Density Guarantee on Finding Multiple Subgraphs and Subtensors

QUANG-HUY DUONG*, Norwegian University of Science and Technology, Norway
HERI RAMAMPIARO, Norwegian University of Science and Technology, Norway
KJETIL NØRVÅG, Norwegian University of Science and Technology, Norway
THU-LAN DAM*, Norwegian University of Science and Technology, Norway

Dense subregion (subgraph & subtensor) detection is a well-studied area, with a wide range of applications, and numerous efficient approaches and algorithms have been proposed. Approximation approaches are commonly used for detecting dense subregions due to the complexity of the exact methods. Existing algorithms are generally efficient for dense subtensor and subgraph detection, and can perform well in many applications. However, most of the existing works utilize the state-or-the-art greedy 2-approximation algorithm to capably provide solutions with a loose theoretical density guarantee. The main drawback of most of these algorithms is that they can estimate only one subtensor, or subgraph, at a time, with a low guarantee on its 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 and giving a higher lower bound of the density. In particular, we guarantee and prove a higher bound of the lower-bound density of the estimated subgraph and 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.

Additional Key Words and Phrases: Dense Subtensor, Dense Subgraph, Multiple Subtensor Detection, Density Guarantee, Event Detection

ACM Reference Format:
Quang-Huy Duong, Heri Ramampiaro, Kjetil Nørvåg, and Thu-Lan Dam. 2021. Density Guarantee on Finding Multiple Subgraphs and Subtensors. *ACM Trans. Knowl. Discov. Data.* 1, 1, Article 1 (January 2021), 32 pages. https://doi.org/10.1145/3446668

1 INTRODUCTION

In many real-world applications, generated data are commonly represented in complex structures such as graphs or multidimensional arrays, that can be referred to as *tensors* [8]. Tensors and graphs are used in several important domains, including geometry, physics and biology as well as computer science [16, 29, 37, 47]. As a result of the growth in the number of applications involving tensors, graphs, combined with the increase of researchers' interests, numerous tensor, graph-related approaches have been proposed, including tensor decomposition [23, 42], tensor factorization [30, 32, 46], and dense subgraph detection [12, 17, 22].

*Corresponding author.

Authors' addresses: Quang-Huy Duong, Norwegian University of Science and Technology, Norway, huydqyb@gmail.com; Heri Ramampiaro, Norwegian University of Science and Technology, Norway, heri@ntnu.no; Kjetil Nørvåg, Norwegian University of Science and Technology, Norway, noervaag@ntnu.no; Thu-Lan Dam, Norwegian University of Science and Technology, Norway, lanfict@gmail.com.

1556-4681/2021/1-ART1 $15.00
https://doi.org/10.1145/3446668

Dense subregion detection has been extensively studied and has attracted much interest, due to a wide range of real-life applications [6, 33, 36, 45]. Finding the densest subtensor or the densest subgraph is generally an NP-complete, or an NP-Hard problem [3, 14, 19]; and how hard the problem of detecting the densest subtensor/subgraph is varies with the choice of constraint requirements, e.g, the size and the dimension of the data, and the chosen density measure. Due to the complexity of an exact algorithm, it is infeasible for large data or in dynamic environments, such as streaming data. Therefore, approximation methods are commonly used for detecting the densest subregions [4, 5, 7]. Along this line, GREEDY is an efficient approximation algorithm that was proposed to find the optimal solution in a weighted graph [4]. Charikar [7] introduced a further analysis of the GREEDY, and the analysis showed that the GREEDY method can be solved by using a linear programming technique. The authors proposed a greedy 2-approximation for this optimization problem, with a density guarantee of the dense subgraph greater than half of the maximum density in the graph. Due to its efficiency, several algorithms have adopted the greedy method with a guarantee on the density of dense subgraphs, and implemented it in various real-life applications, such as fraud detection, event detection, and genetics applications [17, 33, 45], among others. Common for these works is that they use the greedy 2-approximation to find a dense subgraph to optimize an objective of a given interest density measure.

Besides graphs, tensor has gradually attracted much interest of researchers, because much of the data being used in many real applications can be represented naturally in the form of a tensor. Hence, various algorithms have been proposed by extending the works on dense (sub)graph detection to tensor data for specific applications, such as network attack detection, change detection in communication networks, and fraud detection [10, 18, 26, 40]. Here, M-Zoom [38] and M-Biz [39] are among the current state-of-the-art dense subtensor detection algorithms. These algorithms extend the approaches on dense (sub)graph detection, such as [7, 11], 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. Nevertheless, the adopted density guarantee is the same as in the original work, without any improvement on the density guarantee. M-Zoom and M-Biz are capable 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 only. Hence, M-Zoom and M-Biz solely provide a density guarantee with respect to the current intermediate tensor rather than the original input tensor. A more recent approach, called DenseAlert [41], 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 a 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, 48], 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. On top of this, we provide a mathematical foundation for provision of a higher density guarantee in detecting both dense subgraphs and dense subtensors. Specifically, the new boundary is better and does not only depend on the dimension of the data space, but our novel found density guarantee is also constrained on the size of the densest subtensor/subgraph. To show this, focus of this paper is two-fold. 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 of providing a better bound of density, but also provides an effective method to estimate these dense subtensors.

To summarize the differences between our method and the existing approaches, Table 1 compares the characteristics of MUST against current state-of-the-art algorithms. With this in mind, the main contributions of this work are as follows:

(1) We introduce a foundation to theoretically guarantee a better density of both estimated subgraph and subtensor in dense subgraph, and dense subtensor detection. Here, 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 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.
   - For the dense subgraph detection, the new density bound is $\frac{1}{2}(1 + \frac{1}{min(y,\sqrt{n})})$, where n and y denote the size of the graph and the densest subgraph, respectively.
(2) We present a novel theoretical foundation, along with proofs showing that it is possible to maintain multiple subtensors with a density guarantee.
(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 2 gives an overview of related work. Section 3 describes the preliminaries for the method. Section 4 elaborates on the theoretical foundation for providing a new density guarantee of dense subtensors. Section 5 provides a new better density guarantee of subgraph in dense subgraph detection problem. Section 6 presents the solution for detecting multiple dense subtensors with a density guarantee. Section 7 discusses the evaluation of our method and explains its applicability. Finally, Section 8 concludes the paper and outlines the future work.

This paper extends our work in [9], and provides an extensive and more thorough study of the problem. In particular, we provide a theoretical foundation on the density guarantee on graph data. We also present new experimental results and provide in-depth discussion of the results in the dense subregion detection problem.
Reproducibility: The source code and data used in the paper are publicly available at https://bitbucket.org/duonghuy/mtensor.

2 BACKGROUND AND RELATED WORK

The problem of finding the densest subgraphs is generally NP-complete or NP-hard [3, 14]. 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 subregions [4, 5, 7]. Ashiro et al. [4] 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 [7] 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 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 [17, 25, 33, 45]. Common for these works is their use of the greedy 2-approximation to find a dense subgraph.

Table 1. A brief comparison of between existing algorithms and MUST

Algorithm	**Approx**	**Multiple estimation support**	**Single density guarantee**	**Multiple density guarantee**	**Number of guaranteed estimations**	**Bound guarantee on Subtensor***	**Bound guarantee on Subgraph****
Goldberg's [14]			✓		1		1
GREEDY [7]	✓		✓		1		$\frac{1}{2}$
GreedyAP [33]	✓		✓		1		$\frac{1}{2}$
M-Zoom [38]	✓	✓	✓		1	$\frac{1}{N}$	
DenseAlert [41]	✓		✓		1	$\frac{1}{N}$	
M-Biz [39]	✓	✓	✓		1	$\frac{1}{N}$	
FrauDar [17]	✓		✓		1	$\frac{1}{N}$	
ISG+D-Spot [6]	✓	✓					
MUST	✓	✓	✓	✓	$\boldsymbol{min(1+\frac{n}{2N}, 1+N(N-1))}$	$\boldsymbol{\frac{1}{N}(1+\frac{N-1}{min(x,\sqrt{n})})}$	$\boldsymbol{\frac{1}{2}(1+\frac{1}{min(x,\sqrt{n})})}$

* 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).
** n and x are the size of data (tensor, graph) and the size of the densest subregion (subgraph, subtensor) respectively.

