2020 IEEE 36th International Conference on Data Engineering (ICDE)

Multiple Dense Subtensor Estimation with High Density Guarantee

Quang-Huy Duong*
Department of Computer Science
NTNU, Trondheim, Norway
huydqyb@gmail.com

Heri Ramampiaro
Department of Computer Science
NTNU, Trondheim, Norway
heri@ntnu.no

Kjetil Nørvåg
Department of Computer Science
NTNU, Trondheim, Norway
noervaag@ntnu.no

***Abstract*—Dense subtensor detection is a well-studied area, with a wide range of applications, and numerous efficient approaches and algorithms have been proposed. Existing algorithms are generally efficient for dense subtensor detection and could perform well in many applications. However, the main drawback of most of these algorithms is that they can estimate only one subtensor at a time, with a low guarantee on the subtensor's density. While some methods can, on the other hand, estimate multiple subtensors, they can give a guarantee on the density with respect to the input tensor for the first estimated subsensor only. We address these drawbacks by providing both theoretical and practical solution for estimating multiple dense subtensors in tensor data. In particular, we guarantee and prove a higher bound of the lower-bound density of the estimated subtensors. We also propose a novel approach to show that there are multiple dense subtensors with a guarantee on its density that is greater than the lower bound used in the state-of-the-art algorithms. We evaluate our approach with extensive experiments on several real-world datasets, which demonstrates its efficiency and feasibility.**

***Index Terms*—Tensor, Dense Subtensor, Dense Subgraph, Multiple Subtensor Detection, Density Guarantee.**

I. Introduction

In many real-world applications, generated data are commonly represented as multidimensional array data, referred to as *tensor* [1]. Tensors have been used in several important domains, including geometry and physics, as well as computer science [2], [3]. As a result of the growth in the number of applications involving tensors, combined with the increase of researchers' interests, numerous tensor-related approaches have been proposed, including tensor decomposition [4], [5] and tensor factorization [6]–[8].

An important task related to tensors, addressed in this paper, is the detection of dense subtensors in a (stream of) tensors. Dense subtensor detection has many applications, including detecting intrusions and changes in communication networks [9], [10], detecting fake reviews [11], and detecting cliques in social networks [12]. The task of detecting dense subtensors is generally hard, and the problem of detecting the densest subtensors is mainly an NP-complete or NP-hard problem [13], [14]. Thus, instead of detecting exactly the densest subtensor, approximation is commonly used, and these methods usually have a polynomial time complexity, depending on the dimensions and the size of the tensor. M-Zoom [15] and M-Biz [16] are among the current state-of-the-art dense subtensor detection algorithms. They extend the approaches on dense (sub)graph detection, such as [17], [18], into tensor detection by considering more dimensions for a specific problem to obtain highly accurate algorithms. Further, they utilize a greedy approach to provide local guarantee for the density of the estimated subtensors. M-Zoom and M-Biz are able of maintaining k subtensors at a time. Each time a search is performed, a snapshot of the original tensor is created, and the density of the estimated subtensor in each single search is guaranteed locally on the snapshot. Hence, M-Zoom and M-Biz only provide a density guarantee with respect to the current intermediate tensor rather than the original input tensor. A newer approach, called DenseAlert [19], was developed to detect an incremental dense subtensor for streaming data. Despite its efficiency, however, DenseAlert can estimate only one subtensor at a time, and it can only provide low density guarantee for the estimated subtensor. Hence, it might miss a huge number of other interesting subtensors in the stream.

Extensive studies have shown that DenseAlert, M-Zoom, and M-Biz generally outperform most other tensor decomposition methods, such as [20], [21], in terms of efficiency and accuracy. Nevertheless, an important drawback of these methods is that they can only provide a loose theoretical guarantee for density detection, and that the results and the efficiency are mostly based on heuristics and empirical observations. More importantly, these methods do not provide any analysis of the properties of multiple estimated subtensors. We aim at addressing these drawbacks by proposing a novel technique for estimating several dense subtensors. First, we provide a well-founded theoretical solution to prove that there exist multiple dense subtensors such that their density are guaranteed to be between specific lower and upper bounds. Second, to demonstrate applicability, we introduce a new algorithm, named MUST (MUltiple Estimated SubTensors), which not only supports the aforementioned proof, but also provides an effective method to estimate these dense subtensors.

To give an overview of the differences between our method and the existing approaches, Table I compares the characteristics of MUST against current state-of-the-art algorithms. In summary, the main contributions of this work are as follows:

1) We present a novel theoretical foundation, along with proofs showing that it is possible to maintain multiple subtensors with a density guarantee.
2) We provide a new method that is capable of estimating subtensors with a density guarantee that is higher than those provided by existing methods. Specifically, the new density bound for the dense subtensor is $\frac{1}{N}(1 + \frac{N-1}{min(a,\sqrt{n})})$, while

* Corresponding author

2375-026X/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDE48307.2020.00061

TABLE I: A brief comparison of between existing algorithms and MUST

	Approximation	**Multiple estimation support**	**Single density guarantee**	**Multiple density guarantee**	**Number of guaranteed estimations**	**Bound guarantee**[*]
Goldberg's [13]			✓		1	1
GREEDY [17]	✓		✓		1	$\frac{1}{N}$
GreedyAP [22]	✓		✓		1	$\frac{1}{N}$
M-Zoom [15]	✓	✓	✓		1	$\frac{1}{N}$
DenseAlert [19]	✓		✓		1	$\frac{1}{N}$
M-Biz [16]	✓	✓	✓		1	$\frac{1}{N}$
FrauDar [23]	✓		✓		1	$\frac{1}{N}$
ISG+D-Spot [24]	✓	✓				
MUST	✓	✓	✓	✓	$\boldsymbol{min}(\mathbf{1}\ +\ \mathbf{\frac{n}{2N}}, \mathbf{1}\ \mathbf{+N(N\text{-}1)})$	$\mathbf{\frac{1}{N}}(\mathbf{1} + \frac{\mathbf{N\text{-}1}}{\boldsymbol{min}(\mathbf{a}, \sqrt{\mathbf{n}})})$

[*] N is the number of ways of tensor (with graph, we consider its number of ways is 2 because we can represent a graph in a form of matrix). a is the size of the densest region.

the current widely-used bound is $1/N$. Here, n and a denote the size of the tensor and the densest subtensor, respectively, and N is the number of ways of the tensor.

3) We prove that there exist at least $min(1 + \frac{n}{2N}, 1 + N(N - 1))$ subtensors that have a density greater than a lower bound in the tensor.
4) We perform an extensive experimental evaluation on real-world datasets to demonstrate the efficiency of our solution. The proposed method is up to 6.9 times faster and the resulting subtensors have up to two million times higher density than state-of-the-art methods.

The rest of this paper is organized as follows. Section II describes the preliminaries for the method and the related work. Section III elaborates on the theoretical foundation for providing a new density guarantee of dense subtensors. Section IV presents the solution for detecting multiple dense subtensors with a density guarantee. Section V discusses the evaluation of our method and explains its applicability. Finally, Section VI concludes the paper and outlines the future work. **Reproducibility**: The source code and data used in the paper are publicly available at https://bitbucket.org/duonghuy/mtensor.

II. Background, related work and Notation

The problem of finding the densest subgraphs is generally NP-complete or NP-hard [13], [25]. Due to the complexity of the exact algorithm with which an exponential number of subgraphs must be considered, it is infeasible for large datasets or data streams. Therefore, approximation methods are commonly used for detecting the densest regions [17], [26], [27]. Ashiro et al. [26] proposed an efficient greedy approximation algorithm to find the optimal solution for detecting the densest subgraph in a weighted graph. Their idea is to find a k-vertex subgraph of an n-vertex weighted graph with the maximum weight by iteratively removing a vertex with the minimum weighted-degree in the currently remaining graph, until there are exactly k vertices left. Charikar [17] studied the greedy approach (GREEDY) further, which showed that the approximation can be solved by using linear programming technique. Specifically, the author proposed a greedy 2-approximation for this optimization problem, with which a density guarantee of the dense subgraph is greater than a half of the maximum density in the graph. Many algorithms have later adopted the greedy method with a guarantee on the density of dense subgraphs targeting specific applications, such as fraud detection, event detection, and genetics applications [22], [23], [28], [29]. Common for these works is their use of the greedy 2-approximation to find a dense subgraph.

Inspired by the theoretical solutions in graphs, numerous approaches have been proposed to detect dense subtensors by using the same min-cut mechanism [16], [19]. As mentioned earlier, mining the densest subtensor in a tensor is hard, and an exact mining approach has a polynomial time complexity [13], thus making it infeasible for streaming data or very large datasets. To cope with this, approximate methods/algorithms are commonly used. Among the proposed algorithms, DenseAlert [19], M-Zoom [15], and M-Biz [16] are – because of their effectiveness, flexibility, and efficiency – the current state-of-the-art methods. They are far more faster than other existing algorithms, such as CPD [20], MAF [9], and CrossSpot [30]. DenseAlert, M-Zoom, and M-Biz adapt the theoretical results from dense (sub)graph detection, i.e., [29], [31], [32], to tensor data by considering more dimensions than two. The algorithms utilize a greedy approach to guarantee the density of the estimated subtensors, which has also been shown to yield high accuracy in practice [30]. However, the adopted density guarantee is the same as in the original work, which also applies for the more recent algorithm, *ISG+D-Spot* [24]. This means that with an N-way tensor, the density guarantee is a fraction of the highest density with the number of the tensor's way N. *ISG+D-Spot* converts an input tensor to a form of graph to reduce the number of ways, but it drops all edges having weight less than a threshold. As a result, *ISG+D-Spot* only provides a loose density guarantee.

