since neither growing nor shrinking increases $\rho_{ari}(\mathfrak{B},\mathfrak{R})$.

value a^- and the tuples with $A_{n^-} = a^-$ are removed from $\mathfrak{B}_{n^-}$ and $\mathfrak{B}$, respectively (lines 14-16). However, if the density of $\mathfrak{B}$ does not increase by growing or shrinking $\mathfrak{B}$, $\mathfrak{B}$ reaches a local optimum and is returned (line 17).

4.3. Efficient Implementation of M-BIZ

We discuss an efficient implementation of M-BIZ focusing on greedy attribute-value selection. Finding an attribute $n \in [N]$ and its value $a \in \mathfrak{R}_n - \mathfrak{B}_n$ that maximize $\rho(\mathfrak{B} \cup \partial_{\mathfrak{B},\mathfrak{R}}(a,n), \mathfrak{R})$ (line 6 of Algorithm 4) can be computationally expensive if all possible attribute values (i.e., $\{(n,a) : n \in [N], a \in \mathfrak{R}_n - \mathfrak{B}_n\}$) should be considered. However, due to Axiom 1, which is assumed to be satisfied by considered density measures, the number of candidates is reduced to N if $M_{\partial_{\mathfrak{B},\mathfrak{R}}(a,n)}$ is known for each dimension $n \in [N]$ and each attribute value $a \in \mathfrak{R}_n - \mathfrak{B}_n$. Lemma 3 formalizes this.

Lemma 3. *If we add a value of attribute A_n to $\mathfrak{B}_n$, adding $a \in \mathfrak{R}_n - \mathfrak{B}_n$ with maximum $M_{\partial_{\mathfrak{B},\mathfrak{R}}(a,n)}$ results in the highest density. Formally, for each $n \in [N]$,*

$$M_{\partial_{\mathfrak{B},\mathfrak{R}}(a',n)} \geq M_{\partial_{\mathfrak{B},\mathfrak{R}}(a,n)}, \forall a \in \mathfrak{R}_n - \mathfrak{B}_n$$
$$\Rightarrow \rho(\mathfrak{B} \cup \partial_{\mathfrak{B},\mathfrak{R}}(a',n), \mathfrak{R}) \geq \rho(\mathfrak{B} \cup \partial_{\mathfrak{B},\mathfrak{R}}(a,n), \mathfrak{R}), \forall a \in \mathfrak{R}_n - \mathfrak{B}_n.$$

Proof. Let $\mathfrak{B}' = \mathfrak{B} \cup \partial_{\mathfrak{B},\mathfrak{R}}(a',n)$ and $\mathfrak{B}'' = \mathfrak{B} \cup \partial_{\mathfrak{B},\mathfrak{R}}(a,n)$. Then, $|\mathfrak{B}'_n| = |\mathfrak{B}''_n|, \forall n \in [N]$. In addition, $M_{\mathfrak{B}'} \geq M_{\mathfrak{B}''}$ since $M_{\mathfrak{B}'} = M_{\mathfrak{B}} + M_{\partial_{\mathfrak{B},\mathfrak{R}}(a',n)} \geq M_{\mathfrak{B}} + M_{\partial_{\mathfrak{B},\mathfrak{R}}(a,n)} = M_{\mathfrak{B}''}$. Hence, by Axiom 1, $\rho(\mathfrak{B} \cup \partial_{\mathfrak{B},\mathfrak{R}}(a',n), \mathfrak{R}) \geq \rho(\mathfrak{B} \cup \partial_{\mathfrak{B},\mathfrak{R}}(a,n), \mathfrak{R})$. ■

By Lemma 3, if we let a_n^+ be $a \in \mathfrak{R}_n - \mathfrak{B}_n$ with maximum $M_{\partial_{\mathfrak{B},\mathfrak{R}}(a,n)}$, we only have to consider the dimension and attribute-value pairs in $\{(n, a_n^+) : n \in [N]\}$ instead of $\{(n,a) : n \in [N], a \in \mathfrak{R}_n - \mathfrak{B}_n\}$ to find the attribute value maximizing the density when

Algorithm 5: Greedy Selection Using Min-Heap and Max-Heap in M-BIZ

```
Input : relation: $\mathfrak{R}$, current subtensor: $\mathfrak{B}$, density measure: $\rho$,
        min-heaps: $\{H_n^{min}\}_{n=1}^N$, max-heaps: $\{H_n^{max}\}_{n=1}^N$
Function choose_attribute_value_to_add()
    for each dimension $n \in [N]$ do
        $a_n^+ \leftarrow a$ with maximum key in $H_n^{max}$          ▷ key= $M_{\partial_{\mathfrak{B},\mathfrak{R}}(a,n)}$
    $n^+ \leftarrow n \in [N]$ maximizing $\rho(\mathfrak{B} \cup \partial_{\mathfrak{B},\mathfrak{R}}(a_n^+, n), \mathfrak{R})$
    $a^+ \leftarrow a_{n^+}^+$
    return $(n^+, a^+)$
Function choose_attribute_value_to_delete()
    for each dimension $n \in [N]$ do
        $a_n^- \leftarrow a$ with minimum key in $H_n^{min}$          ▷ key= $M_{\mathfrak{B}(a,n)}$
    $n^- \leftarrow n \in [N]$ maximizing $\rho(\mathfrak{B} - \mathfrak{B}(a_n^-, n), \mathfrak{R})$
    $a^- \leftarrow a_{n^-}^-$
    return $(n^-, a^-)$
Function update_heaps_for_addition($n^+$, $a^+$, $\partial_{\mathfrak{B},\mathfrak{R}}^{old}(a^+, n^+)$, $\partial_{\mathfrak{B},\mathfrak{R}}^{new}(a^+, n^+)$)
    delete $a^+$ from $H_{n^+}^{max}$
    for each tuple $t \in \partial_{\mathfrak{B},\mathfrak{R}}^{old}(a^+, n^+)$ do
        for each dimension $n \in [N] - \{n^+\}$ do
            increase the key of $t[A_n]$ in $H_n^{min}$ by $t[X]$          ▷ key= $M_{\mathfrak{B}(t[A_n],n)}$
    for each tuple $t \in \partial_{\mathfrak{B},\mathfrak{R}}^{new}(a^+, n^+)$ do
        $n^* \leftarrow n \in [N] - \{n^+\}$ where $t[A_n] \notin \mathfrak{B}_n$
        increase the key of $t[A_{n^*}]$ in $H_{n^*}^{max}$ by $t[X]$          ▷ key= $M_{\partial_{\mathfrak{B},\mathfrak{R}}(t[A_{n^*}],n^*)}$
    add $a^+$ to $H_{n^+}^{min}$ with key= $M_{\partial_{\mathfrak{B},\mathfrak{R}}^{old}(a^+,n^+)}$
Function update_heaps_for_deletion($n^-$, $a^-$, $\partial_{\mathfrak{B},\mathfrak{R}}^{old}(a^-, n^-)$, $\partial_{\mathfrak{B},\mathfrak{R}}^{new}(a^-, n^-)$)
    delete $a^-$ from $H_{n^-}^{min}$
    for each tuple $t \in \partial_{\mathfrak{B},\mathfrak{R}}^{old}(a^-, n^-)$ do
        $n^* \leftarrow n \in [N] - \{n^-\}$ where $t[A_n] \notin \mathfrak{B}_n$
        decrease the key of $t[A_{n^*}]$ in $H_{n^*}^{max}$ by $t[X]$          ▷ key= $M_{\partial_{\mathfrak{B},\mathfrak{R}}(t[A_{n^*}],n^*)}$
    for each tuple $t \in \partial_{\mathfrak{B},\mathfrak{R}}^{new}(a^-, n^-)$ do
        for each dimension $n \in [N] - \{n^-\}$ do
            decrease the key of $t[A_n]$ in $H_n^{min}$ by $t[X]$          ▷ key= $M_{\mathfrak{B}(t[A_n],n)}$
    add $a^-$ to $H_{n^-}^{max}$ with key= $M_{\partial_{\mathfrak{B},\mathfrak{R}}^{new}(a^-,n^-)}$
```

it is added. Likewise, by Lemma 1, if we let a_n^- be $a \in \mathfrak{B}_n$ with minimum $M_{\mathfrak{B}(a,n)}$, we only have to consider the dimension and attribute-value pairs in $\{(n, a_n^-) : n \in [N]\}$ instead of $\{(n, a) : n \in [N], a \in \mathfrak{B}_n\}$ to find the attribute value maximizing density when it is removed. To exploit these facts, our implementation of M-BIZ maintains a max-heap and a min-heap for each attribute A_n where the key of each value $a \in \mathfrak{R}_n - \mathfrak{B}_n$ in the max-heap is $M_{\partial_{\mathfrak{B},\mathfrak{R}}(a,n)}$ and the key of each value $a \in \mathfrak{B}_n$ in the min-heap is $M_{\mathfrak{B}(a,n)}$. These keys are updated whenever the tuples with the corresponding attribute value are added to or removed from $\mathfrak{B}$ or $\partial_{\mathfrak{B},\mathfrak{R}}(a, n)$. The functions in Algorithm 5 describe in detail how to find the attribute value to be added or removed based on these max-heaps and min-heaps and how to update keys in them.

— *choose_attribute_value_to_add*: choose a dimension $n^+ \in [N]$ and an attribute value $a^+ \in \mathcal{R}_{n^+} - \mathcal{B}_{n^+}$ maximizing the density when the value is added to $\mathcal{B}_{n^+}$. Note that all promising dimension and value pairs (i.e., $\{(n, a_n^+)\}_{n=1}^N$) are considered (Lemma 3).

— *choose_attribute_value_to_delete*: choose a dimension $n^- \in [N]$ and an attribute value $a^- \in \mathcal{B}_{n^-}$ maximizing the density when the value is removed from $\mathcal{B}_{n^-}$. Note that all promising dimension and value pairs (i.e., $\{(n, a_n^-)\}_{n=1}^N$) are considered (Lemma 1).