Supported Data	Goldberg's	GREEDY	GreedyAP	M-Zoom	DenseAlert	M-Biz	FrauDar	ISG+D-Spot	Ours
Tensor				✓	✓	✓			✓
Graph	✓	✓	✓					✓	✓

Inspired by the theoretical solutions in graphs, numerous approaches have been proposed to detect dense subtensors by using the same min-cut mechanism [39, 41]. As mentioned earlier, mining the densest subgraph in a graph is generally an NP-hard problem, and an exact mining approach might have a polynomial time complexity [14]. However the degree of the polynomial function of the complexity is extremely high even in a simple case with restricted constraints. In particular, in a simple case of the dense subgraph detection (N=2), there is no weight at vertices, the running time complexity of an exact method for using the average weighted degree as a density measure is $O(n^6)$ [4, 14], 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 [41], M-Zoom [38], and M-Biz [39] 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 [26], and CrossSpot [18]. DenseAlert, M-Zoom, and M-Biz adapt the theoretical results from dense (sub)graph detection, i.e., [1, 2, 45], 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 [18]. However, the adopted density guarantee is the same as in the original work, which also applies for the more recent algorithm, *ISG+D-Spot* [6]. 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.

The greedy 2-approximation approach has been utilized in many algorithms with both types of data, graph and tensor [34–36, 44]. Despite this, most current works, including [6, 17, 28, 35, 39] can only roughly provide a guarantee of half (or $\frac{1}{N}$ for a subtensor) the density of the densest subregion. None of the existing approximation schemes provides a better guarantee than the baseline algorithms [4, 7], and such schemes can only provide a loose theoretical density detection guarantee. As discussed in Section 1, DenseAlert, M-Zoom, and M-Biz employed the same guarantee as in the original work without any further improvement in the density guarantee. Thus, these methods can only guarantee low density subtensors. 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.

3 PRELIMINARIES

In the following, we present the fundamental preliminaries of the dense subtensor, subgraph detection problem, based on [39, 41].

Dense Subgraph Detection

Definition 1 (Graph). *Let G be an undirected graph that is composed by a pair $(V; E)$ of a set vertices V and edges E. We denote the graph as $G(V; E)$. There is a weight a_i at each vertex v_i, and a weight c_{ij} on each edge e_{ij} between two vertices v_i and v_j in G.*

Definition 2 (Density of Graph). *Density of G is denoted by $\rho(G)$ and is defined by: $\rho(G) = \frac{\sum a_i + \sum c_{ij}}{|V|} = \frac{f(G)}{|V|}$, where $|V|$ is number of vertices of G, and $f(G) = \sum a_i + \sum c_{ij}$, $f(G)$ is called the mass of graph G.*

Definition 3 (SubGraph). *Let G be an undirected graph that is composed by a pair $(V; E)$ of a set vertices V and edges E. S is a subgraph of G if S is induced by a subset vertices of V and edges in E.*

Definition 4 (Weight of vertex in Graph). *Given a graph $G(V, E)$ with weight a_i at vertex v_i, and weight c_{ij} on edge between 2 vertices v_i, v_j. Weight of vertex v_i in graph G is denoted by $w_i(G)$, and is defined by: $w_i(G) = a_i + \sum_{v_j \in G \wedge e_{ij} \in E} c_{ij}$.*

Definition 5 (Dense Subgraph Detection Problem). *Given an undirected graph $G = (V; E)$ and a density measure df. The problem of dense subgraph detection is to find subgraphs S induced by a subset of vertices of V and edges in G to maximize density of S.*

The processing of the greedy approximation algorithm is as follows [4, 7]. The algorithm iteratively removes a vertex with the minimum weighted-degree in the currently remaining graph until all vertices are removed. Finally, it picks the highest density subgraph among the estimated subgraphs. The algorithm gives a 2-approximation with a density guarantee of half of the maximum density in the graph. Note that, in the original work, the density measure is average of weighted-degree of the graph. In this paper, we consider a general density measure of both weights at vertices and on edges.

Dense Subtensor Detection

Several dense subtensor detection methods have been proposed by extending the works in dense (sub)graph detection to tensor data. However, they use the same min-cut mechanism as in dense subgraph detection [6, 39, 40]. These methods employed the same guarantee as in the original work without any improvement in density guarantee. In this chapter, we generalize the problem in both dense subtensor and dense subgraph detection and propose our new theoretical proofs to give a better approximation guarantee of the density. In the rest of this chapter, we use subregion to indicate both subtensor and subgraph.

Definition 6 (Tensor). *A tensor T is a multidimensional array data. The order of T is its number of ways. Given an N-way tensor, on each way, there are multiple spaces, each of which is called a slice.*

Definition 7 (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 8 (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 9 (Size of a (sub)Tensor). *Given a (sub)Tensor Q, the size of Q is the number of slices that compose Q.*

Definition 10 (Density). *Given a (sub)tensor Q, the density of Q, denoted by $\rho(Q)$, is computed as: $\rho(Q) = \frac{f(Q)}{\text{size of } Q}$, where $f(Q)$ is mass of the (sub)tensor Q, and is computed as the sum of every entry values of Q.*

Definition 11 (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 12 (D-Ordering). *An ordering π on a (sub)tensor Q is a D-Ordering, if*

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

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 [4, 7].

Definition 13 (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.*

Table 2. Table of notations

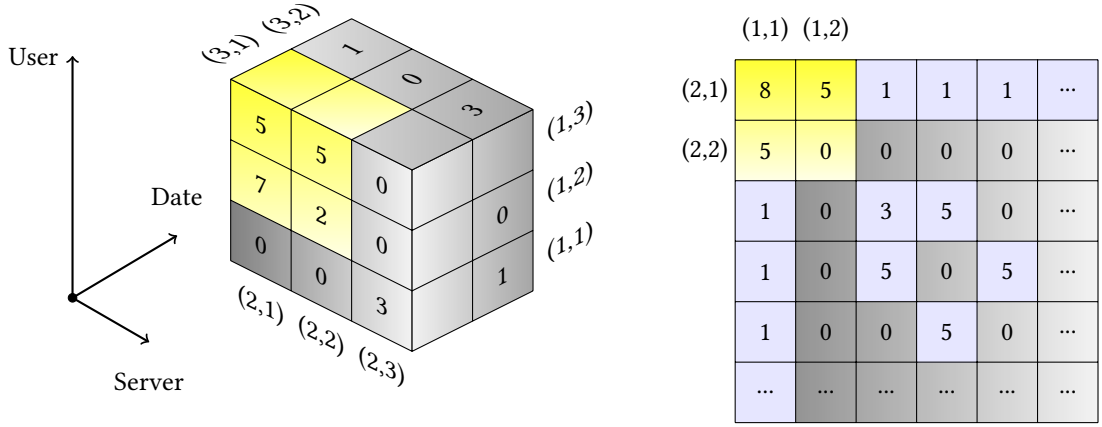
Symbols	Description
T, Q	Tensor data T, Q
I_i	The i-th dimension of tensor I
$\|I_i\|$	Number of slices on way I_i of a tensor I
T^*	Densest subtensor T^*
G	(Sub)Graph data G.
G^*	Densest Subgraph.
v_i, a_i	Vertex v_i, and weight a_i at vertex v_i.
c_{ij}	Weight on edge between two vertices v_i and v_j.
Z, z_0	Zero subtensor Z with zero point z_0
B	Backward subtensor
F	Forward subtensor
n, N	Size (with tensor, it is number of slices, with graph, it is number of vertices), and number of ways of data
ρ, ρ^*	Density ρ, highest density ρ^*
$\rho(Q)$	Density of 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
y	Size of densest subgraph
m	Size of Zero subtensor Z, $m = a + b$
$f(Q)$	Mass of the (sub)tensor Q
$w_q(Q)$	Weight of element q (vertex, or slice) in data Q (graph, or tensor).

For readability, the notations used in this paper are summarized in Table 2. 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 1a. The value in each cell is the number of visits that a user (mode User) visits a web page (mode Server) 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.*

Let π be a D-Ordering on T, and π is defined as in definition 12. Intuitively, the D-Ordering π is the order that we pick and remove the minimum slice sum in each step of our process until there is no slice to proceed. The detailed steps of how we construct π are illustrated as follows.

- *The first slice in π is (1,1) because its weight (the sum of entry values composing by (1,1)) is 1 + 3 = 4, and it is the minimum value among all the sum values of all slices in T.*
- *After the slice (1,1) is removed, the set of the remaining slices is {(1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2)}. The slice (2,3) which now has the minimum slice sum (the sum is 3) is therefore chosen next. The remaining slices are {(1,2), (1,3), (2,1), (2,2), (3,1), (3,2)} after the second step, and the first two slices in π are (1,1) and (2,3).*



(a) 3-way tensor

(b) 2-way tensor

Fig. 1. Examples of tensor

- *The process is then continued in the same way for the remaining slices until all slices are ordered.*
- *The final order in π on T when the process ends is {(1,1), (2,3), (3,2), (2,2), (1,3), (1,2), (2,1), (3,1)}*

Meanwhile in Figure 1b, it is an example of 2-way tensor. The tensor in Figure 1b, T, can be represented as a matrix. The yellow region in the tensor is the densest subtensor in T, and its density is $\frac{8+5+5}{4} = 4.5$. The density of the subtensor Q formed by the first three columns and the first three rows is $\frac{(8+5+1+5+1+3)}{6} = \frac{23}{6}$.