As discussed in Section I, DenseAlert, M-Zoom and M-Biz can only guarantee low density subtensors. These methods employed the same guarantee as in the original work without any further improvement in the density guarantee. To address the limitations of the previous approaches, we generalize the problem by maintaining multiple dense subtensors, with which we provide a concrete proof to guarantee a higher lower bound density and show that they have a higher density guarantee than the solutions in prior works.

In the following, we present the fundamental preliminaries of the dense subtensor detection problem, based on [16], [19].

Definition 1 (Tensor). *A tensor, T, is a multidimensional array data. The order of T is its number of ways. Given a N-way tensor, on each way, there are multiple spaces, each of which is called a slice.*

Definition 2 (SubTensor). *Given an N-way tensor T, Q is a subtensor of T if it is composed by a subset s of the set of slices S of T, and there is at least one slice on each way of T. Intuitively, Q is the left part of T after we remove all slices in S but not in s.*

Definition 3 (Entry of Tensor). *E is an entry of an N-way (sub)tensor T if it is a subtensor of T and is composed by exactly N slices.*

Definition 4 (Size of a (sub)Tensor). *Given a (sub)Tensor Q, the size of Q is the number of slices that compose Q.*

Definition 5 (Density). *Given a (sub)tensor Q, the density of Q, denoted by $\rho(Q)$, is computed as: $\rho(Q) = \frac{f(Q)}{size\ of\ Q}$, where $f(Q)$ is mass of the (sub)tensor Q, and is computed as the sum of every entry values of Q.*

Definition 6 (Weight of Slice in Tensor). *Given a tensor T. The weight of a slice q in T is denoted by $w_q(T)$, and is defined as the sum of entry values composing by the intersection of T and q.*

Definition 7 (D-Ordering). *An ordering π on a (sub)tensor Q is a D-Ordering, if*

$$\forall q \in Q, q = \underset{p\in Q\wedge\pi^{-1}(p)\geq\pi^{-1}(q)}{argmin} \quad w_p(\pi_q), \qquad (1)$$

where $\pi_q = \{x \in Q | \pi^{-1}(x) \geq \pi^{-1}(q)\}$, $\pi^{-1}(q)$ is to indicate the index of the slice q in π ordering, and $w_p(\pi_q)$ is the weight of p in π_q. Intuitively, the D-Ordering is the order that we pick and remove the minimum slice sum in each step.

The principal of D-Ordering in tensor data is the similar to the min-cut mechanism in dense subgraph detection, like GREEDY [17], [26].

Definition 8 (Mining Dense Subtensor Problem). *Given a tensor T. The problem of dense subtensor detection is to find subtensors $Q \in T$ that maximize the density of Q.*

For readability, the notations used in this paper are summarized in Table II. In the rest of the paper, when specifying a (sub)tensor, we use its name or set of its slices interchangeably.

Example 1. *Let us consider an example of 3-way tensor T as in Figure 1. The value in each cell is the number of visits that a user (mode User) visits a web page (mode Page) on a date (mode Date). The values of hidden cells are all zero. The set of slices of tensor T is {(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2)}. A subtensor Q formed by the following slices {(1,2), (1,3), (2,1), (2,2), (3,1)} is the densest subtensor (the yellow region) and the density of Q is (5+5+7+2)/5 = 3.8.*

TABLE II: Table of notations

Symbols	Description
T, Q	Tensor data T, Q
I_i	The i-*th* dimension of tensor I
$\lvert I_i \rvert$	Number of slices on way I_i of a tensor I
T^*	Densest subtensor T^*
Z, z_0	Zero subtensor Z with zero point z_0
B	Backward subtensor
F	Forward subtensor
n, N	Size and number of ways of tensor data
ρ, ρ^*	Density ρ, highest density ρ^* in tensor
$\rho(Q)$	Density of (sub)tensor Q
π	An ordering π
$Q(\pi, i)$	A subtensor of Q formed by a set of slices $\{p \in Q, \pi^{-1}(p) \geq i\}$
$\rho_\pi(i)$	Density of subtensor $Q(\pi, i)$
q	A slice of a tensor
a	Size of densest subtensor
b	Number of slices in Zero subtensor such that not in densest subtensor
m	Size of Zero subtensor Z, $m = a + b$
$f(Q)$	Mass of the (sub)tensor Q
$w_q(T)$	Weight of slice q in T

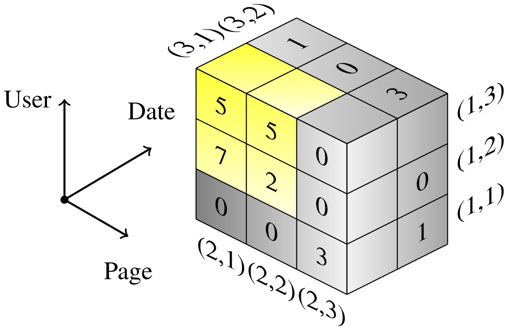


Fig. 1: An example of 3-way tensor.

The problem of mining dense subtensors [16], [19] can be presented and solved as follows. Given a list of n variables $d_\pi(i)$ $(1 \leq i \leq n)$, where $d_\pi(i)$ is calculated during the construction of D-Ordering. Its value at each time is picked by the minimum slice sum of the input (sub)tensor. Then, a *Find-Slices*() function finds the index $i^* = \underset{1\leq i\leq n}{\text{argmax}}\ \rho_\pi(i)$, which is the location to guarantee a subtensor with a density greater than the lower bound. *Find-Slices()*, shown in Algorithm 1, is a function that was originally defined in [15], [16], [19], which is a principal function for estimating a subtensor, such that its density is greater than the lower bound. The density of an estimated subtensor is guaranteed as follows.

Algorithm 1 Find-Slices

```
Require: A D-Ordering $\pi$ on a set of slices Q
Ensure: An estimated subtensor S
$S \leftarrow \emptyset$, $m \leftarrow 0$
$\rho_{max} \leftarrow -\infty$, $q_{max} \leftarrow 0$
for ($j \leftarrow |Q|..1$) do
    $q \leftarrow \pi(j)$, $S \leftarrow S \cup q$
    $m \leftarrow m + d_\pi(q)$
    if $m/|S| > \rho_{max}$ then
        $\rho_{max} \leftarrow m/|S|$
        $q_{max} \leftarrow q$
    end if
end for
return $Q(\pi, \pi^{-1}(q_{max}))$
```

Theorem 1 (Density Guarantee) [16], [19]**).** *The density of the subtensor returned by the Algorithm 1 is greater than or equal to $\frac{1}{N}\rho^*$, where ρ^* is the highest density in the input tensor.*

Proof. The proof of this theorem was provided in [16], [19]. For convenience, we recall their proof as follows. Let $q^* \in T^*$ be the slice such that $\pi^{-1}(q^*) \leq \pi^{-1}(q), \forall q \in T^*$. This means that q^* is the slice in the densest subtensor having the smallest index in π. Therefore $\rho_\pi(i^*) \geq \rho_\pi(\pi^{-1}(q^*)) \geq \frac{1}{N}\rho^*$. □

III. The New Density Guarantee of Subtensor

As can be inferred from the discussion above, the basic principle underlying DenseAlert, M-Zoom, and M-Biz is Theorem 1. It is worth noting that this theorem guarantees the lower bound of the density on only one estimated subtensor from an input tensor. To the best of our knowledge, none of existing approximation approaches provides a better density guarantee than GREEDY. Based on this, we can raise the following questions: (1) Can this lower bound be guaranteed higher? (2) Are there many subtensors having density greater than the lower bound? (3) Can we estimate these subtensors?

In this section, we answer question (1) by providing a proof for a new higher density guarantee. Questions (2) and (3) will be answered in the next section by providing a novel theoretically sound solution to guarantee the estimation of multiple dense subtensors that have higher density than the lower bound.

A. A New Bound of Density Guarantee

We prove that the estimated subtensors provided by the proposed methods have a higher bound than in the state-of-the-art solutions.

In [16], [19], the authors showed that the ,density of the subtensor $\rho_\pi(\pi^{-1}(q^*)) \geq \frac{1}{N}\rho^*$, hence satisfying Theorem 1. A sensible question is: Can we estimate several subtensors with a higher density guarantee than the state-of-the-art algorithms?

In the following subsections, we introduce our new solution to improve the guarantee in the aforementioned *Find-slices()* function and show how a density with higher lower bound than that in [16], [19] can be provided. We present several theorems and properties to support our solution to estimate multiple dense subtensors.