— *update_heaps_for_addition*: update keys in the max-heaps and the min-heaps when attribute value a^+ is added to $\mathcal{B}_{n^+}$. We use $\partial^{old}_{\mathcal{B},\mathcal{R}}(a^+, n^+)$ to denote the set of tuples with $A_{n^+} = a^+$ on the boundary before the change. These tuples are excluded from $\partial_{\mathcal{B},\mathcal{R}}$ and included in $\mathcal{B}$ by the change. We use $\partial^{new}_{\mathcal{B},\mathcal{R}}(a^+, n^+)$ to denote the set of tuples with $A_{n^+} = a^+$ on the boundary after the change.

— *update_heaps_for_deletion*: update keys in the max-heaps and the min-heaps when attribute value a^- is removed from $\mathcal{B}_{n^-}$. We use $\partial^{old}_{\mathcal{B},\mathcal{R}}(a^-, n^-)$ to denote the set of tuples with $A_{n^-} = a^-$ on the boundary before the change. These tuples are excluded from $\partial_{\mathcal{B},\mathcal{R}}$ by the change. We use $\partial^{new}_{\mathcal{B},\mathcal{R}}(a^-, n^-)$ to denote the set of tuples with $A_{n^-} = a^-$ on the boundary after the change. These tuples are removed from $\mathcal{B}$ by the change.

4.4. Complexity and Accuracy Analyses

In this section, we prove the time complexity of M-BIZ and the accuracy guarantees provided by M-BIZ. Theorem 5 states that M-BIZ always converges to a local optimum.

Theorem 5 (Convergence and Local Optimality of M-BIZ). *Algorithm 4 always terminates, and the returned subtensor $\mathcal{B}$ is a local optimum in the given relation $\mathcal{R}$ in terms of the given density measure ρ. That is, $\mathcal{B}$ satisfies the conditions in Definition 6, if there is no size bound, and those in Definition 7, if there are size bounds.*

Proof. Algorithm 4 always terminates since the possible states of $\mathcal{B}$ are finite and $\rho(\mathcal{B}, \mathcal{R})$ strictly increases (otherwise the algorithm terminates) at each iteration. The returned subtensor $\mathcal{B}$ is a local optimum since its local optimality is the termination condition. ■

Although the number of updates until convergence varies depending on initialization methods and density measures, for each update, M-BIZ needs to access only the tuples with the updated attribute value, as formalized in Theorem 6. This is contrast to CROSSSPOT [Jiang et al. 2015], which may need to access all tuples for an update. Moreover, we experimentally show that the total running time of M-BIZ is linear or sub-linear in all aspects of the relation $\mathcal{R}$ in Section 5.4. Regarding its space complexity, M-BIZ has the same space complexity of M-ZOOM (see Theorem 2).

Theorem 6 (Time Complexity of M-BIZ). *Let $L = \max_{n\in[N]} |\mathcal{R}_n|$, and let $a^* \in \mathcal{R}_{n^*}$ be the attribute value added to or removed from $\mathcal{B}_{n^*}$ in an iteration of Algorithm 4. Then, the iteration takes $O(N|\mathcal{R}(a^*, n^*)| \log L)$.*

Proof. See Appendix B.5. ■

M-BIZ and M-ZOOM can be combined by using M-ZOOM as the initialization method in M-BIZ. By doing so, their combination enjoys both the approximation bounds provided by M-ZOOM and the local optimality provided by M-BIZ, as stated in Theorem 7. In addition, their combination shows higher accuracy then the individual methods in practice, as shown in Section 5.3.

Theorem 7 (Combination of M-ZOOM and M-BIZ)**.** *If Algorithm 2 is used as the initialization method, the subtensor returned by Algorithm 4 guarantees both the approximation bounds in Algorithm 2 (see Theorems 3 and 4) and the local optimality in Algorithm 4 (see Theorem 5).*

Proof. Let $\mathcal{B}'$ be the subtensor returned by Algorithm 2 and let $\mathcal{B}''$ be the subtensor returned by Algorithm 4 when $\mathcal{B}'$ is used as the initial subtensor. Since Algorithm 4 never decreases the density of the maintained subtensor, $\rho_{ari}(\mathcal{B}'', \mathcal{R}) \geq \rho_{ari}(\mathcal{B}', \mathcal{R})$ holds, and thus approximation bounds satisfied by $\mathcal{B}'$ are also satisfied by $\mathcal{B}''$. On the other hand, the local optimality of $\mathcal{B}''$ is satisfied regardless of initialization methods, as proven in Theorem 5. ■

5. EXPERIMENTS

We designed and performed experiments to answer the following questions:

- **Q1.** How *fast* and *accurately* do M-ZOOM and M-BIZ detect dense subtensors from real-world tensors?
- **Q2.** Are M-ZOOM and M-BIZ *complementary*? Is their combination more accurate than each individual method?
- **Q3.** Do M-ZOOM and M-BIZ *scale linearly* with all aspects of data?
- **Q4.** Do M-ZOOM and M-BIZ detect many *distinct* dense subtensors from real-world tensors?
- **Q5.** Which *anomalies* or *fraud* do M-ZOOM and M-BIZ spot from real-world tensors?

In addition to the above questions, we analyze the effects of α in entry surplus $\rho_{es(\alpha)}$ on sutensors detected by our algorithms in Appendix C.2.

5.1. Experimental Settings

Machines: All experiments were conducted on a machine with 2.67 GHz Intel Xeon E7-8837 CPUs and 1TB RAM.

Datasets: We used diverse real-world tensor datasets, summarized in Table III. They are grouped as follows:

- **Temporal social networks:** *Youtube(user, user, date, 1)* [Mislove et al. 2007] represents who became a friend of whom when on Youtube. *SMS(user, user, timestamp, #messages)* represents who sent text messages to whom, when, and how many times in a large Asian city.
- **Behavior logs:** *StackO.(user, post, timestamp, 1)* [Kunegis 2013] represents who marked which post as favorite when on Stack Overflow. *KoWiki(user, page, timestamp, #revisions)* and *EnWiki(user, page, timestamp, #revisions)* represent who revised which page, when, and how many times on Korean Wikipedia and English Wikipedia, respectively.
- **Ratings:** *Yelp(user, business, date, score, 1)*, *Android(user, app, date, score, 1)* [McAuley et al. 2015], *Netflix(user, movie, date, score, 1)* [Bennett and Lanning 2007], and *YahooM.(user, item, date, score, 1)* [Dror et al. 2012] represent who gave which score, when, and to which business, app, movie, and item on Yelp, Amazon, Netflix, and Yahoo Music, respectively.
- **TCP dumps:** From TCP dump data for a typical U.S. Air Force LAN, we created a relation *AirForce(protocol, service, flag, src-bytes, dst-bytes, counts, srv-counts, #connections)*. See Appendix C.1 for the description of each attribute.

Timestamps are in hours in all the datasets.

Implementations and Parameter Settings: We compared M-ZOOM and M-BIZ with CROSSSPOT [Jiang et al. 2015], CP Decomposition (CPD) [Kolda and Bader 2009]

Table III: **Summary of real-world datasets.** M: Million, K: Thousand. The underlined attributes are composite primary keys.

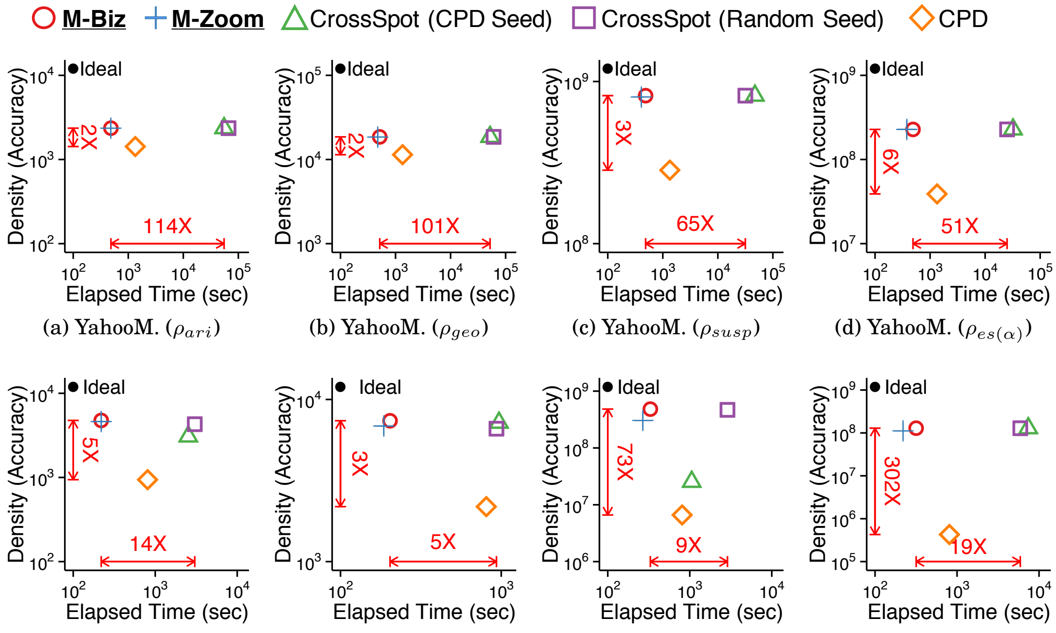
Name	Schema	Volume	#Tuples
Temporal Social Network			
Youtube	(user, user, date, 1)	3.22M × 3.22M × 203	18.7M
SMS	(user, user, timestamp, #messages)	1.25M × 7.00M ×4.39K	103M
Behavior Logs			
StackO.	(user, post, timestamp, 1)	545K × 96.7K × 1.15K	1.30M
KoWiki	(user, page, timestamp, #revisions)	470K × 1.18M × 101K	11.0M
EnWiki	(user, page, timestamp, #revisions)	44.1M × 38.5M × 129K	483M
Ratings			
Yelp	(user, business, date, score, 1)	552K × 77.1K × 3.80K × 5	2.23M
Android	(user, app, date, score, 1)	1.32M × 61.3K × 1.28K × 5	2.64M
Netflix	(user, movie, date, score, 1)	480K × 17.8K × 2.18K × 5	99.1M
YahooM.	(user, item, date, score, 1)	1.00M × 625K × 84.4K × 101	253M
TCP Dumps			
AirForce	(protocol, service, flag, src-bytes, dst-bytes, counts, srv-counts, #connections)	3 × 70 × 11 × 7.20K × 21.5K × 512 × 512	648K

(see Appendix A for details), and MultiAspectForensics (MAF) [Maruhashi et al. 2011]. Methods only applicable to graphs [Hooi et al. 2017; Prakash et al. 2010; Shin et al. 2016a] were excluded from comparison. M-Zoom, M-Biz, and CrossSpot[2] were implemented in Java, while Tensor Toolbox [Bader and Kolda 2007], which gives the state-of-the-art implementations of tensor decomposition, was used for CPD and MAF. Although CrossSpot was originally designed to maximize ρ_{susp}, it can be extended to other density measures. These variants were used depending on the density measure compared in each experiment. In addition, we used CPD as the seed-subtensor selection method of CrossSpot, which outperformed HOSVD used in [Jiang et al. 2015] in terms of both speed and accuracy. For M-Biz, we used M-Zoom as the seed-subtensor selection method unless otherwise stated. The parameter α in $\rho_{es(\alpha)}$ was set to 1 unless otherwise stated.