The problem of mining dense subtensors [39, 41] can be presented and solved as follows. Given a list of n variables $d_\pi(q_i)$ $(1 \leq i \leq n)$, where $d_\pi(q_i)$ is calculated during the construction of D-Ordering, presented as function D-Ordering in Algorithm 1. 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 $i^* = \underset{1 \leq i \leq n}{\operatorname{argmax}}\ \rho_\pi(i)$, 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 [38, 39, 41], 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.

Theorem 1 (Density Guarantee) [39, 41]). *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 [39, 41]. For convenience, we recall their proof as follows. Let $q^* \in T^*$ be the slice such that $\pi^{-1}(q^*) \leq \pi^{-1}(q), \forall q \in T^*$, where T^* is the densest subtensor in T. This means that q^* is the slice in the densest subtensor having the smallest index in π. Therefore $\rho_\pi(i^*) \geq \rho_\pi(\pi^{-1}(q^*)) \geq \frac{1}{N}\rho^*$. □

4 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?

Algorithm 1 Find-Slices

Require: A D-Ordering π on a set of slices Q
Ensure: An estimated subtensor S
1: $S \leftarrow \emptyset, m \leftarrow 0$
2: $\rho_{max} \leftarrow -\infty, q_{max} \leftarrow 0$
3: **for** $(j \leftarrow |Q|..1)$ **do**
4: $q \leftarrow \pi(j), S \leftarrow S \cup q$
5: $m \leftarrow m + d_\pi(q)$
6: **if** $m/|S| > \rho_{max}$ **then**
7: $\rho_{max} \leftarrow m/|S|$
8: $q_{max} \leftarrow q$
9: **end if**
10: **end for**
11: **return** $Q(\pi, \pi^{-1}(q_{max}))$
1: **function** D-Ordering(a set of slices Q)
2: R ←Q, π←$\emptyset$, d_π ←$\emptyset$
3: **for** $(i = 1;\ i \leq |Q|;\ i = i + 1)$ **do**
4: q ←$\underset{p \in R}{\text{argmin}}\ w_p(R)$
5: $\pi(i) \leftarrow q$
6: $d_\pi(q) \leftarrow w_q(R)$
7: R = R \ {q}
8: **end for**
9: **return** π, d_π
10: **end function**

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.

4.1 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 [39, 41], 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 [39, 41] can be provided. We present several theorems and properties to support our solution to estimate multiple dense subtensors.

Definition 14 (Zero Subtensor). *Given a tensor T, T^* is the densest subtensor in T with density ρ^*, π is a D-ordering on T, and $z_0 = \min_{q \in T^*} \pi^{-1}(q)$ 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,

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

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

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

□

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. □

4.2 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

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

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

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

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, $(i \le i \le M)$. According to Theorems 3 and 4, we have

$$f(T^*) \le \sum_{i=1}^{M} w_{q_i}(T_i) \le \sum_{i=1}^{M} N\rho(T_i) \le MN\rho_{max} \tag{5}$$

$$\Rightarrow a\rho^* \le N(a - N + 1)\rho_{max} \tag{6}$$

$$\Rightarrow \rho_{max} \ge \frac{\rho^*}{N}\frac{a}{a - N + 1}. \tag{7}$$

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

$$\rho_{max} \ge \rho(T_1) \ge \frac{1}{N}(1 + \frac{a(N-1)}{n})\rho^* \tag{8}$$

Furthermore, by Inequation 7, we also have

$$\rho_{max} \ge \frac{\rho^*}{N}\frac{a}{a - N + 1} \ge \frac{1}{N}(1 + \frac{N-1}{a})\rho^* \tag{9}$$

By combining Eq. 8 and Eq. 9, we get

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

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

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

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

4.3 Illustrated Example

Let's consider an example of a 3-way tensor T as in Figure 1a. The value in each cell is considered as the number of requests that a user (probably an attacker, in mode User) sends to a server (mode Server) in a period of time (mode Date). The values in the hidden cells are all zeros. Our task is to analyze the data to detect attackers. The set of slices of tensor T is {(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2)}.

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 = $\frac{19}{5}$.

Here the number of ways of T is 3, and its size (number of slices that composes T) is 8. The existing methods can only give a guarantee of the estimated subtensor as a fraction of the highest density. The guarantee in this case is:

$$\frac{1}{N}\rho^* = \frac{19}{5} \times \frac{1}{3} = \frac{19}{15}.$$

However, by using our new proof, we can prove that the new lower bound of density in this example is guaranteed to be greater than:

$$\frac{1}{N}(1 + \frac{N-1}{min(a,\sqrt{n})})\rho^* \ge \frac{19}{15}(1 + \frac{3-1}{\sqrt{8}}) = \frac{2+\sqrt{2}}{2}\frac{19}{15} \ge \frac{1.7 \times 19}{15}$$

Comparing the two guarantees, our proposed guarantee on the density is 1.7 ($\simeq 1 + \frac{\sqrt{2}}{2}$) times greater than the guarantee provided by the existing state-of-the-art methods. Hence, our proposed guarantee is more than 70% higher than the current guarantee.

5 THE NEW DENSITY GUARANTEE OF SUBGRAPH

As aforementioned, most existing methods in dense subregion detection in both tensor and graph employed the same guarantee as in the original work without any improvement in density guarantee. To address this limitation, in this study, we generalize the problem in both dense subtensor and dense subgraph detection. We propose our new theoretical proofs to give a better approximation guarantee of the density for the problem in both tensor and graph data. In Section 4, we proved and provided a new bound on the density in a tensor data. Now we raise the following questions with the original problem of detecting dense subgraph: (1) Can we provide a higher guarantee on the density of the dense estimated subgraph in a graph? (2) Is this bound constrained to any other information rather than the dimension of the data space? This section answers the question by introducing our proofs to give a better approximation guarantee on the density of the estimated subgraph in a graph, that is the original foundation for both dense subgraph and dense subtensor detection problems. Our novel mathematical proof here is capable of giving a better guarantee for the current state-of-the-art methods, and shows that the bound is also constrained to the size of the densest subgraph.

Theorem 6 (Density Guarantee of Dense Subgraph Detection). *Given an undirected graph $G(V;E)$ with size $n = |V|$. Let G^* be the densest subgraph in G. There exists a number $p \geq 0$ such that the lower bound density of estimated subgraph in the GREEDY [7] is $\frac{2y+p}{2(y+p)}\rho^*$, where ρ^* is the density of the densest subgraph G^*, y is the size of G^*, and $y \leq (y+p) \leq n$.*

Proof. Let G_1 be the subgraph that is right before we pick the first vertex of the densest subgraph G^* to be removed, we denote the vertex is v_{s1}. So definitely we have: $G^* \subseteq G_1$, and the size of G_1 is $n_1 \leq n$. We have:

$$\begin{aligned} 2f(G_1) &= 2\sum_{v_i \in G_1} a_i + 2\sum_{v_i, v_j \in G_1} c_{ij} \\ &= \sum_{v_i \in G_1} a_i + \sum_{v_i \in G_1} w_i(G_1) \\ &= \sum_{v_i \in G_1} a_i + \sum_{v_i \in G_1 \wedge v_i \in G^*} w_i(G_1) + \sum_{v_i \in G_1 \wedge v_i \notin G^*} w_i(G_1) \end{aligned}$$

Let's denote $V(G_1 \backslash G^*) = \{v_i, v_i \in G_1 \wedge v_i \notin G^*\}$ and $p = |V(G_1 \backslash G^*)|$. Because v_{s1} is chosen for the cut, it means that v_{s1} has the minimum cut weight, so we get: $w_j(G_1) \geq w_{s1}(G_1), \forall v_j \in G_1$, and G^* is the densest subgraph then $w_{s1}(G^*) \geq \rho^*$. Therefore:

$$\sum_{v_i \in G_1 \wedge v_i \notin G^*} w_i(G_1) \geq p \times w_{s1}(G_1) \geq p \times w_{s1}(G^*) \geq p \times \rho^*.$$