Definition 9 (Zero Subtensor)**.** *Given a tensor T, T^* is the densest subtensor in T with density ρ^*, π is a D-ordering on T, and $z_0 = \underset{q \in T^*}{min}\ \pi^{-1}(q)$ is the smallest indices in D-Ordering π of all slices in T^*. A subtensor called Zero Subtensor of T on π, denoted as $Z = T(\pi, z_0)$, and z_0 is called zero point.*

Theorem 2 (Lower Bound Density of the Estimated Subtensor)**.** *Given an N-way tensor T, and a D-ordering π on T. Let Z and z_0 be a Zero Subtensor and a zero point, respectively. Then, there exists a number $b \geq 0$ such that the density of the estimated subtensor Z is not less than $\frac{Na+b}{N(a+b)}\rho^*$, where a and ρ^* are the size and density of the densest subtensor T^*.*

Proof. We denote $w_0 = w_{\pi(z_0)}(Z)$. Further, note that because T^* is the densest subtensor. Then,

$$\forall q \in T^*, w_q(T^*) \geq \rho^* \Rightarrow w_0 \geq \rho^*.$$

Due to the characteristic of D-Ordering, we have

$$w_q(Z) \geq w_{\pi(z_0)}(Z) = w_0, \forall q \in Z.$$

Consider a way I_i among the N ways of the tensor T. Then,

$$f(Z) = \sum_{q \in T^* \wedge q \in I_i} w_q(Z) + \sum_{q \notin T^* \wedge q \in I_i} w_q(Z).$$

Furthermore, regarding the way we choose Z, we have

$$T^* \subseteq Z \Rightarrow \sum_{q \in T^* \wedge q \in I_i} w_q(Z) \geq \sum_{q \in T^* \wedge q \in I_i} w_q(T^*) = f(T^*).$$

Therefore,

$$f(Z) \geq f(T^*) + \sum_{q \notin T^* \wedge q \in I_i} w_q(Z) \geq f(T^*) + b_{I_i} w_0, \quad (2)$$

where b_{I_i} is the number of slices in Z on dimension I_i that are not in T^*. Let $b = \sum_{i=1}^{N} b_{I_i}$. Applying Eq. 2 on N ways, we get

$$\begin{aligned} & Nf(Z) && \geq Nf(T^*) + w_0 \sum b_{I_i} \\ \Rightarrow & N(a+b)\rho(Z) && \geq Na\rho^* + w_0 b \\ \Rightarrow & N(a+b)\rho(Z) && \geq Na\rho^* + b\rho^* \\ \Rightarrow & \rho(Z) && \geq \frac{Na+b}{N(a+b)}\rho^* \end{aligned}$$ □

The equality happens when $b = 0$ or in the simple case when $N = 1$. However, if these conditions hold, the Zero Subtensor becomes the densest subtensor T^*. In the next paragraphs, we consider the higher order problem of tensor with order $N > 1$.

Property 1. *The lower bound density in Theorem 2 is greater than $\frac{1}{N}$ of the highest density and this bound is within $[\frac{1}{N}(1 + \frac{a(N-1)}{n}), 1]$.*

Proof. Let Z be the fraction of the density of the estimated subtensor, and R denote densest subtensor. We have the following properties about the lower bound fraction:

1) In the simplest case, when $N = 1$, the lower bound rate values both in the previous proof and in this proof are 1. This means that the estimated subtensor Z is the densest subtensor, with the highest density value. Otherwise, $R \geq \frac{Na+b}{N(a+b)} = \frac{a+b}{N(a+b)} + \frac{(N-1)a}{N(a+b)} > \frac{1}{N}, \forall N > 1$.

   Moreover, since the size of Z is not greater than n, we have $R \geq \frac{1}{N}(1 + \frac{(N-1)a}{(a+b)}) \geq \frac{1}{N}(1 + \frac{a(N-1)}{n})$.
2) In conclusion, we have the following boundary of the density of estimated Zero Subtensor, Z:
   $\rho(Z) = \begin{cases} \rho^*, & \text{if } N = 1 \vee b = 0 \\ \frac{1}{N}(1 + \frac{a(N-1)}{n})\rho^*, & \text{if } a + b = n. \end{cases}$

In an ideal case, when the value of b goes to zero, the estimated subtensor becomes the densest subtensor, and its density can be guaranteed to be the highest. □

B. A New Higher Density Guarantee

In this subsection, we provide a new proof to give a new higher density guarantee of dense subtensor.

Theorem 3 (Upper Bound of the Min-Cut Value in Tensor)**.** *Given an N-way tensor T with size n, and a slice q is chosen for the minimum cut, such that the weight of q in T is minimum. Then, the weight of q in T satisfies the following inequality:*

$$w_q(T) \leq N\rho(T) \tag{3}$$

Proof. Because q is a slice having the minimum cut, we have $w_q(T) \leq w_p(T), \forall p \in T$. Summing all the slices in the tensor gives

$$|T|w_q(T) \leq \sum_{p \in T} w_p(T) = Nf(T)$$
$$\Rightarrow w_q(T) \leq \frac{Nf(T)}{|T|} = N\rho(T) \qquad □$$

Let $T_i (1 \leq i \leq a)$ be the subtensor right before we remove i-th slice of T^*, and q_i be the slice of T^* having the minimum cut w_i at the step of processing T_i. Since the size of the densest T^* is a, we have a indexes from 1 to a. Note that T_1 is the Zero subtensor Z. Further, let M_{I_i} denote the index of the last slice in way I_i of T^* that will be removed. Then, we have following property:

Property 2 (Upper Bound of the Last Removed Index)**.** *The minimum index of all $M_{I_i}, 1 \leq i \leq N$, denoted by M, is not greater than $(a - N + 1)$, i.e., $M = min(M_{I_i}) \leq a - N + 1$.*

Proof. Let M_{I_i}, M_{I_j} be the indexes of the last removed slices of the two ways I_i and I_j. Further, assume that the difference between M_{I_i}, M_{I_j} is $\Delta(M_{I_i}, M_{I_j}) = |M_{I_i} - M_{I_j}| \geq 1$, and that we have N numbers (N ways) and the maximum (the last index) is a. Then, we get

$$max(M_{I_i}) - min(M_{I_i}) \geq N - 1$$
$$\Rightarrow M = min(M_{I_i}) \leq a - N + 1 \qquad □$$

Theorem 4. *The sum of min-cut of all slices from index 1 to M is greater than the mass of the densest subtensor T^*:*

$$\sum_{i=1}^{M} w_{q_i}(T_i) \geq f(T^*) \tag{4}$$

Proof. Let E be any entry of the densest subtensor T^* and E is composed by the intersection of N slices, $q_{I_x} (1 \leq x \leq N)$, q_{I_x} is on the way I_x.

Assume that the first removed index of all the slices composing E is at index i. Since this index cannot be greater than M, the entry E is in T_i, and its value is counted in $w_{q_x}(T_x)$. Therefore, we have: $\sum_{i=1}^{M} w_{q_i}(T_i) \geq f(T^*)$ □

Let ρ_{max} be the maximum density among all subtensors $T_i, (i \leq i \leq M)$. According to Theorems 3 and 4, we have

$$f(T^*) \leq \sum_{i=1}^{M} w_{q_i}(T_i) \leq \sum_{i=1}^{M} N\rho(T_i) \leq MN\rho_{max} \tag{5}$$
$$\Rightarrow a\rho^* \leq N(a - N + 1)\rho_{max}$$
$$\Rightarrow \rho_{max} \geq \frac{\rho^*}{N}\frac{a}{a - N + 1}. \tag{6}$$

Theorem 5 (Better Density Guarantee of Dense Subtensor)**.** *The density guarantee of dense subtensor mining by min-cut mechanism is greater than $\frac{1}{N}(1 + \frac{N-1}{min(a,\sqrt{n})})\rho^*$.*

Proof. According to Theorem 2 and Property 1, we have

$$\rho_{max} \geq \rho(T_1) \geq \frac{1}{N}(1 + \frac{a(N-1)}{n})\rho^* \tag{7}$$

Furthermore, by Inequation 6, we also have

$$\rho_{max} \geq \frac{\rho^*}{N}\frac{a}{a - N + 1} \geq \frac{1}{N}(1 + \frac{N-1}{a})\rho^* \tag{8}$$

By combining Eq. 7 and Eq. 8, we get

$$\rho_{max} \geq \frac{1}{N}(1 + \frac{1}{2}(\frac{a(N-1)}{n} + \frac{N-1}{a}))\rho^*$$
$$\Rightarrow \rho_{max} \geq \frac{1}{N}(1 + \frac{N-1}{\sqrt{n}})\rho^*$$

Note that since $\rho_{max} \geq \frac{1}{N}(1 + \frac{N-1}{a})\rho^*$, we finally have

$$\rho_{max} \geq \frac{1}{N}(1 + \frac{N-1}{min(a,\sqrt{n})})\rho^* \qquad □$$

IV. The Solution For Multiple Dense Subtensors

As shown in Theorem 2, $\rho(Z) \geq \frac{Na+b}{N(a+b)}\rho^*$, where $Z = T(\pi, z_0)$ is the Zero subtensor. As discussed before, the state-of-the-art algorithm, DenseAlert, can estimate only one subtensor at a time, and a density guarantee is low, i.e., $\frac{1}{N}$ of the highest density. M-Zoom (or M-Biz) is, on the other hand, able of maintaining k subtensors at a time by repeatedly calling the *Find-Slices()* function k times, with the input (sub)tensor being a snapshot of the whole tensor (i.e., the original one). Recall, however, that such processing cannot guarantee any density boundary of the estimated subtensors with respect to the original input tensor. Therefore, the estimated density of the subtensors is very low. With this, an important question is: How many subtensors in n subtensors of D-ordering as in Algorithm 1 having density greater than a lower bound density and what is the guarantee on the lower bound density with respect to highest density? This section answers this question.