5.2. Q1. Running Time and Accuracy of M-Zoom and M-Biz

We compare the speed of the considered methods and the densities of the subtensors found by the methods in real-world datasets. Specifically, we measured time taken to find three subtensors and the maximum density among the three subtensors. Figure 5 shows the results with different density measures in YahooM. and SMS datasets. M-Zoom and M-Biz clearly provided the best trade-off between speed and accuracy in both datasets regardless of density measures. For example, when ρ_{ari} was used as the density measure, M-Zoom and M-Biz were 114 times faster than CrossSpot, while detecting subtensors with similar densities. Compared with CPD, our methods detected 2 times denser subtensors 2.8 times faster. Although the results are not included in Figure 5, MAF found several orders of magnitude sparser subtensors than the other methods, with speed similar to that of CPD. As seen in Figure 11 in Ap-

[2] We referred the open-sourced implementation at http://github.com/mjiang89/CrossSpot.



(a) YahooM. (ρ_{ari}) (b) YahooM. (ρ_{geo}) (c) YahooM. (ρ_{susp}) (d) YahooM. ($\rho_{es(\alpha)}$)

(e) SMS (ρ_{ari}) (f) SMS (ρ_{geo}) (g) SMS (ρ_{susp}) (h) SMS ($\rho_{es(\alpha)}$)

Fig. 5: **Only M-ZOOM and M-BIZ achieve both speed and accuracy.** In each plot, points represent the speed of different methods and the highest density of the three subtensors found by the methods. Upper-left region indicates better performance. M-ZOOM and M-BIZ give the best trade-off between speed and density regardless of used density measures. Specifically, they are up to **114× faster** than CROSSSPOT with similarly dense subtensors. Similar results are obtained also in other datasets, as seen in Figure 11 in Appendix C.4.

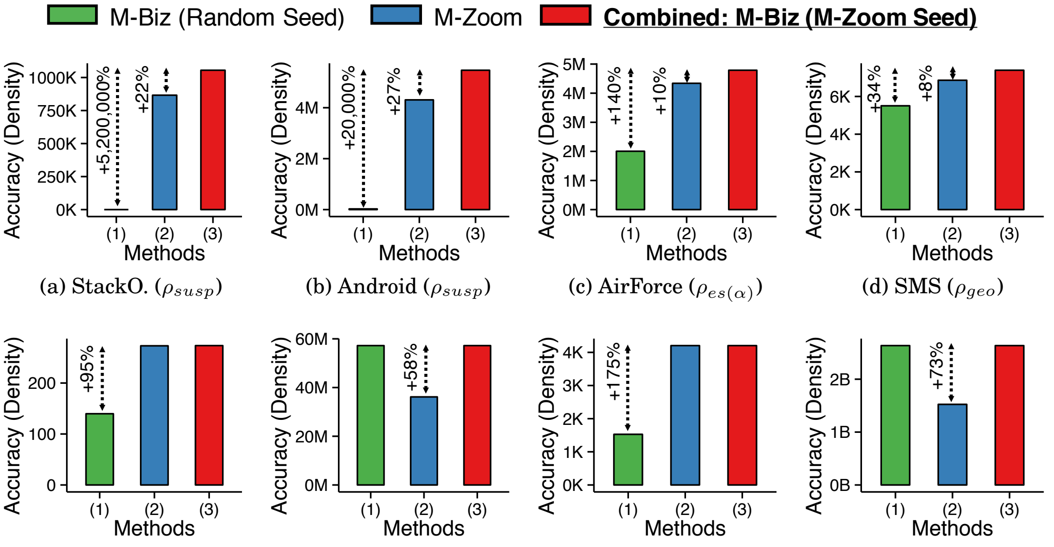
pendix C.4, our methods gave the best trade-off between speed and accuracy regardless of density measures also in most of the other datasets.

5.3. Q2. Complementarity of M-ZOOM and M-BIZ

We show that the combination of M-ZOOM and M-BIZ is more accurate than the individual methods. In Figure 6, the accuracies of M-BIZ using random seed tensors, M-ZOOM, and their combination (i.e., M-BIZ using M-ZOOM for seed-subtensor selection) are compared. As seen in Figures 6(a)-(d), the combined method detected up to 27% denser subtensors than the best individual method. Moreover, the combined method achieved high accuracy regardless of density measures, while the accuracies of the individual methods depended on density measures, as shown in Figures 6(e)-(h).

5.4. Q3. Scalability of M-ZOOM and M-BIZ

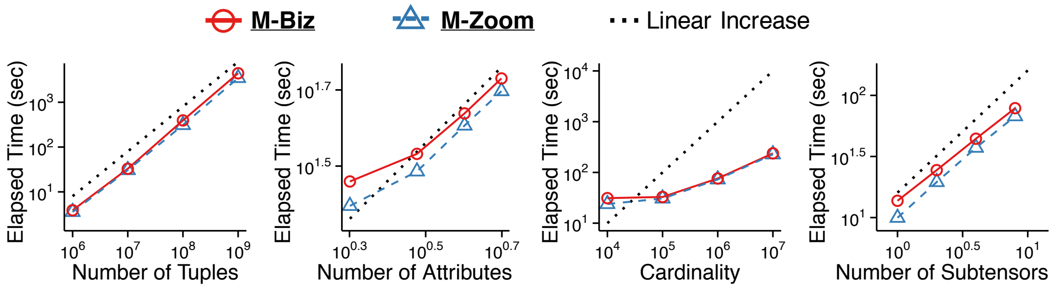
We experimentally demonstrate the scalability of M-ZOOM and M-BIZ. Specifically, we measured the scalability of them with regard to the number of tuples, the number of attributes, the cardinalities of attributes, and the number of subtensors we aim to find. We started with finding one subtensor in a randomly generated 10 millions tuples with three attributes each of whose cardinality is 100 thousands. Then, we measured the running times by changing one factor at a time while fixing the others. As seen in Figure 7, M-ZOOM and M-BIZ scaled linearly with the number of tuples, the number of attributes, and the number of subtensors we aim to find. Moreover, M-ZOOM and M-



(a) StackO. (ρ_{susp}) (b) Android (ρ_{susp}) (c) AirForce ($\rho_{es(\alpha)}$) (d) SMS (ρ_{geo})

(e) KoWiki (ρ_{ari}) (f) KoWiki (ρ_{susp}) (g) EnWiki (ρ_{ari}) (h) EnWiki (ρ_{susp})

Fig. 6: **M-Zoom and M-Biz are complementary.** (a)-(d) their combination (i.e., M-Biz using M-Zoom for seed-subtensor selection) achieves up to 27% higher accuracy than the individual methods. (e)-(h) the combined method achieves high accuracy regardless of density measures, while the accuracies of the individual methods depend on density measures.



(a) Number of Tuples ($|\mathcal{R}|$) (b) Number of Dimension Attributes (N) (c) Cardinality of Dimension Attributes ($|\mathcal{R}_n|$) (d) Number of Subtensors We Aim to Find (k)

Fig. 7: **M-Zoom and M-Biz are scalable.** M-Zoom and M-Biz scale linearly or sub-linearly with the number of tuples, the number of attributes, the cardinalities of attributes, and the number of subtensors we aim to find.

Biz scaled sub-linearly with the cardinalities of attributes, as shown mathematically in Theorems 1 and 6. Although ρ_{susp} was used for the results in Figure 7, similar trends were obtained regardless of density measures.

5.5. Q4. Diversity of Subtensors Found by M-Zoom and M-Biz

We compare the diversity of dense subtensors found by each method. The ability to detect many distinct dense subtensors is important since distinct subtensors may indicate different anomalies or fraud. We define the diversity as the average dissimilarity

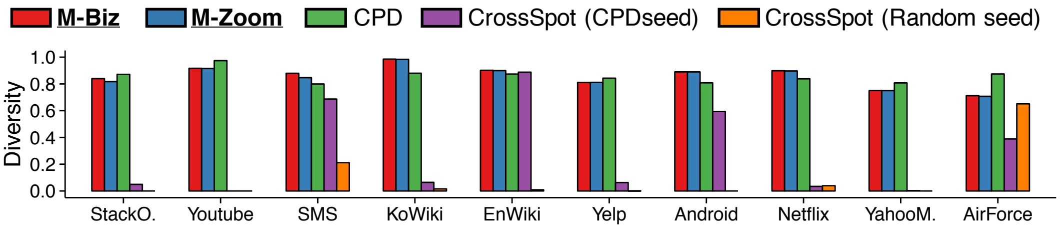


Fig. 8: **M-Zoom and M-Biz detect many distinct dense subtensors.** The dense subtensors found by M-Zoom, M-Biz, and CPD have high diversity, while the dense subtensors found by CrossSpot tend to be almost the same. Here, arithmetic average mass (ρ_{ari}) is used as the density measure. Similar results are obtained with the other density measures, as seen in Figure 10 in Appendix C.3.