On the other hand, we have:

$$\begin{aligned} \sum_{v_i \in G_1} a_i + \sum_{v_i \in G_1 \wedge v_i \in G^*} w_i(G_1) &\geq \sum_{v_i \in G^*} a_i + \sum_{v_i \in G_1 \wedge v_i \in G^*} w_i(G^*) \\ &\geq 2(\sum_{v_i \in G^*} a_i + \sum_{v_i, v_j \in G^*} c_{ij}) \\ &\geq 2f(G^*) \end{aligned}$$

Note that, size of G_1 is $n_1 = y + p$, finally we have:

$$\begin{aligned} & 2f(G_1) \geq 2f(G^*) + p \times \rho^* \\ \Rightarrow \quad & 2(y+p)\rho(G_1) \geq 2y\rho^* + p \times \rho^* \\ \Rightarrow \quad & \rho(G_1) \geq \frac{2y+p}{2y+2p} \times \rho^*, \end{aligned}$$

where ρ^* is the highest density and $y \leq n_1 = (y + p) \leq n$. The theorem is proved. □

Theorem 7 (Density Guarantee Boundary In Graph). *The density of the subgraph G_1 as in Theorem 6 is $\rho(G_1)$, and this density is in $[\frac{1}{2}(1 + \frac{y}{n})\rho^*, \rho^*]$, where ρ^* is the highest density in G.*

Proof. Because ρ^* is the highest density so $\rho(G_1) \leq \rho^*$. Moreover, by Theorem 6, we have (because $n_1 = y + p \leq n$):

$$\frac{\rho(G_1)}{\rho^*} \geq \frac{2y+p}{2y+2p} = \frac{1}{2}(1 + \frac{y}{y+p}) \tag{10}$$

$$\Rightarrow \rho(G_1) \geq \frac{1}{2}(1 + \frac{y}{n})\rho^*$$ □

According to Theorem 7, the density of the subgraph G_1 is in the boundary $[\frac{1}{2}(1 + \frac{y}{n})\rho^*, \rho^*]$. We denote G_1, G_2, …, G_m are subgraphs right before we are going to remove vertex v_1, v_2, …, v_y of G^*. Intuitively, G_i is the subgraph right before we remove i-*th* vertex of G^*. The corresponding min-cut at the step of processing G_i is denoted as w_i. We have a following property about the min-cut value.

Property 3 (Upper Bound Of The Min-Cut Value In Graph). *Given an undirected graph $G(V, E)$ with vertex v_i having the minimum cut (its weight is minimum). The weight of vertex v_i in graph G satisfies the following inequality:*

$$w_i(G) \leq 2\rho(G) - \overline{a}(G), \tag{11}$$

where $\overline{a}(G) = \frac{\sum_{v_k \in G} a_k}{|V|}$ is the average weight of all vertices in G.

Proof. Because v_i is a vertex having the minimum cut, so we have $w_i(G) \leq w_k(G), \forall v_k \in G$. Sum up of all the vertices in the graph, we get:

$$\begin{aligned} & |V| w_i(G) \leq \sum_{v_k \in G} w_k(G) = \sum_{v_k \in G} a_k + 2 \sum_{v_k, v_j \in G} c_{kj} \\ \Rightarrow \quad & |V| w_i(G) \leq 2(\sum_{v_k \in G} a_k + \sum_{v_k, v_j \in G} c_{kj}) - \sum_{v_k \in G} a_k \\ \Rightarrow \quad & w_i(G) \leq \frac{2(\sum_{v_k \in G} a_k + \sum_{v_k, v_j \in G} c_{kj}) - \sum_{v_k \in G} a_k}{|V|} \\ \Rightarrow \quad & w_i(G) \leq 2\rho(G) - \overline{a}(G). \end{aligned}$$

□

Let ρ_{max} be the maximum density among subgraphs G_i,

$$\rho_{max} = max(\rho(G_i)) \tag{12}$$

We have:

$$\sum_{i=1}^{y-1} w_i(G_i) + a_y = w_1(G_1) + w_2(G_2) + \cdots + w_{y-1}(G_{y-1}) + a_y$$
$$\geq \sum_{v_i \in G^*} a_i + \sum_{v_i, v_j \in G^*} c_{ij} = f(G^*)$$

if we assume that $a_n = 0$ as in the GREEDY algorithm [7], or in many other works in the literature, they assume that weight at vertices are zero [6, 17], so we have:

$$\sum_{i=1}^{y-1} w_i(G_i) \geq f(G^*) \tag{13}$$
$$\Rightarrow 2(y-1)\rho_{max} \geq y\rho^* \tag{14}$$
$$\Rightarrow \rho_{max} \geq \frac{y}{2(y-1)}\rho^* \tag{15}$$
$$\Rightarrow \rho_{max} \geq \frac{1}{2}(1 + \frac{1}{y})\rho^* \tag{16}$$

Theorem 8 (Better Density Guarantee of Dense Subgraph). *The density guarantee of dense subgraph mining by the min-cut mechanism is greater than* $\frac{1}{2}(1 + \frac{1}{min(y,\sqrt{n})})\rho^*$, *where* ρ^* *is the highest density value in the graph.*

Proof. According to Theorem 6, we have:

$$\rho_{max} \geq \rho(G_1) \geq \frac{1}{2}(1 + \frac{y}{n})\rho^*. \tag{17}$$

Furthermore, note that we have

$$\rho_{max} \geq \frac{1}{2}(1 + \frac{1}{y})\rho^*, \text{ by Inequation [16]}$$

We combine together two inequations [16-17], we get:

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

Note that $\rho_{max} \geq \frac{1}{2}(1 + \frac{1}{y})\rho^*$, so finally we have:

$$\rho_{max} \geq \frac{1}{2}(1 + \frac{1}{min(y, \sqrt{n})})\rho^* \qquad \square$$

6 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.

6.1 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 $z_0 = \min_{q \in T^*} \pi^{-1}(q)$ is the smallest indices in π among all slices in T^* (cf. Definition 14).

Definition 15 (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:

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

From the above inequality, we get the following theorem.

Theorem 9. *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

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

□

Property 4. *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 9, 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. □

6.2 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 16 (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 5. *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 10. *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:

$$Kf(Z) \geq Kf(T^*) + \sum_{q \in Z \wedge q \notin T^*} w_q(Z), \tag{18}$$

By substitution, we get

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

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:

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

□

Theorem 11. *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,

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

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

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

□

6.3 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 12. *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 9, 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 10-11, 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:

$$\begin{cases} \frac{b}{N}, & \text{by Theorem 10.} \\ min(\frac{a}{N}, \frac{(a+b)(N-1)}{N^2}), & \text{by Theorem 11.} \end{cases} \tag{19}$$

From Eq. 19, 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 11, 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 9, 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.

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 $O(kNn\log n)$ [39]. 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.

7 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).

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

Table 3. 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
Enron Graph	(sender, receiver, weight)	weight	6,066×5,699	151,738	2	Social Network Graph
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

Algorithm 2 Multiple Estimated Subtensors

Require: A D-Ordering π on a set of slices Q of tensor T
Ensure: Multiple estimated subtensors with guarantee on density

```
Initialization()                                   ▷ density measure ρ, build tensor
TS ← ∅,  S ← ∅
Number of estimated subtensors: mul ← 0
mul ← min(1 + n/(2N), 1 + N(N − 1))
for (j ← |Q|..1) do
    q ← π(j)
    S ← S ∪ q
    TS.add (S, ρ(S))
end for
Sort TS by descending order of density
return top-mul subtensors having highest density in TS
```

7.1 Experimental Setup

We used four widely-used density measures in our experiments: arithmetic average mass (ρ_a) [7]; geometric average mass (ρ_g) [7]; entry surplus (ρ_e) [45], with which the surplus parameter α was set to 1 as default; and suspiciousness (ρ_s) [18]. 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.

We implemented our approach based on the implementation used in the previous approaches [38, 39, 41]. 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].

7.2 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 11 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. [39].
- *Android*, which contains product reviews and rating metadata of applications for Android from Amazon [15].
- *Darpa*, which is a dataset collected by MIT Lincoln Lab to evaluate the performance of intrusion detection systems (IDSs) in cooperation with DARPA [24].
- *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.
- *Enron Graph* was modified from *Enron Emails* to have a form of a graph such that each instance has a structure of (*sender*, *receiver*, *weight*). The *weight* is a value representing the connections between a *sender* and a *receiver*. It is calculated as the sum of count of all instances in *Enron Emails* that have the same *sender* and *receiver*.

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

- *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 [31]. 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 [13].
- *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 [21].
- *YouTube*, which consists of the friendship connections between YouTube users [27].