A. Forward Subtensor from Zero Point

Again, given a tensor T, T^* is the densest subtensor in T with density ρ^*. π is a D-ordering on T, and the zero point $z_0 = \underset{q \in T^*}{min}\ \pi^{-1}(q)$ is the smallest indices in π among all slices in T^* (cf. Definition 9).

Definition 10 (Forward Subtensor)**.** *A subtensor is called i-Forward subtensor in T on π, denoted by F_i, if $F_i = T(\pi, z_0 - i), 0 \leq i < z_0$.*

Let us consider an i-forward subtensor $F_i = T(\pi, i), i < z_0$. Because $i < z_0$, $Z \subseteq F_i$. This means that $f(F_i) \geq f(Z)$. As a result of Theorem 2, we have the following:

$$\begin{aligned}
Nf(Z) \quad & \geq (Na+b)\rho^* \\
\Rightarrow (Na+b)\rho^* \quad & \leq Nf(Z) \leq Nf(F_i) \\
\Rightarrow (Na+b)\rho^* \quad & \leq N(a+b+i)\rho(F_i) \\
\Rightarrow \rho(F_i) \quad & \geq \frac{Na+b}{N(a+b+i)}\rho^*.
\end{aligned}$$

From the above inequality, we get the following theorem.

Theorem 6. *The density of every i-Forward subtensor $F_i = T(\pi, i)$, where $i \leq N \times (N-1)$ is greater than or equal to $1/N$ of the highest density in T, ρ^*.*

Proof. From the above inequality, $\rho(F_i) \geq \frac{Na+b}{N(a+b+i)}\rho^*$. If we have $i \leq N(N-1)$, then

$$\begin{aligned}
\Rightarrow a+b+i \quad & \leq a+b+N(N-1) \\
\Rightarrow \frac{Na+b}{N(a+b+i)}\rho^* \quad & \geq \frac{Na+b}{N(a+b+N(N-1))}\rho^* \\
\Rightarrow \frac{Na+b}{N(a+b+i)}\rho^* \quad & \geq \frac{a+b+a(N-1)}{N(a+b+N(N-1))}\rho^* \\
\Rightarrow \frac{Na+b}{N(a+b+i)}\rho^* \quad & \geq \frac{a+b+N(N-1)}{N(a+b+N(N-1))}\rho^* \\
\Rightarrow \rho(F_i) \quad & \geq \frac{Na+b}{N(a+b+i)}\rho^* \geq \frac{1}{N}\rho^*
\end{aligned}$$ □

Property 3. *Among n subtensors $T(\pi, i), 1 \leq i \leq n$, there is at least min$(z_0, 1 + N(N-1))$ subtensors having a density greater than $\frac{1}{N}$ of the densest subtensor in T.*

Proof. According to Theorem 6, there is at least $min(z_0, 1 + N(N-1))$ forward subtensors that have density greater than $\frac{1}{N}$ of the highest density. □

B. Backward Subtensor from Zero Point

We have considered subtensors formed by adding more slices to Z. Next, we continue investigating the density of the subtensors by sequentially removing slices in Z.

Definition 11 (Backward Subtensor)**.** *A subtensor is called i-Backward subtensor in T on π, denoted by B_i, if $B_i = T(\pi, z_0 + i), i \geq 0$.*

Let us consider an i-backward subtensor B_i. We show that its density is also greater than the lower bound density.

Property 4. *The density of the 1-Backward Subtensor, B_1 is greater than or equal to $\frac{1}{N}\rho^*$.*

Proof. Due to the limitation of space, we omit the proof and provide it in an extension supplement upon request. □

Theorem 7. *Let B_k denote the k-Backward subtensor, $B_k = T(\pi, z_0 + k)$. Density of B_k is greater than or equal to $1/N$ of the highest density in $T, \forall k \leq \frac{b}{N}$.*

Proof. Note that $f(B_i) = f(B_{i+1}) + w_{\pi(z_0+i)}(B_i)$. Let $B_0 = Z$, and in the following we let $w_i(B_i) = w_{\pi(z_0+i)}(B_i)$ for short. Then, we have

$$\begin{aligned}
Kf(Z) &= K(f(B_1) + w_0(B_0)) \\
&= K(f(B_2) + w_0(B_0) + w_1(B_1)) \\
&= Kf(B_k) + K\sum_{i=0}^{k-1} w_i(B_i).
\end{aligned}$$

Because $T^* \subseteq Z$, then:

$$Kf(Z) \geq Kf(T^*) + \sum_{q \in Z \wedge q \notin T^*} w_q(Z), \tag{9}$$

By substitution, we get

$$Kf(B_k) + K\sum_{i=0}^{k-1} w_i(B_i) \geq Kf(T^*) + \sum_{q \in Z \wedge q \notin T^*} w_q(Z)$$

$$\Rightarrow Kf(B_k) \geq Kf(T^*) + \sum_{q \in Z \wedge q \notin T^*} w_q(Z) - K\sum_{i=0}^{k-1} w_i(B_i).$$

We denote the set $Q = \{q|\ q \in Z \wedge q \notin T^*\}$ by $\{q_1, q_2, \ldots, q_b\}$. Note that $B_i \subseteq Z$. Thus $\forall j, i, w_{q_j}(Z) \geq w_{q_j}(B_i) \geq w_i(B_i)$, and $w_{\pi(z_0)}(Z) \geq w_{\pi(z_0)}(T^*) \geq \rho^*$.

On the other hand, we have the condition of k: $b - k \times K \geq b - k \times N \geq 0$. In conclusion, this gives the following inequality:

$$\begin{aligned}
Kf(B_k) - Kf(T^*) &\geq \sum_{q \in Z \wedge q \notin T^*} w_q(Z) - K\sum_{i=0}^{k-1} w_i(B_i) \\
&\geq \sum_{i=0}^{k-1}\sum_{j=1}^{K} w_{q_{(i \times K + j)}}(Z) - Kw_i(B_i) + \sum_{i=k \times K+1}^{b} w_{q_i}(Z) \\
&\geq (b - k \times K) \times E_{\pi(z_0)}(Z) \\
&\geq (b - k \times K)\rho^*
\end{aligned}$$

$$\begin{aligned}
&\Rightarrow K\rho(B_k)(a+b-k) \geq Ka\rho^* + (b - k \times K)\rho^* \\
&\Rightarrow \rho(B_k) \geq \frac{Ka + b - k \times K}{K(a+b-k)}\rho^* \\
&\Rightarrow \rho(B_k) \geq \frac{K(a-k)+b}{K(a+b-k)}\rho^* \\
&\Rightarrow \rho(B_k) \geq \frac{1}{K}\rho^* \geq \frac{1}{N}\rho^*.
\end{aligned}$$ □

Theorem 8. *Assume that the size of the Zero subtensor Z, (a+b), is sufficiently big. Let B_k denote the k-Backward subtensor.*

The density of B_k is greater than or equal to $1/N$ of the highest density in $T, \forall k \leq min(\frac{a}{N}, \frac{(a+b)(N-1)}{N^2})$.

Proof. Assume I_x is the way that has the smallest number of slices in T^*, with a number of slices s. Then, $s \leq a/N$.

Let $Q = \{q \in Z\} = \{q_1, \ldots, q_s, \ldots, q_a, \ldots, q_{a+b}\}$, denote the set of slices in Z, and $(a+b)$ be the size of the Zero subtensor.