Table IV: **M-Zoom and M-Biz detect bot activities in English Wikipedia.** The table lists the three subtensors detected by M-Zoom and M-Biz in EnWiki Dataset.

Method	Order	Volume	Mass	Density (ρ_{geo})	Anomaly Type
M-Zoom	1	1×1,585×6,733	1.93M	8,772	Bot activities
	2	8×12×67.9K	2.43M	13.0K	Bot activities
	3	1×1×90	17.6K	3,933	Bot activities
M-Biz (random seed)	1	4×3,909×15.2K	3.75M	6,062	Bot activities
	2	8×13×73.7K	2.56M	13.0K	Bot activities
	3	2,635 × 29.3K × 87.9K	42.4M	2,240	-
M-Biz (M-Zoom seed)	1	1×2,151×6,811	2.20M	8,978	Bot activities
	2	8×12×70.3K	2.47M	13.1K	Bot activities
	3	1×1×146	21.2K	4,021	Bot activities

between the pairs of subtensors, and the dissimilarity of each pair is defined as

$$dissimilarity(\mathcal{B}, \mathcal{B}') = 1 - \frac{\sum_{n=1}^{N} |\mathcal{B}_n \cap \mathcal{B}'_n|}{\sum_{n=1}^{N} |\mathcal{B}_n \cup \mathcal{B}'_n|}.$$

The average diversity among the three subtensors found by each method using ρ_{ari} as the density measure is shown in Figure 8. In all the datasets, M-Zoom, M-Biz, and CPD successfully detected distinct dense subtensors. CrossSpot, however, found the same subtensor repeatedly or subtensors with slight difference, even when it used different seed-subtensor selection methods. Similar results were obtained when the other density measures were used instead of ρ_{ari}, as seen in Figure 10 in Appendix C.3.

5.6. Q5. Anomaly/Fraud Detection by M-Zoom and M-Biz in Real-world Data

We demonstrate the effectiveness of M-Zoom and M-Biz for anomaly and fraud detection by analyzing dense subtensors detected by them in real-world datasets.

M-Zoom and M-Biz spot bot activities in English Wikipedia. Table V lists the three dense subtensors found by M-Zoom and M-Biz in EnWiki Dataset. All the subtensors detected by M-Zoom and M-Biz (M-Zoom seed) and two subtensors detected by M-Biz (random seed) indicate the activities of bots, which changed the same pages hundreds of thousands of times. Figure 1(d) lists the bots and the pages changed by the bots corresponding to the second subtensor found by all the methods. M-Biz (M-Zoom seed) detected the periods of the bot activities more accurately than M-Zoom and M-Biz (random seed) resulting in subtensors with higher density.

Table V: **M-Zoom and M-Biz (M-Zoom seed) detect edit wars in Korean Wikipedia.** The table lists the three subtensors detected by M-Zoom and M-Biz in KoWiki Dataset.

Method	Order	Volume	Mass	Density (ρ_{ari})	Anomaly Type
M-Zoom	1	2×2×2	546	273.0	Edit war
	2	2×2×3	574	246.0	Edit war
	3	11×10×16	2305	186.9	Edit war
M-Biz (random seed)	1	8,838×41.1K×52.5K	4.77M	139.7	-
	2	18.6K×117K×80.3K	7.06M	98.12	-
	3	35.3K×157K×71.1K	5.15M	58.66	-
M-Biz (M-Zoom seed)	1	2×2×3	638	273.4	Edit war
	2	2×2×3	574	246.0	Edit war
	3	11×10×20	2621	191.8	Edit war

Table VI: **M-Zoom and M-Biz identify network attacks with near-perfect accuracy.** The five dense subtensors found by M-Zoom and M-Biz in AirForce Dataset are composed mostly by network attacks. The fourth and fifth subtensors found by M-Zoom are merged into the fourth subtensor found by M-Biz (M-Zoom seed).

Method	Order	Volume	Density (ρ_{geo})	# Connections	# Attacks (Ratio)	Attack Type
M-Zoom	1	2	2.05M	2.26M	2.26M (100%)	Smurf
	2	1	263K	263K	263K (100%)	Smurf
	3	8.15K	263K	952K	952K (99.9%)	Neptune
	4	1.05M	153K	1.11M	1.06M (95.2%)	Neptune
	5	287K	134K	807K	807K (99.9%)	Neptune
M-Biz (random seed)	1	8	1.88M	2.53M	2.53M (100%)	Smurf
	2	8.65K	264K	963K	963K (99.9%)	Neptune
	3	1.16M	149K	1.09M	1.04M (95.0%)	Neptune
	4	24.8B	23.1K	706K	176K (24.9%)	Smurf
	5	1.28T	19.2K	1.03M	446K (43.2%)	Smurf
M-Biz (M-Zoom seed)	1	2	2.05M	2.26M	2.26M (100%)	Smurf
	2	8.65K	264K	963K	963K (99.9%)	Neptune
	3	1	263K	263K	263K (100%)	Smurf
	4	1.12M	153K	1.12M	1.06M (95.1%)	Neptune
	5	173K	108K	604K	494K (81.8%)	Smurf

M-Zoom and M-Biz spot edit wars in Korean Wikipedia. Table V lists the three dense subtensors found by M-Zoom and M-Biz in KoWiki Dataset. As seen in Figure 1(c), which visualizes the third subtensor found by M-Biz, all the subtensors detected by M-Zoom and M-Biz (M-Zoom seed) indicate edit wars. That is, users with conflicting opinions revised the same set of pages hundreds of times within several hours. The subtensors detected by M-Zoom were extended by M-Biz (M-Zoom seed) so that they indicate more accurate periods of the edit wars and thus have higher density. However, M-Biz (random seed) failed to accruately spot anomalies. The subtensors returned by M-Biz (random seed) were larger but sparser than the subtensors detected by the other methods.

M-Zoom and M-Biz spot network intrusions. Table VI lists the five dense subtensors found by M-Zoom and M-Biz in AirForce Dataset. Based on the provided

Table VII: **M-ZOOM and M-BIZ identify network attacks more accurately than their competitors.**

Method	Density Measure	Area Under ROC Curve (AUC)
CPD [Kolda and Bader 2009]	ρ_{susp}	0.85
MAF [Maruhashi et al. 2011]	ρ_{susp}	0.91
CrossSpot (CPD Seed) [Jiang et al. 2015]	ρ_{susp}	0.92
CrossSpot (Random Seed) [Jiang et al. 2015]	ρ_{geo}	0.93
M-BIZ (Random Seed)	ρ_{geo}	0.95
M-BIZ (M-ZOOM Seed) [Proposed]	ρ_{geo}	0.98
M-ZOOM [Proposed]	ρ_{geo}	0.98

ground-truth labels, most of the connections composing the subtensors were attacks. This indicates that malicious connections form dense subtensors due to the similarity in their behaviors. Based on this observation, we could accurately separate normal connections and attacks based on the densities of the subtensors they belong (i.e., the denser the densest subtensor including a connection is, the more suspicious the connection is). Especially, we got the highest AUC (Area Under the Curve) 0.98 with M-ZOOM and M-BIZ (M-ZOOM seed), as shown in Table VII and the ROC curve in Figure 1(b). This is since M-ZOOM and M-BIZ (M-ZOOM seed) detect many different dense subtensors accurately, as shown in the previous experiments. For each method, we used the best density measure that led to the highest AUC, which is listed in Table VII.

6. RELATED WORK

Dense Subgraph/Submatrix/Subtensor Detection. The *densest subgraph problem*, the problem of finding the subgraph that maximizes average degree, has been extensively studied in theory (see [Lee et al. 2010] for a survey). The two major directions are max-flow based exact algorithms [Goldberg 1984; Khuller and Saha 2009] and greedy algorithms [Charikar 2000; Khuller and Saha 2009] giving a 1/2-approximation to the densest subgraph. The latter direction has been extended to streaming settings [Epasto et al. 2015; McGregor et al. 2015; Bhattacharya et al. 2015] as well as distributed settings [Bahmani et al. 2012; Bahmani et al. 2014]. Variants allow for size restrictions [Andersen and Chellapilla 2009], providing a 1/3-approximation to the densest subgraph for the lower bound case. Variants also allow for multiple dense-subgraph detection [Balalau et al. 2015; Galbrun et al. 2016] or more general density measures, such as edge surplus [Tsourakakis et al. 2013]. A related line of research deals with dense submatrices (or subtensors) in binary matrices (or tensors) where the definition of density is designed for the purpose of frequent itemset mining [Seppänen and Mannila 2004] or formal concept mining [Cerf et al. 2008; Ignatov et al. 2013].

Anomaly/Fraud Detection based on Dense Subgraphs. Spectral approaches make use of eigendecomposition or SVD of adjacency matrices for dense-subgraph detection. Such approaches have been used to spot anomalous pattens in a patent graph [Prakash et al. 2010], lockstep followers in a social network [Jiang et al. 2014b], and stealthy or small-scale attacks in social networks [Shah et al. 2014]. Other approaches include NETPROBE [Pandit et al. 2007], which used belief propagation to detect fraud-accomplice bipartite cores in an auction network, and COPYCATCH [Beutel et al. 2013], which used one-class clustering and sub-space clustering to identify 'Like' boosting on Facebook. In addition, ODDBALL [Akoglu et al. 2010] spotted near-cliques in a graph of posts in blogs based on egonet features. Recently, FRAUDAR [Hooi et al. 2017] and CORESCOPE [Shin et al. 2016a], which generalize densest subgraph-detection methods so that the suspiciousness of each node and edge can be incorporated, spotted follower-buying services on Twitter.