The *Air Force* dataset was modified from the KDD Cup 1999 Data[5]. We kept fields (features) such that each instance has a structure of ($protocol\_type$, $service$, $flag$, src_bytes, dst_bytes, $count$, srv_count) as described in Table 3, while other fields were removed. The *Android* dataset was obtained from Stanford Network Analysis Project at this address[6]. The *Darpa* dataset was provided in the prior work, DenseAlert [41], and we downloaded the dataset at this address[7]. The *Enron Emails*, *NIPS Pubs*, and *LBNL-Network* were directly downloaded from an open source project, The Formidable Repository of Open Sparse Tensors and Tools (FROSTT) [43], at this address[8]. The *StackO* and *YouTube* were directly downloaded from The Koblenz Network Collection repository [21], and we got the datasets at this address[9]. The Kowiki and Enwiki datasets were downloaded from Wikipedia. We selected these datasets because of their diversity, and because they are widely used as benchmark datasets in the literature [39, 41]. A more detailed information about the datasets are listed in Table 3.

7.3 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%, 466% higher on the Darpa and EnronGraph datasets, respectively, 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. More specifically, in the top three estimated subtensors by MUST, M-Zoom, and M-Biz in evaluation of network attack detection on the Air Force dataset (Section 7.5), the density of the second and the third subtensors found by the compared methods drops significantly and are much lower than in our proposed method. The densities of the second and the third estimated subtensors found by

[3]https://dumps.wikimedia.org/
[4]https://nips.cc/
[5]http://kdd.ics.uci.edu/databases/kddcup99/kddcup.data.gz
[6]http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/ratings_ Apps_for_Android.csv
[7]http://www.cs.cmu.edu/ kijungs/codes/alert/data/darpa.zip
[8]http://frostt.io/tensors/
[9]http://konect.uni-koblenz.de/downloads/tsv

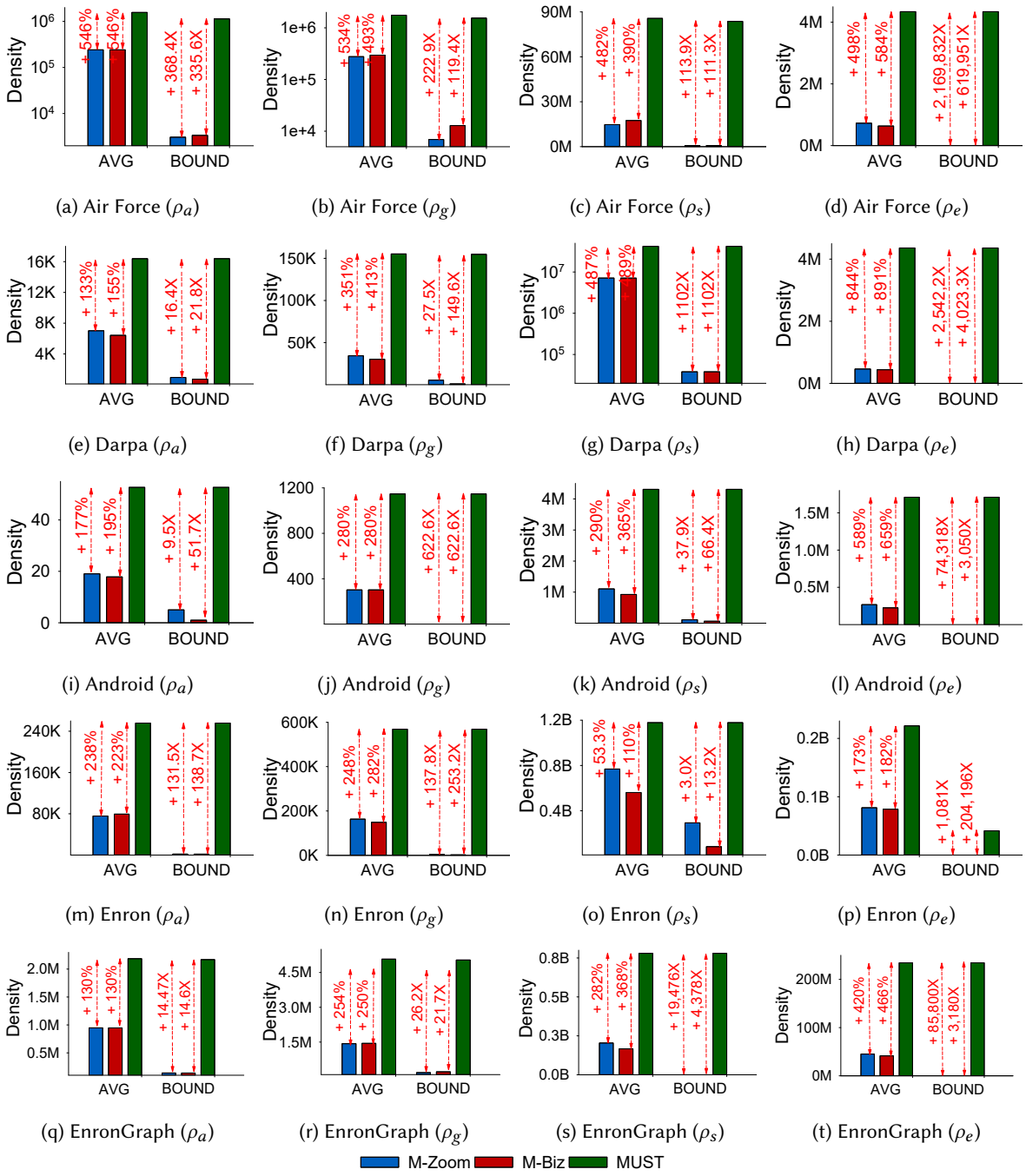
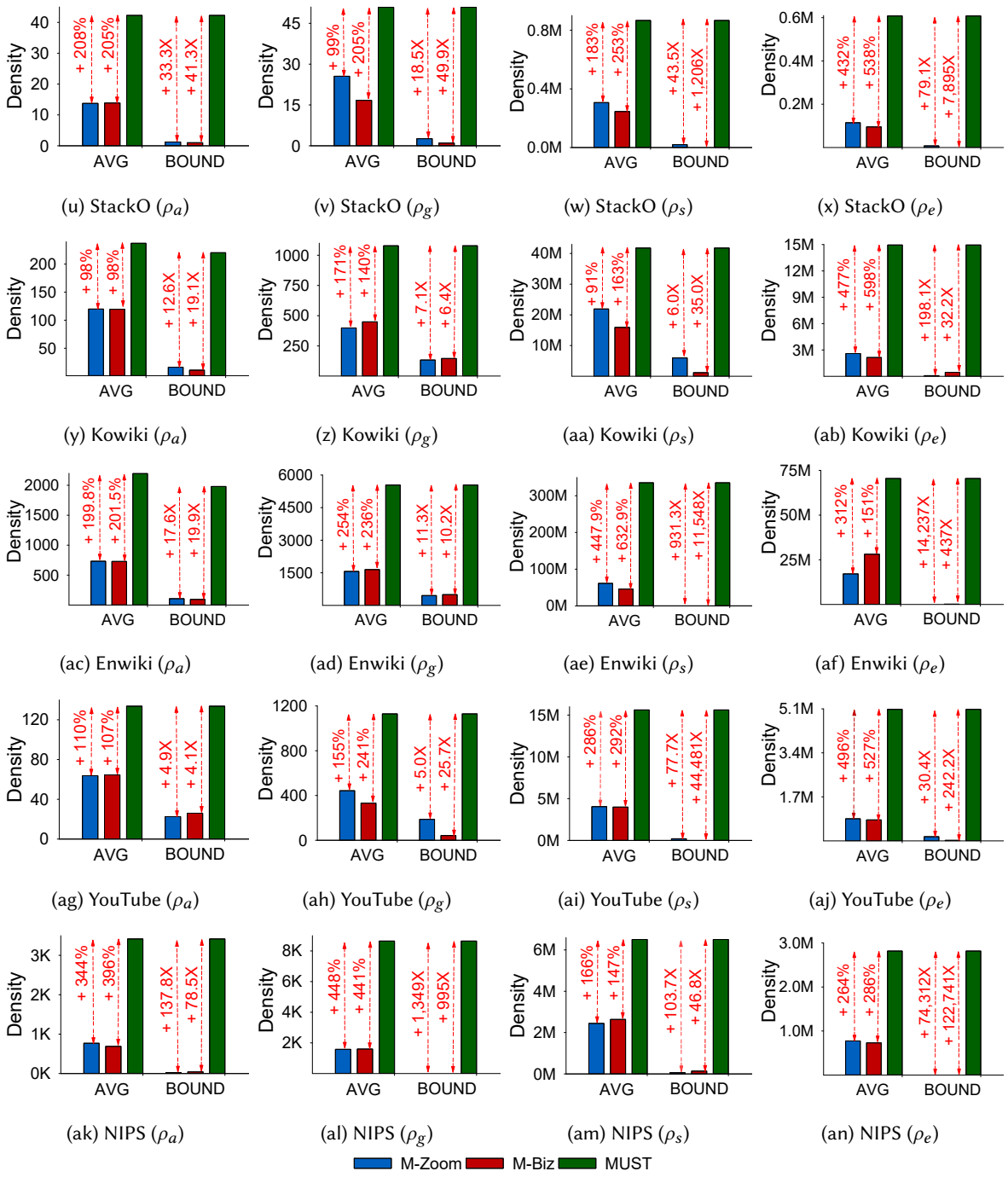
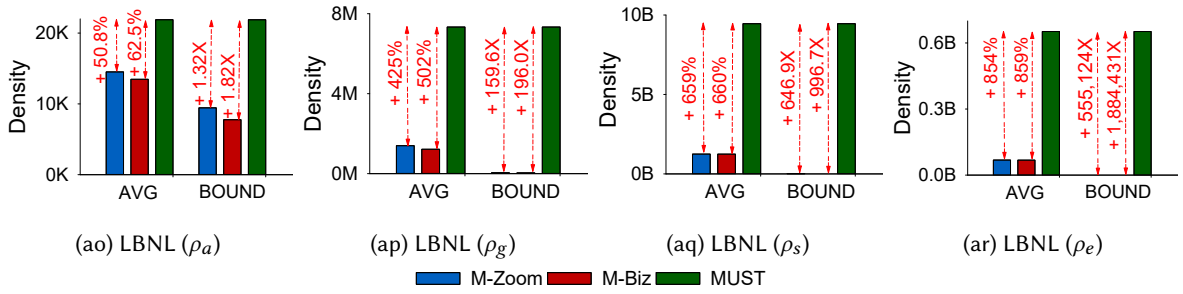