Let B_k be a k-Backward Subtensor of T, with $1 \leq k \leq \frac{(a+b)}{N}$. Then,

$$Nf(Z) = \sum_{i=1}^{s} w_{q_i}(Z) + \sum_{i=s+1}^{a+b} w_{q_i}(Z) \geq f(T^*) + \sum_{i=s+1}^{a+b} w_{q_i}(Z).$$

Because $Nf(Z) = N(f(B_k) + \sum_{i=0}^{k-1} w_i(B_i))$, the above inequality can be rewritten as

$$\Rightarrow N(f(B_k) + \sum_{i=0}^{k-1} w_i(B_i)) \geq f(T^*) + \sum_{i=s+1}^{a+b} w_{q_i}(Z).$$

The subtensor B_i is a backward subtensor of Z by removing i slices in Z, i.e., $B_i \subseteq Z$ and $\forall j, i, E_{q_j}(Z) \geq E_{q_j}(B_i) \geq E_{\pi(z_0+i)}(B_i)$. Hence,

$$\begin{aligned}
Nf(B_k) &\geq f(T^*) + \sum_{i=s+1}^{a+b} w_{q_i}(Z) - N\sum_{i=0}^{k-1} w_i(B_i) \\
&= f(T^*) + \sum_{i=0}^{k-1}\sum_{j=1}^{N} w_{q_{(s+i\times N+j)}}(Z) - Nw_i(B_i) \\
&+ \sum_{i=s+k\times N+1}^{a+b} w_{q_i}(Z) \\
&\geq f(T^*) + (a+b-kN-s)w_{\pi(z_0)}(Z).
\end{aligned}$$

Because

$$\begin{aligned}
a+b-kN-s &\geq a+b-kN-\frac{a}{N} \\
&\geq \frac{(a+b)(N-1)+b}{N} - kN \\
&\geq 0, \forall k \leq \frac{(a+b)(N-1)}{N^2},
\end{aligned}$$

we have

$$\begin{aligned}
&Nf(B_k) \geq a\rho^* + (a+b-kN-s)\rho^* \\
&Nf(B_k) \geq (2a+b-kN-s)\rho^* \\
&\Rightarrow \rho(B_k) \geq \frac{(2a+b-kN-s)}{N(a+b-k)}\rho^* \\
&\Rightarrow \rho(B_k) \geq \frac{1}{N}\frac{2a+b-kN-s}{a+b-k}\rho^* \\
&\Rightarrow \rho(B_k) \geq \frac{1}{N}\frac{(a+b-k)+(a-k(N-1)-a/N)}{a+b-k}\rho^* \\
&\Rightarrow \rho(B_k) \geq \frac{1}{N}(1+\frac{(a-kN)(N-1)}{N(a+b-k)})\rho^* \\
&\Rightarrow \rho(B_k) \geq \frac{\rho^*}{N}, \forall k \leq \frac{a}{N}.
\end{aligned}$$

□

C. Multiple Dense Subtensors with High Density Guarantee

In this subsection, we show that there exist multiple subtensors that have density values greater than a lower bound in the tensor.

Theorem 9. *Given an N-way tensor T with size $n >> N$, an order π is a D-Ordering on T, and Algorithm 1 processes $m = (n-N)$ subtensors. Then, there are at least $min(1+\frac{n}{2N}, 1+N(N-1))$ subtensors among m subtensors, such that they have density greater than $1/N$ of the highest density subtensor in T.*

Proof. Let Z denote the Zero subtensor of T on π by Algorithm 1, and the zero index is z_0, such that $N \leq n - z_0$. Then, we have the following:

1) By Theorem 6, there are at least $min(N(N-1), z_0)$ forward subtensors $F_1, F_2, \ldots,$ having density higher than $\frac{1}{N}\rho^*$.
2) By Theorems 7-8, there are backward subtensors $B_1, B_2, \ldots,$ having density higher than $\frac{1}{N}\rho^*$. The principle of the number of backward subtensors having density greater than $\frac{1}{N}$ of the highest density is as follows:

$$\begin{cases} \frac{b}{N}, & \text{by Theorem 7.} \\ min(\frac{a}{N}, \frac{(a+b)(N-1)}{N^2}), & \text{by Theorem 8.} \end{cases} \tag{10}$$

From Eq. 10, there is at least $max(\frac{b}{N}, min(\frac{a}{N}, \frac{(a+b)(N-1)}{N^2}))$ backward subtensors having density greater than the lower bound.

If $\frac{a}{N} \leq \frac{(a+b)(N-1)}{N^2}$, then number of backward subtensors having density greater than the lower bound is at least $max(\frac{a}{N}, \frac{b}{N}) \geq \frac{a+b}{2N}$.

Otherwise, we have

$$min(\frac{a}{N}, \frac{(a+b)(N-1)}{N^2}) = \frac{(a+b)(N-1)}{N^2} \geq \frac{a+b}{2N}.$$

Hence, the number of backward subtensors is at least $\frac{a+b}{2N}$. Further, if we combine this with the number of forward subtensors, then there is at least $min(1+\frac{n}{2N}, 1+N(N-1))$ subtensors in the tensor having density greater than a lower bound. This can be proved as follows.

According to Theorem 8, we have the number of backward subtensors having density greater than the lower bound, denoted by bw, and $bw \geq \frac{(a+b)}{2N}$. By Theorem 6, we have the number of subtensors having density greater than the lower bound, we denote this by fw, and $fw \geq min(N(N-1), z_0)$.

If $z_0 \geq N(N-1)$, then the number of subtensors that have density values greater than a lower bound is $1 + fw + bw \geq 1+N(N-1)$, where 1 is used to account for the zero subtensor.

Otherwise (i.e., $z_0 \leq N(N-1)$), we have $a + b + z_0 = n$, and we get

$$\begin{aligned}
&1 + fw + bw \geq 1 + \frac{(a+b)}{2N} + z_0 \\
\Rightarrow &1 + fw + bw \geq 1 + \frac{(n-z_0)}{2N} + z_0 \\
\Rightarrow &1 + fw + bw \geq 1 + \frac{n}{2N} + \frac{z_0(2N-1)}{2N} \\
\Rightarrow &1 + fw + bw \geq 1 + \frac{n}{2N}.
\end{aligned}$$

This gives that the number of subtensors having density values greater than the lower bound is $1+fw+bw \geq min(1+\frac{n}{2N}, 1+N(N-1))$.

If $(a+b) \leq n - N(N-1)$, then we have at least $N(N-1)$ forward subtensors having density greater than $\frac{1}{N}$ of the highest density.

Otherwise, if $n >> N$ such that

$$\begin{aligned}
&(a+b) \geq n - N(N-1) \geq 2N^3 \\
\Rightarrow \text{ then we get } &\frac{(a+b)}{2N} \geq N(N-1).
\end{aligned}$$

In conclusion, we have at least $N(N-1)$ backward subtensors, each having density greater than $\frac{1}{N}$ of the highest density. By adding the zero subtensors, we have at least $(1+N(N-1))$ subtensors having density greater than $\frac{1}{N}$ of the highest density each. □

Our approach described above can be employed to improve the state-of-the-art algorithms on estimating multiple dense subtensors using Algorithm 2.

Algorithm 2 Multiple Estimated Subtensors

```
Require: A D-Ordering $\pi$ on a set of slices Q of tensor $T$
Ensure: Multiple estimated subtensors with guarantee on density
Initialization()        ▷ density measure $\rho$, build tensor
$TS \leftarrow \emptyset$,  $S \leftarrow \emptyset$
Number of estimated subtensors: $mul \leftarrow 0$
$mul \leftarrow min(1 + \frac{n}{2N}, 1 + N(N-1))$
for ($j \leftarrow |Q|..1$) do
    $q \leftarrow \pi(j)$
    $S \leftarrow S \cup q$
    TS.add $(S, \rho(S))$
end for
Sort TS by descending order of density
return top-$mul$ subtensors having highest density in TS
```

Complexity discussion. In order to estimate k dense subtensors, the complexity of M-Zoom and M-Biz are high. The worst-case time complexity of M-Zoom and M-Biz is $O(kNn\log n)$ [16]. Its complexity increases linearly with respect to the number of estimated subtensors, k.

Focusing on the proposed solution, MUST, the complexity includes the cost of D-Ordering, which is $O(Nn\log n)$, and the cost of executing Algorithm 2, which utilizes Google Guava ordering[1], is $O(n\log n)$, in the worst case. In total, the complexity MUST is $O(Nn\log n)$, which does not depend on the number of estimated subtensors k.

V. Experimental results

In this section, we present the results from our experimental evaluation, where we evaluate the performance of our proposed method in terms of both the execution time (i.e., efficiency) and the accuracy of the density of the estimated subtensors (i.e., effectiveness).

A. Experimental Setup

We used four widely-used density measures in our experiments: arithmetic average mass (ρ_a) [17]; geometric average mass (ρ_g) [17]; entry surplus (ρ_e) [29], with which the surplus parameter α was set to 1 as default; and suspiciousness (ρ_s) [30]. Note that in M-Zoom (M-Biz), Dense-Alert, and in this work, the density measure used for the proof of guarantee is arithmetic average mass. Nevertheless, the only difference among the density measures is the choice of coefficients. Hence, we can utilize the same proof for other mass measures to get similar results. In this paper, we specifically provide theoretical proofs for density guarantee of dense subtensors. Here, it is worth noting that we can easily extend and apply our proofs of higher density for dense subgraphs, as well.

We implemented our approach based on the implementation used in the previous approaches [15], [16], [19]. We compared the performance of the proposed solution with the state-of-the-art algorithms, M-Zoom and M-Biz (where M-Zoom was used as the seed-subtensor). To do this, in our experiments, we run the algorithms using M-Zoom, M-Biz, and MUST to get top 10 subtensors that have the highest density. We carried out all the experiments on a computer running Windows 10 as operating system, having a 64-bit Intel i7 2.6 GHz processor and 16GB of RAM. All the algorithms were implemented in Java, including M-Zoom and M-Biz, the source codes for which were provided by the authors[2].

B. Datasets

In order to evaluate the performance of the proposed solution and compare it with the state-of-the-art algorithms, we used the following 10 real-world datasets:

- *Air Force*, which contains TCP dump data for a typical U.S. Air Force LAN. The dataset was modified from the KDD Cup 1999 Data and was provided by Shin et al. [16].
- *Android*, which contains product reviews and rating metadata of applications for Android from Amazon [33].
- *Darpa*, which is a dataset collected by MIT Lincoln Lab to evaluate the performance of intrusion detection systems (IDSs) in cooperation with DARPA [34].
- *Enron Emails*, provided by the Federal Energy Regulatory Commission to analyze the social network of employees during its investigation of fraud detection and counter terrorism.