Anomaly/Fraud Detection based on Dense Subtensors. Spectral methods for dense subgraphs can be extended to tensors where tensor decomposition, such as CP Decomposition and HOSVD [Kolda and Bader 2009], is used to spot dense subtensors (see Appendix A). MAF [Maruhashi et al. 2011], which is based on CP Decomposition, detected dense subtensors corresponding to port-scanning activities from network traffic logs. Another approach is CROSSSPOT [Jiang et al. 2015], which spotted retweet boosting in Weibo, outperforming HOSVD. CROSSSPOT starts from a seed subtensor and updates the values in one attribute of the subtensor at a time, while fixing the values of the other attributes, until ρ_{susp} (see Definition 4) converges. CROSSSPOT, however, was up to $114\times$ slower than M-BIZ, as shown in Section 5.2, since each step in CROSSSPOT involves sorting all the values of the updated attribute and accessing a large number of tuples. These costs, however, are avoided in M-BIZ, which maintains and updates heaps for attribute values inside or on the boundary of the maintained dense subtensor (see Theorem 6).

Our M-ZOOM and M-BIZ non-trivially generalize theoretical results regarding the densest subgraph problem, especially [Andersen and Chellapilla 2009] and [Tsourakakis et al. 2013], for supporting tensors, various density measures, size bound, and multi-subtensor detection. As seen in Table I, M-ZOOM and M-BIZ provide more functionalities than the other methods for detecting dense subgraphs and/or subtensors. Recently, M-ZOOM was extended to the external-memory, distributed, and streaming settings [Shin et al. 2017a; Shin et al. 2017b].

7. CONCLUSION

In this work, we propose M-ZOOM and M-BIZ, complementary and combinable algorithms for fast and accurate dense-subtensor detection. They have the following advantages over state-of-the-art methods:

- **Scalable:** M-ZOOM and M-BIZ are up to **114× faster** than competitors with similar accuracy (Figure 5) due to their near-linear scalability with all input factors (Figure 7).
- **Provably accurate:** M-ZOOM and M-BIZ guarantee the approximation bounds and the local optimality of the subtensors they find, respectively (Theorems 3, 5, and 7). In addition, they show high accuracy in real-world tensors (Figure 5).
- **Flexible:** M-ZOOM and M-BIZ support high-order tensors, various density measures, multi-subtensor detection, and size bounds (Table I).
- **Effective:** M-ZOOM and M-BIZ detected network attacks from a TCP dump with near-perfect accuracy (**AUC=0.98**). They also successfully detected edit wars and bot activities in Wikipedia (Tables IV-VII).

Reproducibility: The source code and data used in the paper are available at **http://www.cs.cmu.edu/~kijungs/codes/mzoom/**.

APPENDIX

A. CP DECOMPOSITION (CPD)

In a graph, dense subgraphs lead to high singular values of the adjacency matrix [Shah et al. 2014]. The singular vectors corresponding to the high singular values roughly indicate which nodes form the dense subgraphs. This idea can be extended to tensors, where dense subtensors are captured by components in CP Decomposition [Kolda and Bader 2009]. Let $\mathbf{A}^{(1)} \in \mathbb{R}^{|\mathcal{R}_1|\times k}$, $\mathbf{A}^{(2)} \in \mathbb{R}^{|\mathcal{R}_2|\times k}$, ..., $\mathbf{A}^{(N)} \in \mathbb{R}^{|\mathcal{R}_N|\times k}$ be the factor matrices obtained by the rank-k CP Decomposition of $\mathcal{R}$. For each $i \in [k]$, we form a subtensor with every attribute value a_n whose corresponding element in the i-th column of $\mathbf{A}^{(n)}$ is at least $1/\sqrt{|\mathcal{R}_n|}$.

B. PROOFS

B.1. Proof of Theorem 1

Proof. In Algorithm 3, lines 1-4 take $O(N)$ for all the considered density measures (i.e., ρ_{ari}, ρ_{geo}, ρ_{susp}, and $\rho_{es(\alpha)}$) if we maintain and update aggregated values (e.g., M_B, S_B, and V_B) instead of computing $\rho(\mathcal{B} - \mathcal{B}(a_n^-, n), \mathcal{R})$ from scratch every time. In addition, line 5 takes $O(\log|\mathcal{R}_{n^-}|)$ and line 8 takes $O(1)$ if we use Fibonacci heaps. Algorithm 2, whose computational bottleneck is line 7, has time complexity $O(N|\mathcal{R}| + N\sum_{n=1}^{N}|\mathcal{R}_n| + \sum_{n=1}^{N}|\mathcal{R}_n|\log|\mathcal{R}_n|)$ since lines 1-4 of Algorithm 3 are executed $S_{\mathcal{R}} = \sum_{n=1}^{N}|\mathcal{R}_n|$ times, line 5 is executed $|\mathcal{R}_{n^-}|$ times for each $n^- \in [N]$, and line 8 is executed $N|\mathcal{R}|$ times. Algorithm 1, whose computational bottleneck is line 4, has time complexity $O(kN|\mathcal{R}| + kN\sum_{n=1}^{N}|\mathcal{R}_n| + k\sum_{n=1}^{N}|\mathcal{R}_n|\log|\mathcal{R}_n|)$ since Algorithm 2 is executed k times. From $L = \max_{n\in[N]}|\mathcal{R}_n|$, the time complexity of Algorithm 1 becomes $O(kN(|\mathcal{R}| + NL + L\log L))$. Since $N = O(\log L)$ (by assumption) and $L \leq |\mathcal{R}|$ (by definition), $|\mathcal{R}| + NL + L\log L = O(|\mathcal{R}|\log L)$. Thus, the time complexity of Algorithm 1 is $O(kN|\mathcal{R}|\log L)$. ■

B.2. Proof of Theorem 2

Proof. Algorithm 2 requires $O(N|\mathcal{R}|)$ space for $\mathcal{R}$ and $\mathcal{B}$; and $O(\sum_{n=1}^{N}|\mathcal{R}_n|)$ space for min-heaps and the order by which attribute values are removed, as explained in Section 3.2. The sum is $O(N|\mathcal{R}| + \sum_{n=1}^{N}|\mathcal{R}_n|) = O(N|\mathcal{R}|)$ since $|\mathcal{R}_n| \leq |\mathcal{R}|$, $\forall n \in [N]$. Since Algorithm 1 requires additional $O(kN|\mathcal{R}|)$ space for storing k subtensors it finds, its space complexity is $O(N|\mathcal{R}| + kN|\mathcal{R}|) = O(kN|\mathcal{R}|)$. ■

B.3. Proof of Lemma 2

Let $\mathcal{B}^{(r)}$ be the relation $\mathcal{B}$ at the beginning of the r-th iteration of Algorithm 2 with ρ_{ari} as the density measure. Then, let $n^{(r)}$ and $a^{(r)}$ be n^- and a^- in the same iteration. That is, in the r-th iteration, value $a^{(r)} \in \mathcal{B}_{n^{(r)}}^{(r)}$ is removed from attribute $A_{n^{(r)}}$.

Lemma 4. *$n^{(r)} \in [N]$ and $a^{(r)} \in \mathcal{B}_n^{(r)}$ are $n \in [N]$ and $a \in \mathcal{B}_n^{(r)}$ minimizing $mass(\mathcal{B}^{(r)}(a, n))$.*

Proof. By line 7 of Algorithm 2, $\rho_{ari}(\mathcal{B}^{(r)} - \mathcal{B}^{(r)}(a^{(r)}, n^{(r)}), \mathcal{R}) \geq \rho_{ari}(\mathcal{B}^{(r)} - \mathcal{B}^{(r)}(a, n), \mathcal{R})$, $\forall n \in [N]$, $\forall a \in \mathcal{B}_n^{(r)}$. From this, we have

$$\begin{aligned} mass(\mathcal{B}^{(r)}(a^{(r)}, n^{(r)})) &= mass(\mathcal{B}^{(r)}) - mass(\mathcal{B}^{(r)} - \mathcal{B}^{(r)}(a^{(r)}, n^{(r)})) \\ &= mass(\mathcal{B}^{(r)}) - \rho_{ari}(\mathcal{B}^{(r)} - \mathcal{B}^{(r)}(a^{(r)}, n^{(r)}), \mathcal{R})\frac{size(\mathcal{B}^{(r)}) - 1}{N} \\ &\leq mass(\mathcal{B}^{(r)}) - \rho_{ari}(\mathcal{B}^{(r)} - \mathcal{B}^{(r)}(a, n), \mathcal{R})\frac{size(\mathcal{B}^{(r)}) - 1}{N} \\ &= mass(\mathcal{B}^{(r)}) - mass(\mathcal{B}^{(r)} - \mathcal{B}^{(r)}(a, n)) = mass(\mathcal{B}^{(r)}(a, n)). \end{aligned}$$

■

Proof of Lemma 2.

Proof. Let r be the first iteration in Algorithm 2 where $a^{(r)} \in \mathcal{B}'_{n^{(r)}}$. Since $\mathcal{B}^{(r)} \supset \mathcal{B}'$, $mass(\mathcal{B}^{(r)}(a^{(r)}, n^{(r)})) \geq mass(\mathcal{B}'(a^{(r)}, n^{(r)})) \geq c$. By Lemma 4, $\forall n \in [N]$, $\forall a \in \mathcal{B}_n^{(r)}$, $mass(\mathcal{B}^{(r)}(a, n)) \geq mass(\mathcal{B}^{(r)}(a^{(r)}, n^{(r)})) \geq c$. ■

B.4. Proof of Theorem 4

Let $\mathcal{B}^{(r)}$ be the relation $\mathcal{B}$ at the beginning of the r-th iteration of Algorithm 2 with ρ_{ari} as the density measure. Then, let $n^{(r)}$ and $a^{(r)}$ be n^{-} and a^{-} in the same iteration. That is, in the r-th iteration, value $a^{(r)} \in \mathcal{B}^{(r)}_{n^{(r)}}$ is removed from attribute $A_{n^{(r)}}$.