Fig. 2. Average and bound of density on datasets (K: thousand, M: million, B: billion). Best viewed in color and zoom mode.



K: thousand, M: million, B: billion. Best viewed in color and zoom mode.

Fig. 2. Average and bound of density on datasets (continued)



(ao) LBNL (ρ_a) (ap) LBNL (ρ_g) (aq) LBNL (ρ_s) (ar) LBNL (ρ_e)

K: thousand, M: million, B: billion. Best viewed in color and zoom mode.

Fig. 2. Average and bound of density on datasets (continued)

MUST are 7 times (~1,930,307/263,295) times and 29 times (~1,772,991/60,524) higher than the compared methods. The explanation for this result is that M-Zoom and M-Biz are not capable of providing a guarantee on the density of these estimated subtensors with respect to the original input data.

7.4 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 [38]. 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 [38, 41].

Table 4 shows the diversity of the top three estimated subtensors by MUST, M-Zoom and M-Biz. 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 and M-Biz, M-Zoom and M-Biz can find more diverse subtensors, which can be explained as follows. M-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. While in M-Biz, it uses the same structure as in M-Zoom, but differ in the way it finds the local optimal subtensor. M-Biz considers to add or remove a slice from a seed subtensor, and uses the same greedy search mechanism to get the local optimal subtensor. That leads to the same locally guarantee in the snapshot only as in M-Zoom. This is one of the drawbacks of M-Zoom M-Biz, and as discussed below (Section 7.5), the effectiveness of M-Zoom and M-Biz on network attack detection greatly drops with multiple subtensors.

Table 4. Diversity of estimated subtensors

Algorithm	Dataset	#	Volume*	Density	Diversity (Percentage)
MUST	Enwiki	1	4 ($1 \times 2 \times 2$)	2,397.6	36.2%
		2	20 ($1 \times 4 \times 5$)	2,375.7	
		3	9 ($1 \times 3 \times 3$)	2,355.9	
	Kowiki	1	8 ($2 \times 2 \times 2$)	273.0	37.2%
		2	80 ($4 \times 4 \times 5$)	258.5	
		3	64 ($4 \times 4 \times 4$)	240.5	
	Air Force	1	2 ($X_1 \times 2 \times 1 \times 1 \times 1$)	1,980,948	20.8%
		2	1 ($X_1 \times 1 \times 1 \times 1 \times 1$)	1,930,307	
		3	8 ($X_1 \times 2 \times 1 \times 2 \times 2$)	1,772,991	
M-Zoom	Enwiki	1	4 ($1 \times 2 \times 2$)	2,397.6	96.7%
		2	6 ($1 \times 2 \times 3$)	1,961.5	
		3	18 ($2 \times 3 \times 3$)	908.3	
	Kowiki	1	8 ($2 \times 2 \times 2$)	273.0	99.4%
		2	12 ($2 \times 2 \times 3$)	246.0	
		3	29,520 ($16 \times 41 \times 45$)	181.6	
	Air Force	1	2 ($X_1 \times 2 \times 1 \times 1 \times 1$)	1,980,948	70.8%
		2	1 ($X_1 \times 1 \times 1 \times 1 \times 1$)	263,295	
		3	4,320 ($X_2 \times 5 \times 4 \times 3 \times 3$)	60,524	
M-Biz	Enwiki	1	4 ($1 \times 2 \times 2$)	2,397.6	96.7%
		2	6 ($1 \times 2 \times 3$)	1,961.5	
		3	24 ($2 \times 3 \times 4$)	915.7	
	Kowiki	1	12 ($2 \times 2 \times 3$)	273.0	99.5%
		2	12 ($2 \times 2 \times 3$)	246.0	
		3	53,568 ($16 \times 62 \times 54$)	191.9	
	Air Force	1	2 ($X_1 \times 2 \times 1 \times 1 \times 1$)	1,980,948	71.2%
		2	1 ($X_1 \times 1 \times 1 \times 1 \times 1$)	263,295	
		3	5,400 ($X_2 \times 5 \times 5 \times 3 \times 3$)	60,699	

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

7.5 Effectiveness on Network Attack Detection

Extensive studies have shown that unexpected dense subregion (subgraph, subtensor) is a high sign of anomaly behaviors [17]. So, dense subregion detection is one of the efficient approaches and is widely-used in fraudulent

Table 5. Network attack detection on Air Force in the top five subtensors

	#	Volume	Density (ρ_a)	# Connections	# Attack Connections	# Normal Connections	# Ratio of Attack
MUST	1	$1 \times 1 \times 1 \times 2 \times 1 \times 1 \times 1$	1,980,948	2,263,941	2,263,941	0	100%
	2	$1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1$	1,930,307	1,930,307	1,930,307	0	100%
	3	$1 \times 1 \times 1 \times 2 \times 1 \times 2 \times 2$	1,772,991	2,532,845	2,532,845	0	100%
	4	$1 \times 1 \times 1 \times 2 \times 1 \times 1 \times 2$	1,764,860	2,269,106	2,269,106	0	100%
	5	$1 \times 1 \times 1 \times 2 \times 1 \times 2 \times 2$	1,612,741	2,534,308	2,534,308	0	100%
M-Zoom	1	$1 \times 1 \times 1 \times 2 \times 1 \times 1 \times 1$	1,980,948	2,263,941	2,263,941	0	100%
	2	$1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1$	263,295	263,295	263,295	0	100%
	3	$3 \times 4 \times 2 \times 5 \times 4 \times 3 \times 3$	60,524	207,513	56,433	151,080	27.2%
	4	$3 \times 3 \times 4 \times 4 \times 2 \times 154 \times 42$	33,901	1,026,723	1,007,762	18,961	98.15%
	5	$2 \times 4 \times 1 \times 7 \times 3 \times 13 \times 12$	16,467	98,806	42,961	55,845	43.5%
M-Biz	1	$1 \times 1 \times 1 \times 2 \times 1 \times 1 \times 1$	1,980,948	2,263,941	2,263,941	0	100%
	2	$1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1$	263,295	263,295	263,295	0	100%
	3	$3 \times 4 \times 2 \times 5 \times 5 \times 3 \times 3$	60,699	216,784	56,433	160,351	26.03%
	4	$2 \times 3 \times 4 \times 3 \times 1 \times 154 \times 42$	33,757	1,007,906	1,007,762	144	99.98%
	5	$2 \times 4 \times 1 \times 7 \times 3 \times 15 \times 12$	17,171	107,934	42,968	64,966	39.8%

behavior detection. In this section, we evaluate the efficiency of dense subregion detection in Network Attack Detection by performing extensive experiment on Air Force dataset. 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 and M-Biz. We analyze the five highest subtensors returned by M-Zoom, M-Biz, and MUST on Air Force, and then we compute how many connections in the estimated subtensors are normal activities or attack[10]. Table 5 shows the connections in the top five subtensors detected by MUST, M-Zoom, and M-Biz using arithmetic average mass (ρ_a) as the density metric. We observe that all connections in the top five 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 and M-Biz, they have the same result as MUST in the top two subtensors. However, in the three remaining subtensors, there are many normal connections that are wrong estimated by M-Zoom and M-Biz. For example, in the third subtensor estimated by M-Zoom, only 56,433 connections are attack, and 151,080 other connections are normal among 207,513 connections. So, the ratio of attack is only 27,2% with M-Zoom, and 26.03% with M-Biz. In real-world applications, particularly in the network attack detection problem, an attacker in fact does not necessarily act alone or join with all other attackers. In a real-world situation, an attacker normally joins several small groups of other attackers. Moreover, the suspicious behaviors of attackers might just only happen at just some certain