[1] https://opensource.google.com/projects/guava

[2] https://github.com/kijungs/mzoom

TABLE III: Summary of the real-world datasets used in the experiments

Dataset	Instance Structure	Entry	Size	#Instances	#Ways	Data Type
Air Force	(protocol, service, flag, s-bytes, d-bytes, counts, srv-counts, #connects)	#connects	3 × 70 × 11 × 7,195 × 21,493 × 512 × 512	4,898,431	7	TCP Dumps
Android	(user, application, score, date, rate)	rate	1,323,884 × 61,275 × 5 × 1,282	2,638,173	4	Ratings
Darpa	(s-ip, d-ip, date, #connects)	#connects	9,484 × 23,398 × 46,574	4,554,344	3	TCP Dumps
Enron Emails	(sender, receive, word, date, count)	count	6,066 × 5,699 × 244,268 × 1,176	54,202,099	4	Text, Social Network
Enwiki	(user, page, time, #revisions)	revisions	4,135,167 × 14,449,530 × 132,079	57,713,231	3	Activity Logs
Kowiki	(user, page, time, #revisions)	revisions	662,370 × 1,918,566 × 125,557	21,680,118	3	Activity Logs
LBNL Network	(s-ip, s-port, d-ip, d-port,date, packet)	packet	1,605 × 4,198 × 1,631 × 4,209 × 868,131	1,698,825	5	Network
NIPS Pubs	(paper, author, word, year, count)	count	2,482 × 2,862 × 14,036 × 17	3,101,609	4	Text, Academic
StackO	(user, post, favourite, time)	favourite	545,195 × 96,678 × 1,154	1,301,942	3	Activity Logs
YouTube	(user, user, connected, date)	connected	1,221,280 × 3,220,409 × 203	9,375,374	3	Social Network

- *Enwiki* and *Kowiki* provided by Wikipedia[3]. Enwiki and Kowiki are metadata representing the number of user revisions on Wikipedia pages at given times (in hour) in English Wikipedia and Korean Wikipedia, respectively.
- *LBNL-Network*, which consists of internal network traffic captured by Lawrence Berkeley National Laboratory and ICSI [35]. Each instance contains the packet size that a source (ip, port) sends to a destination (ip, port) at a time.
- *NIPS Pubs*, which contains papers published in NIPS[4] from 1987 to 2003 [36].
- *StackO*, which represents data of users and posts on the Stack Overflow. Each instance contains the information of a user marked a post as favorite at a timestamp [37].
- *YouTube*, which consists of the friendship connections between YouTube users [38].

We selected these datasets because of their diversity, and because they are widely used as benchmark datasets in the literature [16], [19]. A more detailed information about the datasets are listed in Table III.

C. Density of the Estimated Subtensors

Figure 2 shows the density of the estimated subtensors obtained with M-Zoom, M-Biz, and MUST. In the figure, we plot the average (AVG) and the low boundary (BOUND) density of the top-10 estimated subtensors. As shown, although the estimated subtensors found by M-Zoom and M-Biz have guarantee locally on the snapshot, the density of the subtensors drops dramatically with respect to the increasing number of the estimated subtensors, k. On all the datasets, the average and the bound density of the estimated subtensors with MUST are much higher than those obtained with M-Zoom and M-Biz in all density measures. MUST also outperforms M-Zoom and M-Biz on density accuracy of estimated subtensors, focusing on both the average and boundary of density of the top ten estimated subtensors.

In particular, on the Air Force dataset, the average density with MUST is up to 546% higher than with M-Zoom and M-Biz, using the arithmetic average mass measure, and more than 891% higher on the Darpa dataset using entry surplus measure. In terms of lower bound of density of the estimated subtensors, there is a huge gap between the proposed algorithm and the baseline algorithms. For instance, on the Air Force dataset, the lower bound of density of the estimated subtensors with MUST are more than 360 times and two million times bigger than with both baseline algorithms, when applying arithmetic average mass and entry surplus measure, respectively.

D. Diversity and Overlap Analysis

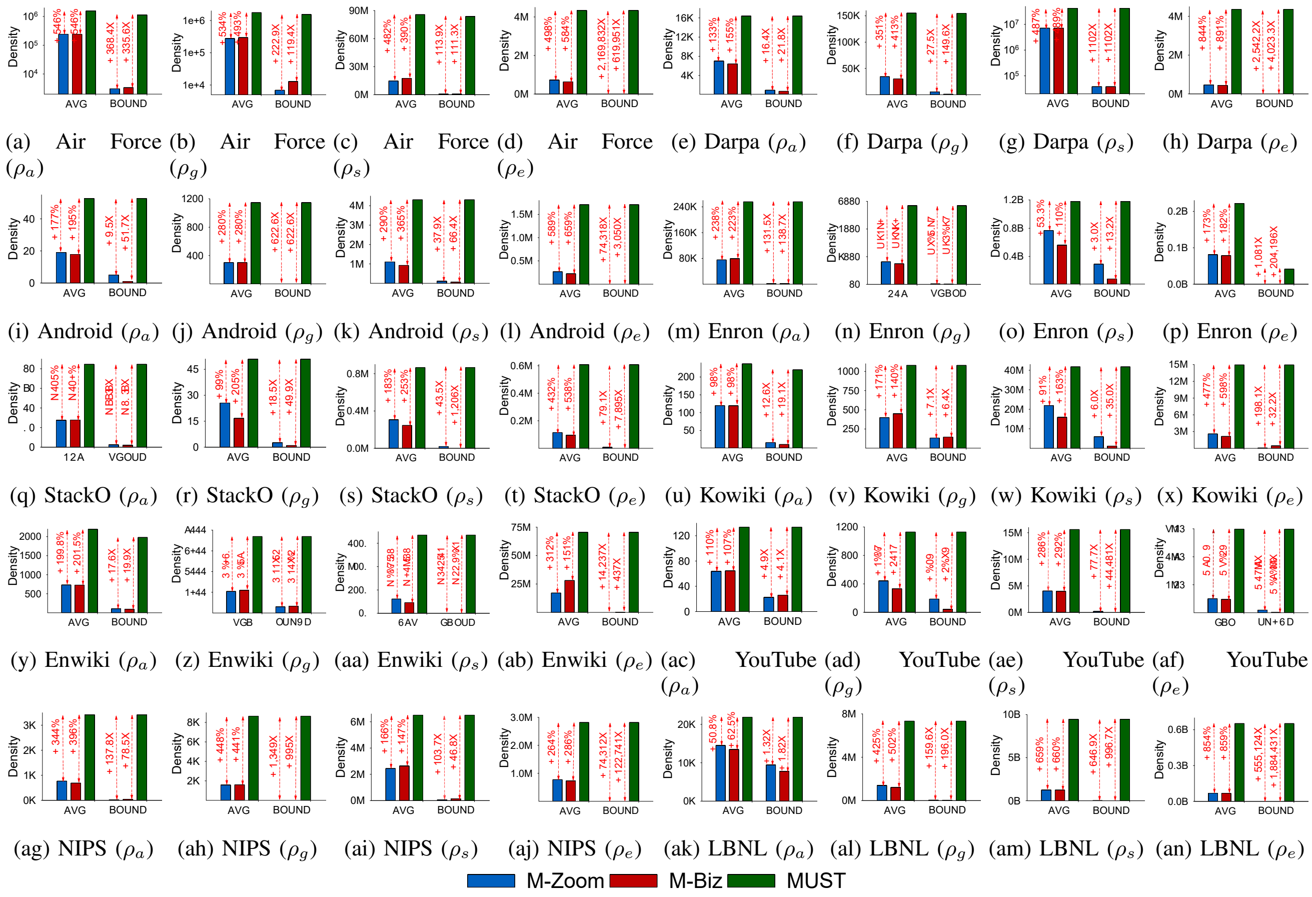
An important difference between MUST and other approaches is its ability to estimate multiple subtensors. Hence, important aspects worth evaluating and discussing are (1) how much difference it is between estimated subtensors, and (2) the fractions of overlap among the detected subtensors. Intuitively, MUST sequentially removes one slice which has a minimum slice weight at a time. Finally, the algorithm picks the top k highest densities among estimated subtensors.

In this subsection, we evaluate the diversity of the top three estimated subtensors by MUST, M-Zoom on the Enwiki, Kowiki, and Air Force datasets to analyze the overlap fractions of subtensors. We use arithmetic average mass (ρ_a) as the density metric and the used diversity measure is the same as in [15]. The diversity of two subtensors is the average dissimilarity between pairs of them. Here, we chose the Enwiki, Kowiki, and Air Force datasets because they contain anomaly and fraud events, and that they are commonly used for this type of benchmark [15], [19].