Lemma 5. *For any* $\alpha \in [0,1]$*, there exists a subtensor* $\mathcal{B}'$ *satisfying* $\forall n \in [N]$*,* $\forall a \in \mathcal{B}'_n$*,* $mass(\mathcal{B}'(a,n)) \geq \alpha \cdot \rho_{ari}(\mathcal{R},\mathcal{R})/N$ *and* $mass(\mathcal{B}') \geq (1-\alpha) \cdot mass(\mathcal{R})$*.*

Proof. Let s be the first iteration in Algorithm 2 where $mass(\mathcal{B}^{(s)}(a^{(s)},n^{(s)})) \geq \alpha \cdot \rho_{ari}(\mathcal{R},\mathcal{R})/N$. Such s exists, otherwise

$$mass(\mathcal{R}) = \sum\nolimits_{r=1}^{size(\mathcal{R})} mass(\mathcal{B}^{(r)}(a^{(r)},n^{(r)})) < \frac{\alpha \cdot \rho_{ari}(\mathcal{R},\mathcal{R})}{N} size(\mathcal{R}) = \alpha \cdot mass(\mathcal{R}),$$

which is a contradiction. Then,

$$\begin{aligned} mass(\mathcal{R}) &= \sum\nolimits_{r=1}^{size(\mathcal{R})} mass(\mathcal{B}^{(r)}(a^{(r)},n^{(r)})) \\ &= \sum\nolimits_{r=1}^{s-1} mass(\mathcal{B}^{(r)}(a^{(r)},n^{(r)})) + \sum\nolimits_{r=s}^{size(\mathcal{R})} mass(\mathcal{B}^{(r)}(a^{(r)},n^{(r)})) \\ &\leq (s-1) \cdot \alpha \cdot \rho_{ari}(\mathcal{R},\mathcal{R})/N + mass(\mathcal{B}^{(s)}) \\ &\leq \alpha \cdot size(\mathcal{R}) \cdot \rho_{ari}(\mathcal{R},\mathcal{R})/N + mass(\mathcal{B}^{(s)}) = \alpha \cdot mass(\mathcal{R}) + mass(\mathcal{B}^{(s)}). \end{aligned}$$

Thus, $mass(\mathcal{B}^{(s)}) \geq (1-\alpha) \cdot mass(\mathcal{R})$. ■

Proof of Theorem 4

Proof. Let $\alpha = N/(N+1)$. By Lemma 5, there exists a subtensor $\bar{\mathcal{B}} \subset \mathcal{B}^*$ satisfying $\forall n \in [N]$, $\forall a \in \bar{\mathcal{B}}_n$, $mass(\bar{\mathcal{B}}(a,n)) \geq \alpha \cdot \rho_{ari}(\mathcal{B}^*,\mathcal{B}^*)/N = \alpha \cdot \rho_{ari}(\mathcal{B}^*,\mathcal{R})/N$ and $mass(\bar{\mathcal{B}}) \geq (1-\alpha) \cdot mass(\mathcal{B}^*)$. Let s be the first iteration of Algorithm 2 where $a^{(s)} \in \bar{\mathcal{B}}_{n^{(s)}}$. By Lemma 4 and $\mathcal{B}^{(s)} \supset \bar{\mathcal{B}}$, we have $\forall n \in [N]$, $\forall a \in \mathcal{B}^{(s)}_n$, $mass(\mathcal{B}^{(s)}(a,n)) \geq mass(\mathcal{B}^{(s)}(a^{(s)},n^{(s)})) \geq mass(\bar{\mathcal{B}}(a^{(s)},n^{(s)})) \geq \alpha \cdot \rho_{ari}(\mathcal{B}^*,\mathcal{R})/N$ and $mass(\mathcal{B}^{(s)}) \geq (1-\alpha) \cdot mass(\mathcal{B}^*)$.
If $size(\mathcal{B}^{(s)}) \geq S_{min}$,

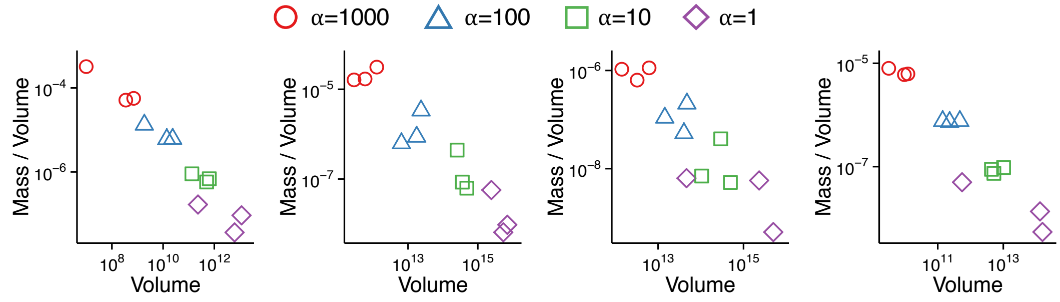
$$\begin{aligned} \rho_{ari}(\mathcal{B}',\mathcal{R}) &\geq \rho_{ari}(\mathcal{B}^{(s)},\mathcal{R}) = \frac{mass(\mathcal{B}^{(s)})}{size(\mathcal{B}^{(s)})/N} \\ &= \frac{\sum_{n\in[N]}\sum_{a\in\mathcal{B}^{(s)}_n} mass(\mathcal{B}^{(s)}(a,n))}{size(\mathcal{B}^{(s)})} \geq \frac{\alpha \cdot size(\mathcal{B}^{(s)})\rho_{ari}(\mathcal{B}^*,\mathcal{R})/N}{size(\mathcal{B}^{(s)})} = \frac{\rho_{ari}(\mathcal{B}^*,\mathcal{R})}{N+1}. \end{aligned}$$

If $size(\mathcal{B}^{(s)}) < S_{min}$, we consider $\mathcal{B}^{(q)}$ where $size(\mathcal{B}^{(q)}) = S_{min}$ and thus $q < s$. Then,

$$\begin{aligned} \rho_{ari}(\mathcal{B}',\mathcal{R}) &\geq \rho_{ari}(\mathcal{B}^{(q)},\mathcal{R}) = \frac{mass(\mathcal{B}^{(q)})}{size(\mathcal{B}^{(q)})/N} \geq \frac{mass(\mathcal{B}^{(s)})}{size(\mathcal{B}^{(q)})/N} \\ &\geq \frac{(1-\alpha)\cdot mass(\mathcal{B}^*)}{size(\mathcal{B}^{(q)})/N} = \frac{mass(\mathcal{B}^*)/(N+1)}{S_{min}/N} \geq \frac{mass(\mathcal{B}^*)/(N+1)}{size(\mathcal{B}^*)/N} = \frac{\rho_{ari}(\mathcal{B}^*,\mathcal{R})}{N+1}. \end{aligned}$$

Hence, regardless of $size(\mathcal{B}^{(s)})$, we have $\rho_{ari}(\mathcal{B}',\mathcal{R}) \geq \rho_{ari}(\mathcal{B}^*,\mathcal{R})/(N+1)$. ■

We remark that the above proof of Theorem 4 is a multi-dimensional generalization of the proof of Theorem 1 in [Andersen and Chellapilla 2009].



(a) StackO. (b) SMS (c) KoWiki (d) Yelp

Fig. 9: **Subtensors detected by M-BIZ are configurable by adjusting α in $\rho_{es(\alpha)}$.** With high α values, M-BIZ detects relatively small compact subtensors. With small α values, however, M-BIZ detects relatively large sparse subtensors.

B.5. Proof of Theorem 6

Proof. The functions *choose_attribute_value_to_add* and *choose_attribute_value_to_delete* in Algorithm 5 take $O(N)$ since, for all the considered density measures (i.e., ρ_{ari}, ρ_{geo}, ρ_{susp}, and $\rho_{es(\alpha)}$), computing $\rho(\boldsymbol{\mathcal{B}} \cup \partial_{\boldsymbol{\mathcal{B}},\boldsymbol{\mathcal{R}}}(a_n^+, n), \boldsymbol{\mathcal{R}})$ or $\rho(\boldsymbol{\mathcal{B}} - \boldsymbol{\mathcal{B}}(a_n^-, n), \boldsymbol{\mathcal{R}})$ takes $O(1)$ if we maintain and update aggregated values (e.g., M_B, S_B, and V_B) instead of computing them from scratch. The functions *update_heaps_for_addition* and *update_heaps_for_deletion* in Algorithm 5 take $O(N|\boldsymbol{\mathcal{R}}(a^*, n^*)| \log L)$ since the number of key updates is at most $N|\boldsymbol{\mathcal{R}}(a^*, n^*)|$ and each update takes $O(\log L)$.