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

periods. Analyzing the user behaviors, detecting connections between attackers and detecting all the potential and suspicious indicators of attack are important. This is because any insight analysis of the suspiciousness will help to detect network attacks, and therefore protect the net from the attackers and fraudsters. The results of our experiments in the diversity and overlap analysis of the top three estimated subtensors in Table 4 on Airforce and the efficiency of our method for network attack detection in the top five estimated subtensors in Table 5 reflect our discussion of the meaningful and the important of multiple subregion estimation in practical applications. In other words, M-Zoom and M-Biz produce a high rate of false positive, which in turn means that MUST outperforms M-Zoom, M-Biz when used in the task of network attack detection, using the Air Force dataset.

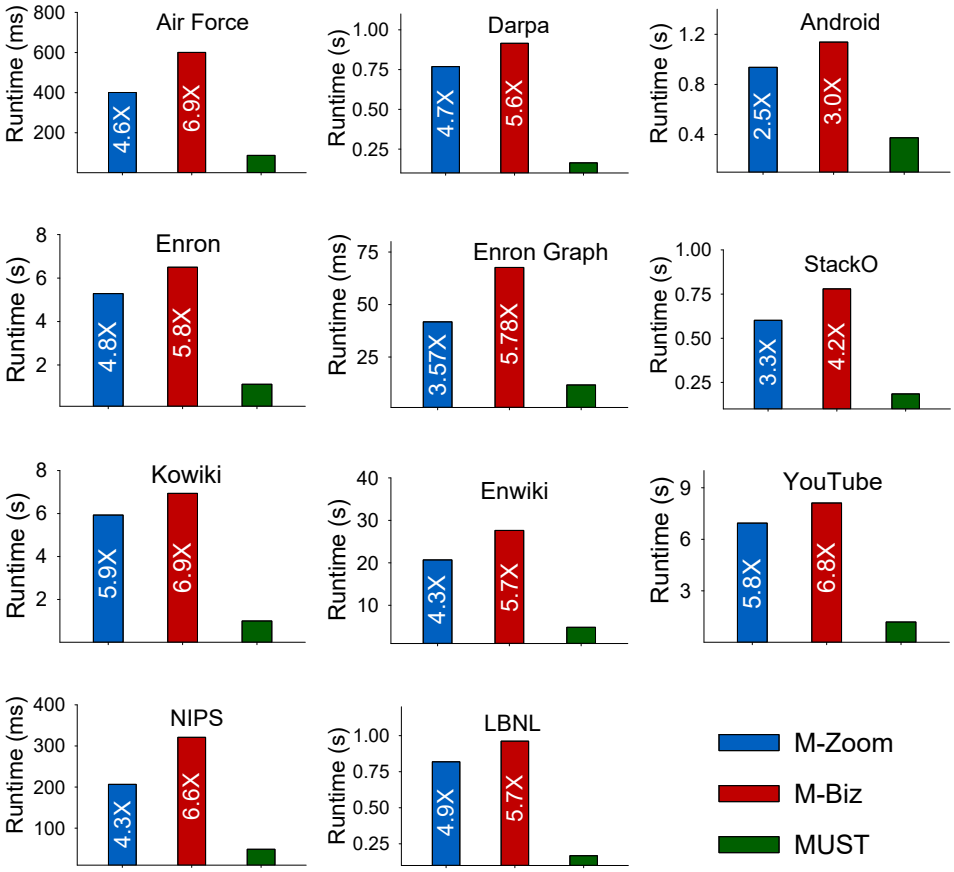


Fig. 3. Average runtime for a (sub)tensor on datasets. Best viewed in color.

7.6 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 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 6. 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.

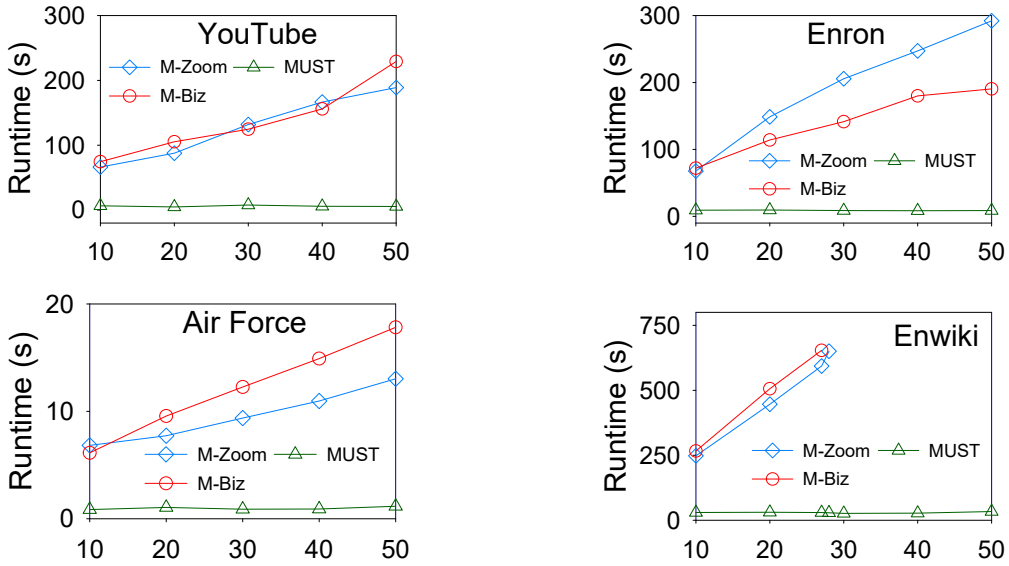


Fig. 4. Runtime while varying k. Best viewed in color.

7.7 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, YouTube, Air Force, and Enwiki 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. On Enwiki dataset, both M-Zoom and M-Biz run out of memory when the setting value of $k \geq 30$. 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 6.

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.

8 CONCLUSION

In this paper, we proposed a new technique to improve the task of dense subtensor and dense subgraph detection. As discussed, the contributions are both theoretical and practical. First, we developed concrete theoretical proofs for dense subtensors estimation in a tensor problem, as well as theoretical proofs for dense subgraph detection.

An important purpose of this was to provide a guarantee for a higher lower bound density of the estimation in both dense subtensor and subgraph detection. 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 has lower complexity 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 the dense subtensor detection problem, in terms of both efficiency and effectiveness. In conclusion, the proposed method is not only theoretically sound, but is also applicable for detecting dense subgraph and 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 finding multiple dense subgraphs in a graph data, and using it to solve event detection problems, such as change and anomaly detection.

REFERENCES

[1] Reid Andersen. 2010. A Local Algorithm for Finding Dense Subgraphs. *ACM Trans. Algorithms* 6, 4 (2010), 60:1–60:12.

[2] Reid Andersen and Kumar Chellapilla. 2009. Finding Dense Subgraphs with Size Bounds. In *Proceedings of WAW*. 25–37.

[3] Yuichi Asahiro, Refael Hassin, and Kazuo Iwama. 2002. Complexity of Finding Dense Subgraphs. *Discrete Appl. Math.* 121, 1 (2002), 15–26.

[4] Yuichi Asahiro, Kazuo Iwama, Hisao Tamaki, and Takeshi Tokuyama. 2000. Greedily Finding a Dense Subgraph. *J. Algorithms* 34, 2 (2000), 203–221.

[5] Oana Denisa Balalau, Francesco Bonchi, T-H. Hubert Chan, Francesco Gullo, and Mauro Sozio. 2015. Finding Subgraphs with Maximum Total Density and Limited Overlap. In *Proceedings of the 8th ACM WSDM*. 379–388.