Table IV shows the diversity of the top three estimated subtensors by MUST and M-Zoom. We observe that the obtained diversities by MUST are 36.2%, 37.2%, and 20.8% on Enwiki, Kowiki, and Air Force, respectively. The overlap between the subtensors are acceptable and considerable in many contexts, e.g. anomaly and fraud detection, because groups of fraudulent users might share some common smaller groups or some fraudsters. Another reason is that fraudulent behaviors of users might happen in just some specific periods of time. Compared to M-Zoom, M-Zoom can find more diverse subtensors, which can be explained as follows. M-

[3] https://dumps.wikimedia.org/

[4] https://nips.cc/



(a) Air Force (ρ_a) (b) Air Force (ρ_g) (c) Air Force (ρ_s) (d) Air Force (ρ_e) (e) Darpa (ρ_a) (f) Darpa (ρ_g) (g) Darpa (ρ_s) (h) Darpa (ρ_e)

(i) Android (ρ_a) (j) Android (ρ_g) (k) Android (ρ_s) (l) Android (ρ_e) (m) Enron (ρ_a) (n) Enron (ρ_g) (o) Enron (ρ_s) (p) Enron (ρ_e)

(q) StackO (ρ_a) (r) StackO (ρ_g) (s) StackO (ρ_s) (t) StackO (ρ_e) (u) Kowiki (ρ_a) (v) Kowiki (ρ_g) (w) Kowiki (ρ_s) (x) Kowiki (ρ_e)

(y) Enwiki (ρ_a) (z) Enwiki (ρ_g) (aa) Enwiki (ρ_s) (ab) Enwiki (ρ_e) (ac) YouTube (ρ_a) (ad) YouTube (ρ_g) (ae) YouTube (ρ_s) (af) YouTube (ρ_e)

(ag) NIPS (ρ_a) (ah) NIPS (ρ_g) (ai) NIPS (ρ_s) (aj) NIPS (ρ_e) (ak) LBNL (ρ_a) (al) LBNL (ρ_g) (am) LBNL (ρ_s) (an) LBNL (ρ_e)

Fig. 2: Average and bound of density on datasets (K: thousand, M: million, B: billion). Best viewed in color and zoom mode.

Zoom is specifically designed to find different subtensors by creating a snapshot of the data at each detection process, and it mines a block in this intermediate tensor. The results of this is, however, that M-Zoom cannot provide guarantee on the density of the detected subtensors, except on the first subtensor. This is one of the drawbacks of M-Zoom, and as discussed below (Section V-E), the effectiveness of M-Zoom on network attack detection greatly drops with multiple subtensors.

E. Effectiveness on Network Attack Detection

Air Force is specifically suitable for evaluating network attack detection ability. As mentioned earlier, it is a dataset of TCP dump data of a typical U.S. Air Force LAN. It contains the ground truth labels of connections, including both intrusions (or attacks) connections, and normal connections. In detail, there are 972,781 connections as normal, while other connections are attacks. This dataset is widely used for the task of detecting anomaly and network attacks.

Here, we demonstrate the efficiency, and the effectiveness of our proposed method on anomaly and network attack detection, and compare it with M-Zoom. We analyze the three highest subtensors returned by M-Zoom and MUST in the experiment in Section V-D on Air Force, and then we compute how many connections in the estimated subtensors are normal activities or attack[5]. Table V shows the connections in the top three subtensors detected by MUST and M-Zoom using arithmetic average mass (ρ_a) as the density metric. We observe that all connections in the top three subtensors found by MUST are attack connections with no false positive. This is because MUST guarantees the density of all multiple subtensors it finds. With M-Zoom, it has the same result in the top two subtensors. However, in the third subtensor, only 56,433 connections are attack, and 151,080 other connections are normal among 207,513 connections. In other words, M-Zoom produces a high rate of false positives, which in turn means that MUST outperforms M-Zoom, when used in the task of network attack detection, using the Air Force dataset.

F. Execution Time

In terms of execution time, to evaluate the performance of the algorithms, we recorded the runtime of the algorithms on real-world datasets using four measures of the density to return

[5] We provide the Matlab code to analyze attack connections in the code repository at https://bitbucket.org/duonghuy/mtensor/src/master/data/.

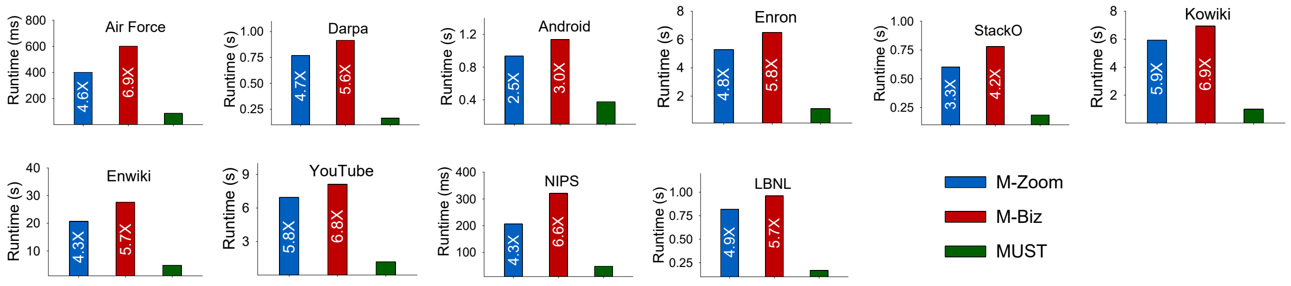


Fig. 3: Average runtime for a (sub)tensor on datasets. Best viewed in color.

TABLE IV: Diversity of estimated subtensors

	Dataset	#	Volume	Density	Diversity	Dataset	#	Volume	Density	Diversity	Dataset	#	Volume*	Density	Diversity
MUST		1	1 × 2 × 2	2397.6			1	2 × 2 × 2	273.0			1	X_1 × 2 × 1 × 1 × 1	1,980,948	
	Enwiki	2	1 × 4 × 5	2375.7	36.2%	Kowiki	2	4 × 4 × 5	258.5	37.2%	Air Force	2	X_1 × 1 × 1 × 1 × 1	1,930,307	20.8%
		3	1 × 3 × 3	2355.9			3	4 × 4 × 4	240.5			3	X_1 × 2 × 1 × 2 × 2	1,772,991	
M-Zoom		1	1 × 2 × 2	2397.6			1	2 × 2 × 2	273.0			1	X_1 × 2 × 1 × 1 × 1	1,980,948	
	Enwiki	2	1 × 2 × 3	1961.5	96.7%	Kowiki	2	2 × 2 × 3	246.0	99.4%	Air Force	2	X_1 × 1 × 1 × 1 × 1	263,295	70.8%
		3	2 × 3 × 3	908.25			3	16 × 41 × 45	181.6			3	X_2 × 5 × 4 × 3 × 3	60,524	

* Where $X_1 = 1 \times 1 \times 1$, and $X_2 = 3 \times 4 \times 2$.

TABLE V: Network attack detection on Air Force in the top three subtensors

	#	Volume	Density (ρ_a)	# Connections	# Attack Connections	# Normal Connections	# Ratio of Attack
MUST	1	2 (1 × 1 × 1 × 2 × 1 × 1 × 1)	1,980,948	2,263,941	2,263,941	0	100%
	2	1 (1 × 1 × 1 × 1 × 1 × 1 × 1)	1,930,307	1,930,307	1,930,307	0	100%
	3	8 (1 × 1 × 1 × 2 × 1 × 2 × 2)	1,772,991	2,532,845	2,532,845	0	100%
M-Zoom	1	2 (1 × 1 × 1 × 2 × 1 × 1 × 1)	1,980,948	2,263,941	2,263,941	0	100%
	2	1 (1 × 1 × 1 × 1 × 1 × 1 × 1)	263,295	263,295	263,295	0	100%
	3	4320 (3 × 4 × 2 × 5 × 4 × 3 × 3)	60,524	207,513	56,433	151,080	27%

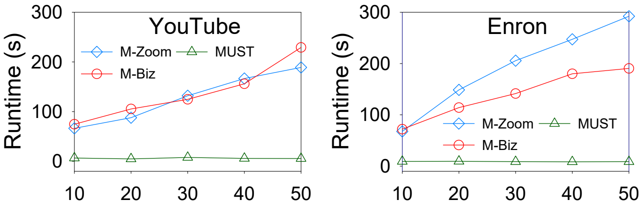


Fig. 4: Runtime while varying k. Best viewed in color.

top ten density subtensors. Then, we calculated the average runtime of the algorithms per each estimated subtensor. The results from this experiment are shown in Figure 3. We observe that MUST is much faster than M-Zoom and M-Biz on all the datasets. Specifically, it is up to 6.9 times faster than M-Zoom and M-Biz to estimate a subtensor. The obtained results fit well with our hypothesis and or complexity discussion in Section IV. The explanation for this is that in MUST the algorithm needs only a single maintaining process to get dense subtensors, while in M-Zoom and M-Biz, they repeatedly call the search function k times to be able to get k dense subtensors. The proposed method, MUST, runs nearly in constant time independent of the increase of the number of subtensors; whereas the execution times of both M-Zoom and M-Biz increase (near)linearly with respect to value of k.