The overall time complexity of an iteration in Algorithm 4 is $O(N|\boldsymbol{\mathcal{R}}(a^*, n^*)| \log L)$. This is because (1) each function in Algorithm 5, which takes $O(N|\boldsymbol{\mathcal{R}}(a^*, n^*)| \log L)$, is called at most once, and (2) the remaining parts of the iteration take $O(N|\boldsymbol{\mathcal{R}}(a^*, n^*)|)$. ■

C. ADDITIONAL EXPERIMENTS

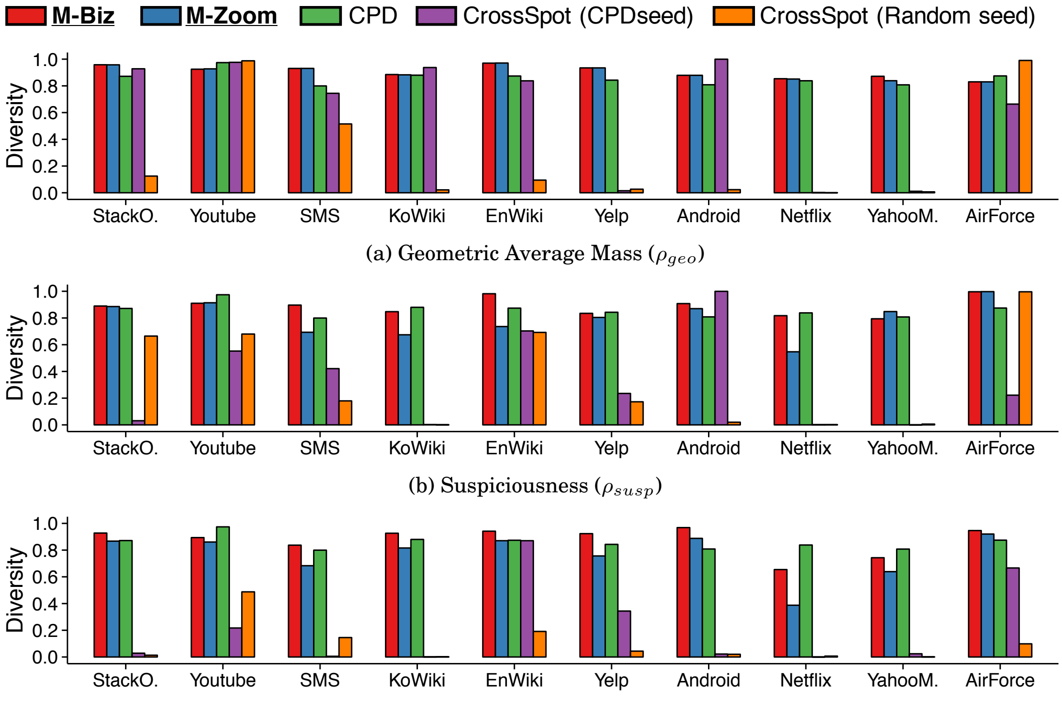
C.1. AirForce Dataset

The descriptions of the attributes in AirForce Dataset are as follows:

- *protocol* (A_1): type of protocol (e.g. tcp, udp, etc.)
- *service* (A_2): type of network service on destination (e.g., http, telnet, etc)
- *flag* (A_3): normal or error status of each connection
- *src-bytes* (A_4): amount of data bytes from source to destination
- *dst-bytes* (A_5): amount of data bytes from destination to source
- *counts* (A_6): number of connections to the same host in the past two seconds
- *srv-counts* (A_7): number of connections to the same service in the past two seconds
- *#connections* (X): number of connections with the corresponding dimension attribute values.

C.2. Effect of α in Entry Surplus on Detected Subtensors

We show that subtensors detected by our methods are configurable by adjusting α in entry surplus $\rho_{es(\alpha)}$. Figure 9 shows the volumes and average entry values of the subtensors found by M-BIZ in real-world datasets when $\rho_{es(\alpha)}$ was used as the density measure. With large α values, M-BIZ tended to detect relatively small compact subtensors. However, with small α values, M-BIZ tended to detect relatively large sparse subtensors. Similar results were obtained with M-ZOOM.



(a) Geometric Average Mass (ρ_{geo})

(b) Suspiciousness (ρ_{susp})

(c) Entry Surplus ($\rho_{es(\alpha)}$)

Fig. 10: **M-ZOOM and M-BIZ detect many distinct dense subtensors regardless of used density measures.** The dense subtensors found by M-ZOOM, M-BIZ, and CPD have high diversity in all datasets, while the dense subtensors found by CROSSSPOT are almost the same in many datasets.

C.3. Diversity of Subtensors Found by M-ZOOM and M-BIZ

Figure 10 shows the diversity (see Section 5.5 for the definition) of the subtensors found by each method with different density measures. As explained in Section 5.5, M-ZOOM, M-BIZ, and CPD successfully detected distinct dense subtensors, while CROSSSPOT tended to find the same subtensor repeatedly or subtensors with slight difference.

C.4. Running Time and Accuracy of M-ZOOM and M-BIZ

Figure 11 shows the speed and accuracy of each dense-subtensor detection method in real-world datasets. As explained in Section 5.2, M-ZOOM and M-BIZ provided the best trade-off between speed and accuracy regardless of density measures, in most datasets. Especially, M-ZOOM and M-BIZ detected denser subtensors up to 54 × faster than CROSSSPOT when ρ_{ari} was used as the density measure.

REFERENCES

Leman Akoglu, Mary McGlohon, and Christos Faloutsos. 2010. Oddball: Spotting anomalies in weighted graphs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 410–421.

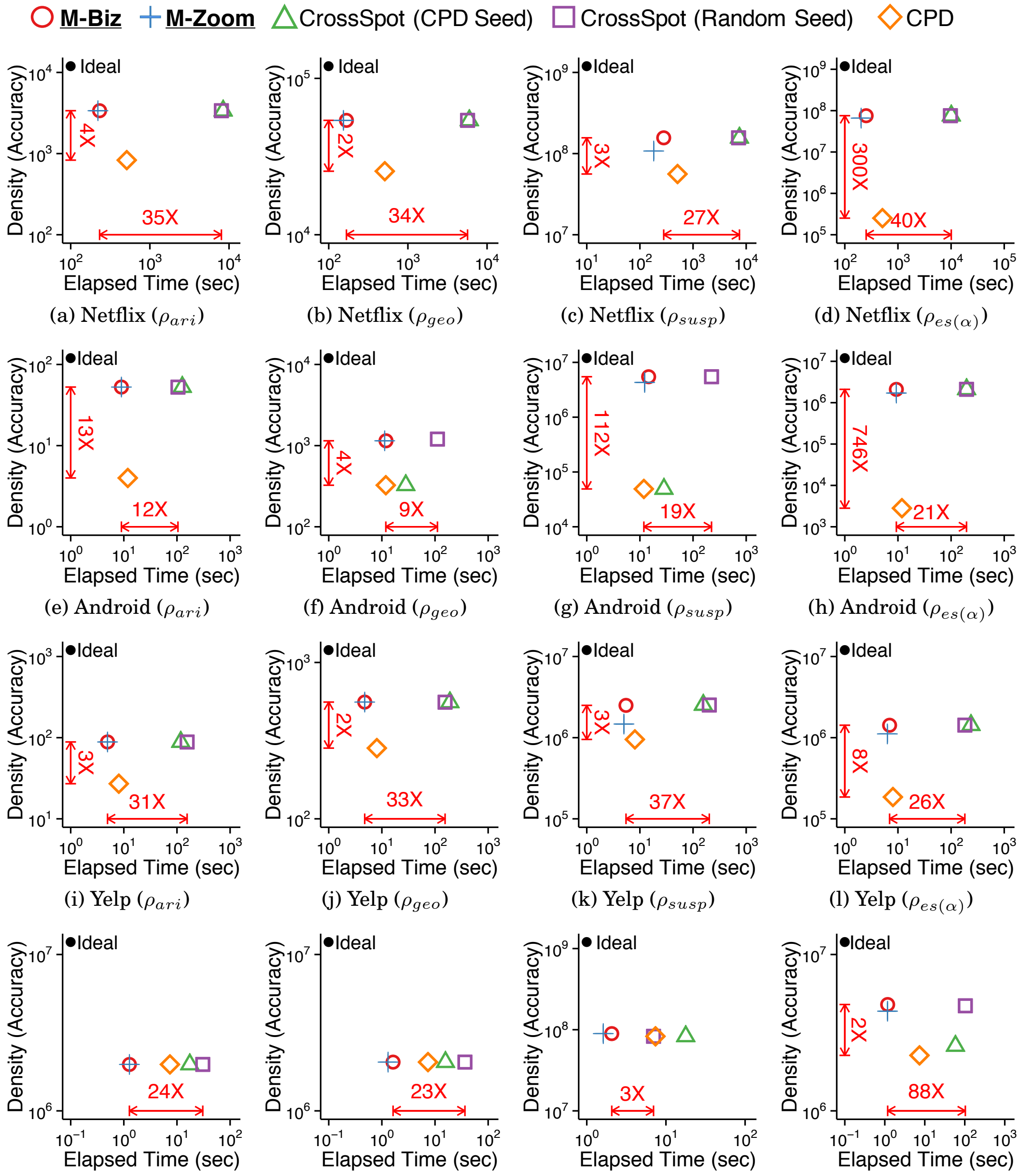
Reid Andersen and Kumar Chellapilla. 2009. Finding dense subgraphs with size bounds. In *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 25–37.

Brett W. Bader and Tamara G. Kolda. 2007. Efficient MATLAB computations with sparse and factored tensors. *SIAM Journal on Scientific Computing* 30, 1 (2007), 205–231.

Bahman Bahmani, Ashish Goel, and Kamesh Munagala. 2014. Efficient primal-dual graph algorithms for mapreduce. In *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 59–78.

Bahman Bahmani, Ravi Kumar, and Sergei Vassilvitskii. 2012. Densest subgraph in streaming and mapreduce. *Proceedings of the VLDB Endowment* 5, 5 (2012), 454–465.

Oana Denisa Balalau, Francesco Bonchi, TH Chan, Francesco Gullo, and Mauro Sozio. 2015. Finding subgraphs with maximum total density and limited overlap. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, 379–388.