[6] Yikun Ban, Xin Liu, Yitao Duan, Xue Liu, and Wei Xu. 2019. No Place to Hide: Catching Fraudulent Entities in Tensors. In *Proceedings of The Web Conference, WWW*. 83–93.

[7] Moses Charikar. 2000. Greedy Approximation Algorithms for Finding Dense Components in a Graph. In *Proceedings of APPROX*. 84–95.

[8] P. Comon. 2014. Tensors : A Brief Introduction. *IEEE Signal Process. Mag.* 31, 3 (2014), 44–53.

[9] Quang-Huy Duong, Heri Ramampiaro, and Kjetil Nørvåg. 2020. Multiple Dense Subtensor Estimation with High Density Guarantee. In *Proceedings of the 36th IEEE ICDE*. 637–648.

[10] Quang-Huy Duong, Heri Ramampiaro, and Kjetil Nørvåg. 2019. Sketching Streaming Histogram Elements using Multiple Weighted Factors. In *Proceedings of the 28th ACM CIKM*. 19–28.

[11] Esther Galbrun, Aristides Gionis, and Nikolaj Tatti. 2016. Top-k Overlapping Densest Subgraphs. *Data Min. Knowl. Discov.* 30, 5 (2016), 1134–1165.

[12] 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*. 721–732.

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

[14] A. V. Goldberg. 1984. *Finding a Maximum Density Subgraph.* Technical Report.

[15] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proceedings of the 25th The Web Conference, WWW*. 507–517.

[16] Daniel N. Holtmann-Rice, Benjamin S. Kunsberg, and Steven W. Zucker. 2018. Tensors, Differential Geometry and Statistical Shading Analysis. *J. Math. Imaging Vis.* 60 (2018), 968–992.

[17] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. 2016. FRAUDAR: Bounding Graph Fraud in the Face of Camouflage. In *Proceedings of the 22nd ACM SIGKDD*. 895–904.

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

[19] Samir Khuller and Barna Saha. 2009. On Finding Dense Subgraphs. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming: Part I (ICALP '09)*. 597–608.

[20] Tamara G. Kolda and Brett W. Bader. 2009. Tensor Decompositions and Applications. *SIAM Rev.* 51, 3 (2009), 455–500.

[21] Jérôme Kunegis. 2013. KONECT: The Koblenz Network Collection. In *Proceedings of the 22nd The Web Conference, WWW*. 1343–1350.

[22] Theodoros Lappas, Kun Liu, and Evimaria Terzi. 2009. Finding a Team of Experts in Social Networks. In *Proceedings of the 15th ACM SIGKDD*. 467–476.

[23] Xinsheng Li, Kasim Selçuk Candan, and Maria Luisa Sapino. 2018. M2TD: Multi-Task Tensor Decomposition for Sparse Ensemble Simulations. In *Proceedings of the 34th IEEE ICDE*. 1144–1155.

[24] Richard Lippmann, Joshua W. Haines, David J. Fried, Jonathan Korba, and Kumar Das. 2000. The 1999 DARPA Off-line Intrusion Detection Evaluation. *Comput. Netw.* 34, 4 (2000), 579–595.

[25] X. Liu, S. Ji, W. Glänzel, and B. De Moor. 2013. Multiview Partitioning via Tensor Methods. *IEEE Trans. Knowl. Data Eng.* 25, 5 (2013), 1056–1069.

[26] Koji Maruhashi, Fan Guo, and Christos Faloutsos. 2011. MultiAspectForensics: Pattern Mining on Large-Scale Heterogeneous Networks with Tensor Analysis. In *Proceedings of ASONAM*. 203–210.

[27] 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*. 29–42.

[28] Muhammad Anis Uddin Nasir, Aristides Gionis, Gianmarco De Francisci Morales, and Sarunas Girdzijauskas. 2017. Fully Dynamic Algorithm for Top-k Densest Subgraphs. In *Proceedings of the 2017 ACM CIKM*. 1817–1826.

[29] D. Nion and N. D. Sidiropoulos. 2010. Tensor Algebra and Multidimensional Harmonic Retrieval in Signal Processing for MIMO Radar. *IEEE Trans. Sig. Proc.* 58, 11 (2010), 5693–5705.

[30] Sejoon Oh, Namyong Park, Lee Sael, and U Kang. 2018. Scalable Tucker Factorization for Sparse Tensors - Algorithms and Discoveries. In *Proceedings of the 34th IEEE ICDE*. 1120–1131.

[31] Ruoming Pang, Mark Allman, Vern Paxson, and Jason Lee. 2006. The Devil and Packet Trace Anonymization. *ACM SIGCOMM Comput. Commun. Rev.* 36, 1 (2006), 29–38.

[32] Namyong Park, Sejoon Oh, and U. Kang. 2019. Fast and Scalable Method for Distributed Boolean Tensor Factorization. *The VLDB Journal* 28, 4 (2019), 549–574.

[33] Polina Rozenshtein, Aris Anagnostopoulos, Aristides Gionis, and Nikolaj Tatti. 2014. Event Detection in Activity Networks. In *Proceedings of the 20th ACM SIGKDD*. 1176–1185.

[34] Polina Rozenshtein, Francesco Bonchi, Aristides Gionis, Mauro Sozio, and Nikolaj Tatti. 2018. Finding Events in Temporal Networks: Segmentation Meets Densest-Subgraph Discovery. In *Proceedings of the 2018 IEEE ICDM*. 397–406.

[35] Polina Rozenshtein, Nikolaj Tatti, and Aristides Gionis. 2017. Finding Dynamic Dense Subgraphs. *ACM Trans. Knowl. Discov. Data* 11, 3 (2017), 27:1–27:30.

[36] Konstantinos Semertzidis, Evaggelia Pitoura, Evimaria Terzi, and Panayiotis Tsaparas. 2019. Finding Lasting Dense Subgraphs. *Data Min. Knowl. Discov.* 33, 5 (2019), 1417–1445.

[37] Preya Shah, Arian Ashourvan, Fadi Mikhail, Adam Pines, Lohith Kini, Kelly Oechsel, Sandhitsu R Das, Joel M Stein, Russell T Shinohara, Danielle S Bassett, Brian Litt, and Kathryn A Davis. 2019. Characterizing the Role of the Structural Connectome in Seizure Dynamics. *Brain* 142, 7 (2019), 1955–1972.

[38] Kijung Shin, Bryan Hooi, and Christos Faloutsos. 2016. M-Zoom: Fast Dense-Block Detection in Tensors with Quality Guarantees. In *Proceedings of ECML PKDD*. 264–280.

[39] Kijung Shin, Bryan Hooi, and Christos Faloutsos. 2018. Fast, Accurate, and Flexible Algorithms for Dense Subtensor Mining. *ACM Trans. Knowl. Discov. Data* 12, 3 (2018), 28:1–28:30.

[40] Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos. 2017. D-Cube: Dense-Block Detection in Terabyte-Scale Tensors. In *Proceedings of the 10th ACM WSDM*. 681–689.

[41] Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos. 2017. DenseAlert: Incremental Dense-Subtensor Detection in Tensor Streams. In *Proceedings of the 23rd ACM SIGKDD*. 1057–1066.

[42] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos. 2017. Tensor Decomposition for Signal Processing and Machine Learning. *IEEE Trans. Sig. Proc.* 65, 13 (2017), 3551–3582.

[43] Shaden Smith, Jee W. Choi, Jiajia Li, Richard Vuduc, Jongsoo Park, Xing Liu, and George Karypis. 2017. FROSTT: The Formidable Repository of Open Sparse Tensors and Tools. http://frostt.io/

[44] Nikolaj Tatti and Aristides Gionis. 2015. Density-friendly Graph Decomposition. In *Proceedings of the 24th The Web Conference, WWW*. 1089–1099.

[45] 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*. 104–112.

[46] Fan Yang, Fanhua Shang, Yuzhen Huang, James Cheng, Jinfeng Li, Yunjian Zhao, and Ruihao Zhao. 2017. LFTF: A Framework for Efficient Tensor Analytics at Scale. *Proceedings of the VLDB Endowment* 10, 7 (2017), 745–756.

[47] W. Zhang, Z. Lin, and X. Tang. 2011. Learning Semi-Riemannian Metrics for Semisupervised Feature Extraction. *IEEE Trans. Knowl. Data Eng.* 23, 4 (2011), 600–611.

[48] Shuo Zhou, Nguyen Xuan Vinh, James Bailey, Yunzhe Jia, and Ian Davidson. 2016. Accelerating Online CP Decompositions for Higher Order Tensors. In *Proceedings of the 22nd ACM SIGKDD*. 1375–1384.