G. *Scalability*

We also evaluate the impact of the number of estimated subtensors (k) to the performance of the algorithms. Here, we performed experiments on the Enron and YouTube datasets. With arithmetic average mass, we measured the runtime while varying k within {10, 20, 30, 40, 50}. Figure 4 shows the results of this experiment. As shown in the figure, the execution time of M-Zoom and M-Biz increase linearly with the increasing value of k, while the running time of MUST is constant with respect to the value of k. These results conform well with our complexity analysis in Section IV.

In conclusion, MUST outperforms the current state-of-the-art algorithms for solving the dense subtensor detection problem, from both a theoretical and experimental perspective.

VI. Conclusion

In this paper, we proposed a new technique to improve the task of dense subtensor detection. As discussed, the contributions are both theoretical and practical. First, we developed concrete theoretical proofs for dense subtensors estimation in a tensor problem. An important purpose of this was to provide a guarantee for a higher lower bound density of the estimated subtensors. In addition, we developed a new theoretical foundation to guarantee a high density of multiple subtensors. Second, extending existing dense subtensor detection methods, we developed a new algorithm called MUST that is less complex and thus more efficient than existing methods. Our experimental experiments demonstrated that the proposed method significantly outperformed the current state-of-the-art algorithms for dense subtensor detection problem. It is significantly more efficient and effective than the baseline methods. In conclusion, the proposed method is not only theoretically sound, but is also applicable for detecting dense subtensors. Nevertheless, when developing the proposed method, we observed that existing approaches (including ours) treat each tensor slice independently, and that they do not consider the relation among the slices within a tensor. To address this, in our future work, we will study the connection among slices when projecting on a way of a tensor. In addition, we will explore applying our method on graph data, and using it to solve event detection problems, such as change and anomaly detection.

References

[1] P. Comon, "Tensors : A Brief Introduction," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 44–53, 2014.

[2] W. Zhang, Z. Lin, and X. Tang, "Learning Semi-Riemannian Metrics for Semisupervised Feature Extraction," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 4, pp. 600–611, 2011.

[3] D. N. Holtmann-Rice, B. S. Kunsberg, and S. W. Zucker, "Tensors, Differential Geometry and Statistical Shading Analysis," *J. Math. Imaging Vis.*, vol. 60, pp. 968–992, 2018.

[4] N. D. Sidiropoulos, L. D. Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor Decomposition for Signal Processing and Machine Learning," *IEEE Trans. Sig. Proc.*, vol. 65, no. 13, pp. 3551–3582, 2017.

[5] X. Li, K. S. Candan, and M. L. Sapino, "M2TD: Multi-Task Tensor Decomposition for Sparse Ensemble Simulations," in *Proceedings of the 34th IEEE ICDE*, 2018, pp. 1144–1155.

[6] F. Yang, F. Shang, Y. Huang, J. Cheng, J. Li, Y. Zhao, and R. Zhao, "LFTF: A Framework for Efficient Tensor Analytics at Scale," *Proceedings of the VLDB Endowment*, vol. 10, no. 7, pp. 745–756, 2017.

[7] S. Oh, N. Park, L. Sael, and U. Kang, "Scalable Tucker Factorization for Sparse Tensors - Algorithms and Discoveries," in *Proceedings of the 34th IEEE ICDE*, 2018, pp. 1120–1131.

[8] N. Park, S. Oh, and U. Kang, "Fast and Scalable Method for Distributed Boolean Tensor Factorization," *The VLDB Journal*, vol. 28, no. 4, pp. 549–574, 2019.

[9] K. Maruhashi, F. Guo, and C. Faloutsos, "MultiAspectForensics: Pattern Mining on Large-Scale Heterogeneous Networks with Tensor Analysis," in *Proceedings of ASONAM*, 2011, pp. 203–210.

[10] B. Hooi, L. Akoglu, D. Eswaran, A. Pandey, M. Jereminov, L. Pileggi, and C. Faloutsos, "ChangeDAR: Online Localized Change Detection for Sensor Data on a Graph," in *Proceedings of the 27th ACM CIKM*, 2018, pp. 507–516.

[11] K. Shin, B. Hooi, J. Kim, and C. Faloutsos, "D-Cube: Dense-Block Detection in Terabyte-Scale Tensors," in *Proceedings of the 10th ACM WSDM*, 2017, pp. 681–689.

[12] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear Analysis of Image Ensembles: TensorFaces," in *Proceedings of ECCV*, 2002, pp. 447–460.

[13] A. V. Goldberg, "Finding a Maximum Density Subgraph," Tech. Rep., 1984.

[14] S. Khuller and B. Saha, "On Finding Dense Subgraphs," in *Proceedings of the 36th International Colloquium on Automata, Languages and Programming: Part I*, ser. ICALP '09, 2009, pp. 597–608.

[15] K. Shin, B. Hooi, and C. Faloutsos, "M-Zoom: Fast Dense-Block Detection in Tensors with Quality Guarantees," in *Proceedings of ECML PKDD*, 2016, pp. 264–280.

[16] K. Shin, B. Hooi, and C. Faloutsos, "Fast, Accurate, and Flexible Algorithms for Dense Subtensor Mining," *ACM Trans. Knowl. Discov. Data*, vol. 12, no. 3, pp. 28:1–28:30, 2018.

[17] M. Charikar, "Greedy Approximation Algorithms for Finding Dense Components in a Graph," in *Proceedings of APPROX*, 2000, pp. 84–95.

[18] E. Galbrun, A. Gionis, and N. Tatti, "Top-k Overlapping Densest Subgraphs," *Data Min. Knowl. Discov.*, vol. 30, no. 5, pp. 1134–1165, 2016.

[19] K. Shin, B. Hooi, J. Kim, and C. Faloutsos, "DenseAlert: Incremental Dense-Subtensor Detection in Tensor Streams," in *Proceedings of the 23rd ACM SIGKDD*, 2017, pp. 1057–1066.

[20] T. G. Kolda and B. W. Bader, "Tensor Decompositions and Applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[21] S. Zhou, N. X. Vinh, J. Bailey, Y. Jia, and I. Davidson, "Accelerating Online CP Decompositions for Higher Order Tensors," in *Proceedings of the 22nd ACM SIGKDD*, 2016, pp. 1375–1384.

[22] P. Rozenshtein, A. Anagnostopoulos, A. Gionis, and N. Tatti, "Event Detection in Activity Networks," in *Proceedings of the 20th ACM SIGKDD*, 2014, pp. 1176–1185.

[23] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, "FRAUDAR: Bounding Graph Fraud in the Face of Camouflage," in *Proceedings of the 22nd ACM SIGKDD*, 2016, pp. 895–904.

[24] Y. Ban, X. Liu, Y. Duan, X. Liu, and W. Xu, "No Place to Hide: Catching Fraudulent Entities in Tensors," in *Proceedings of The Web Conference, WWW*, 2019, pp. 83–93.

[25] Y. Asahiro, R. Hassin, and K. Iwama, "Complexity of Finding Dense Subgraphs," *Discrete Appl. Math.*, vol. 121, no. 1, pp. 15–26, 2002.

[26] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama, "Greedily Finding a Dense Subgraph," *J. Algorithms*, vol. 34, no. 2, pp. 203–221, 2000.

[27] O. D. Balalau, F. Bonchi, T.-H. H. Chan, F. Gullo, and M. Sozio, "Finding Subgraphs with Maximum Total Density and Limited Overlap," in *Proceedings of the 8th ACM WSDM*, 2015, pp. 379–388.

[28] X. Liu, S. Ji, W. Glänzel, and B. De Moor, "Multiview Partitioning via Tensor Methods," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 5, pp. 1056–1069, 2013.

[29] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. Tsiarli, "Denser Than the Densest Subgraph: Extracting Optimal Quasi-cliques with Quality Guarantees," in *Proceedings of the 19th ACM SIGKDD*, 2013, pp. 104–112.

[30] M. Jiang, A. Beutel, P. Cui, B. Hooi, S. Yang, and C. Faloutsos, "A General Suspiciousness Metric for Dense Blocks in Multimodal Data," in *Proceedings of the IEEE ICDM*, 2015, pp. 781–786.

[31] R. Andersen and K. Chellapilla, "Finding Dense Subgraphs with Size Bounds," in *Proceedings of WAW*, 2009, pp. 25–37.

[32] R. Andersen, "A Local Algorithm for Finding Dense Subgraphs," *ACM Trans. Algorithms*, vol. 6, no. 4, pp. 60:1–60:12, 2010.

[33] R. He and J. McAuley, "Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering," in *Proceedings of the 25th The Web Conference, WWW*, 2016, pp. 507–517.

[34] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA Off-line Intrusion Detection Evaluation," *Comput. Netw.*, vol. 34, no. 4, pp. 579–595, 2000.

[35] R. Pang, M. Allman, V. Paxson, and J. Lee, "The Devil and Packet Trace Anonymization," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 29–38, 2006.

[36] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, "Euclidean Embedding of Co-occurrence Data," *J. Mach. Learn. Res.*, vol. 8, pp. 2265–2295, 2007.

[37] J. Kunegis, "KONECT: The Koblenz Network Collection," in *Proceedings of the 22nd The Web Conference, WWW*, 2013, pp. 1343–1350.

[38] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks," in *Proceedings of the 7th ACM SIGCOMM*, 2007, pp. 29–42.