(a) Netflix (ρ_{ari}) (b) Netflix (ρ_{geo}) (c) Netflix (ρ_{susp}) (d) Netflix ($\rho_{es(\alpha)}$)

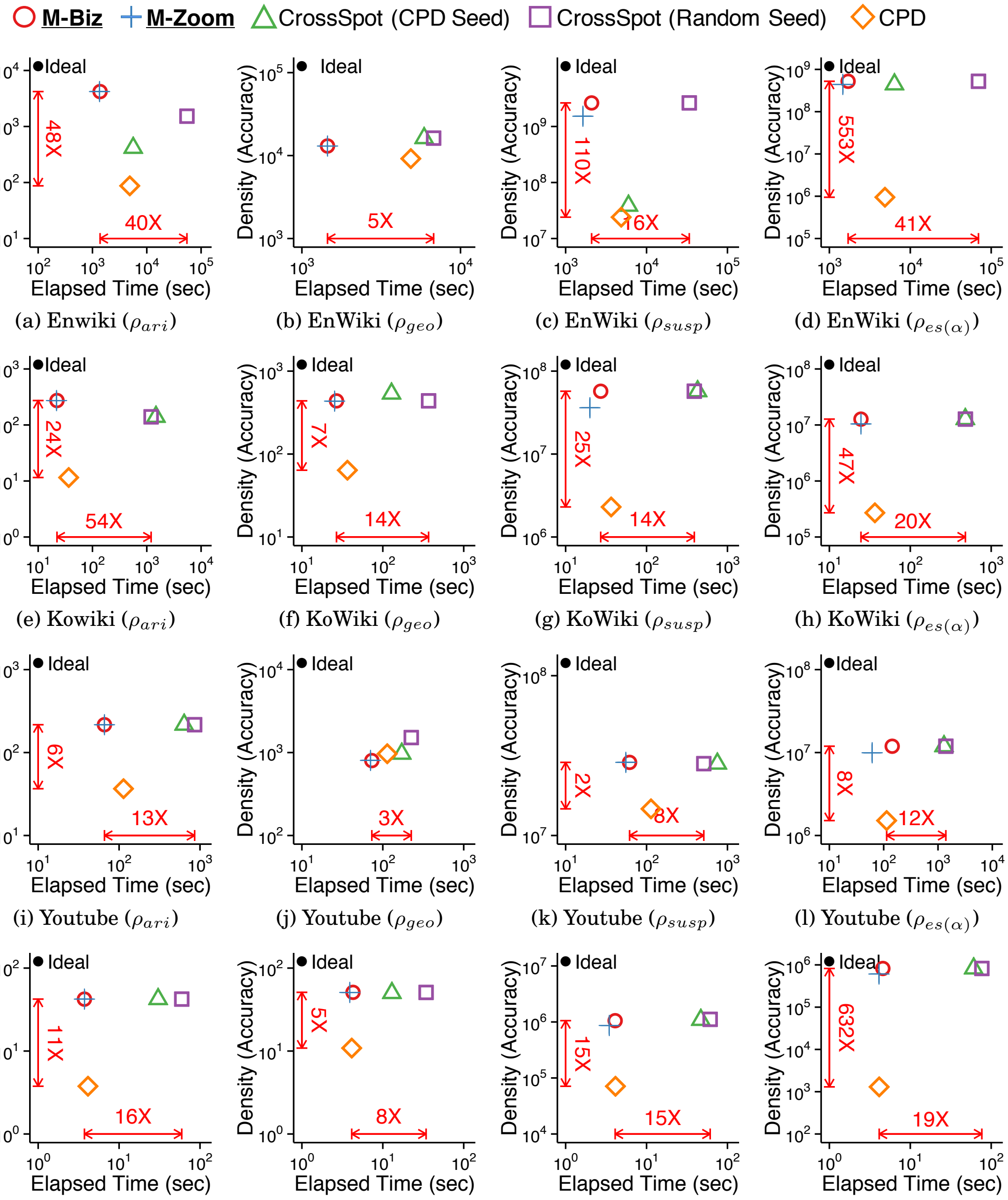
(e) Android (ρ_{ari}) (f) Android (ρ_{geo}) (g) Android (ρ_{susp}) (h) Android ($\rho_{es(\alpha)}$)

(i) Yelp (ρ_{ari}) (j) Yelp (ρ_{geo}) (k) Yelp (ρ_{susp}) (l) Yelp ($\rho_{es(\alpha)}$)

(m) Intrusion (ρ_{ari}) (n) Intrusion (ρ_{geo}) (o) Intrusion (ρ_{susp}) (p) Intrusion ($\rho_{es(\alpha)}$)

(Continues on the next page)

(Continues from the previous page)



(a) Enwiki (ρ_{ari}) (b) EnWiki (ρ_{geo}) (c) EnWiki (ρ_{susp}) (d) EnWiki ($\rho_{es(\alpha)}$)

(e) Kowiki (ρ_{ari}) (f) KoWiki (ρ_{geo}) (g) KoWiki (ρ_{susp}) (h) KoWiki ($\rho_{es(\alpha)}$)

(i) Youtube (ρ_{ari}) (j) Youtube (ρ_{geo}) (k) Youtube (ρ_{susp}) (l) Youtube ($\rho_{es(\alpha)}$)

(m) StackO. (ρ_{ari}) (n) StackO. (ρ_{geo}) (o) StackO. (ρ_{susp}) (p) StackO. ($\rho_{es(\alpha)}$)

Fig. 11: **Only M-Zoom and M-Biz achieve both speed and accuracy.** In each plot, points represent the speed of different methods and the highest density of the three subtensors found by the methods. Upper-left region indicates better performance. In most datasets, M-Zoom and M-Biz give the best trade-off between speed and density regardless of density measures.

James Bennett and Stan Lanning. 2007. The netflix prize. In *Proceedings of KDD cup and workshop*, Vol. 2007. ACM, 35.

Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, and Christos Faloutsos. 2013. Copycatch: stopping group attacks by spotting lockstep behavior in social networks. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 119–130.

Sayan Bhattacharya, Monika Henzinger, Danupon Nanongkai, and Charalampos Tsourakakis. 2015. Space- and time-efficient algorithm for maintaining dense subgraphs on one-pass dynamic streams. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*. ACM, 173–182.

Loïc Cerf, Jérémy Besson, Céline Robardet, and Jean-François Boulicaut. 2008. Data Peeler: Contraint-Based Closed Pattern Mining in n-ary Relations. In *proceedings of the 2008 SIAM International conference on Data Mining*. SIAM, 37–48.

Moses Charikar. 2000. Greedy approximation algorithms for finding dense components in a graph. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*. Springer, 84–95.

Gideon Dror, Noam Koenigstein, Yehuda Koren, and Markus Weimer. 2012. The Yahoo! Music Dataset and KDD-Cup'11. In *Proceedings of KDD Cup 2011*. ACM, 3–18.

Alessandro Epasto, Silvio Lattanzi, and Mauro Sozio. 2015. Efficient densest subgraph computation in evolving graphs. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 300–310.

Esther Galbrun, Aristides Gionis, and Nikolaj Tatti. 2016. Top-k overlapping densest subgraphs. *Data Mining and Knowledge Discovery* 30, 5 (2016), 1134–1165.

David Gibson, Ravi Kumar, and Andrew Tomkins. 2005. Discovering large dense subgraphs in massive graphs. In *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment, 721–732.

Andrew V Goldberg. 1984. *Finding a maximum density subgraph*. University of California Berkeley, CA.

Bryan Hooi, Kijung Shin, Hyun Ah Song, Alex Beutel, Neil Shah, and Christos Faloutsos. 2017. Graph-Based Fraud Detection in the Face of Camouflage. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11, 4 (2017), 44.

Dmitry I Ignatov, Sergei O Kuznetsov, Jonas Poelmans, and Leonid E Zhukov. 2013. Can triconcepts become triclusters? *International Journal of General Systems* 42, 6 (2013), 572–593.

Meng Jiang, Alex Beutel, Peng Cui, Bryan Hooi, Shiqiang Yang, and Christos Faloutsos. 2015. A general suspiciousness metric for dense blocks in multimodal data. In *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE, 781–786.

Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, and Shiqiang Yang. 2014a. Catchsync: catching synchronized behavior in large directed graphs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 941–950.

Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, and Shiqiang Yang. 2014b. Inferring strange behavior from connectivity pattern in social networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 126–138.

Ravi Kannan and V Vinay. 1999. *Analyzing the structure of large graphs*. Rheinische Friedrich-Wilhelms-Universität Bonn.

Samir Khuller and Barna Saha. 2009. On Finding Dense Subgraphs. In *International Colloquium on Automata, Languages, and Programming*. Springer, 597–608.

Tamara G Kolda and Brett W Bader. 2009. Tensor decompositions and applications. *SIAM review* 51, 3 (2009), 455–500.

Jérôme Kunegis. 2013. Konect: the koblenz network collection. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 1343–1350.

Victor E Lee, Ning Ruan, Ruoming Jin, and Charu Aggarwal. 2010. A survey of algorithms for dense subgraph discovery. In *Managing and Mining Graph Data*. Springer, 303–336.

Koji Maruhashi, Fan Guo, and Christos Faloutsos. 2011. Multiaspectforensics: Pattern mining on large-scale heterogeneous networks with tensor analysis. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*. IEEE, 203–210.

Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 785–794.

Andrew McGregor, David Tench, Sofya Vorotnikova, and Hoa T Vu. 2015. Densest subgraph in dynamic graph streams. In *International Symposium on Mathematical Foundations of Computer Science*. Springer, 472–482.

Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement and Analysis of Online Social Networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 29–42.

Shashank Pandit, Duen Horng Chau, Samuel Wang, and Christos Faloutsos. 2007. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 201–210.

B Aditya Prakash, Ashwin Sridharan, Mukund Seshadri, Sridhar Machiraju, and Christos Faloutsos. 2010. Eigenspokes: Surprising patterns and scalable community chipping in large graphs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 435–448.

Jouni K Seppänen and Heikki Mannila. 2004. Dense itemsets. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 683–688.

Neil Shah, Alex Beutel, Brian Gallagher, and Christos Faloutsos. 2014. Spotting suspicious link behavior with fbox: An adversarial perspective. In *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, 959–964.

Kijung Shin, Tina Eliassi-Rad, and Christos Faloutsos. 2016a. CoreScope: Graph Mining Using k-Core Analysis - Patterns, Anomalies and Algorithms. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 469–478.

Kijung Shin, Bryan Hooi, and Christos Faloutsos. 2016b. M-Zoom: Fast Dense-Block Detection in Tensors with Quality Guarantees. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 264–280.

Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos. 2017a. D-cube: Dense-block detection in terabyte-scale tensors. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 681–689.

Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos. 2017b. DenseAlert: Incremental Dense-Subtensor Detection in Tensor Streams. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1057–1066.

Charalampos Tsourakakis, Francesco Bonchi, Aristides Gionis, Francesco Gullo, and Maria Tsiarli. 2013. Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 104–112.