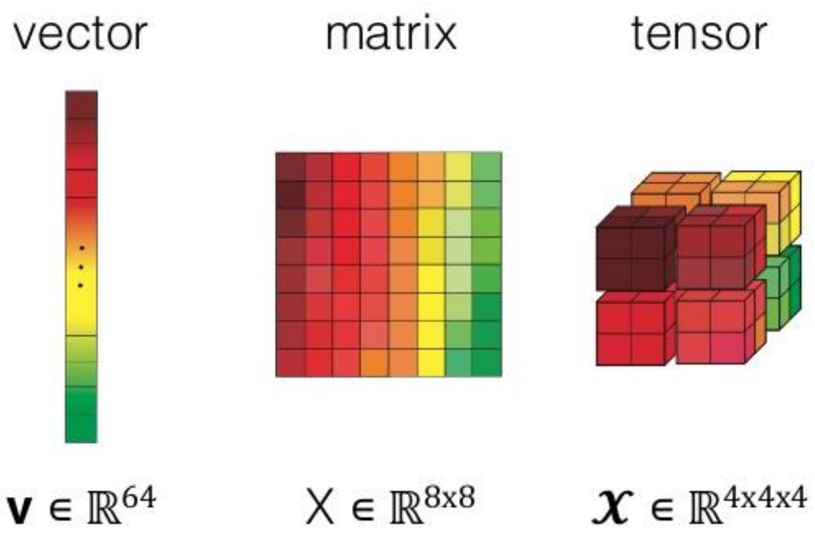
Supported Data Type

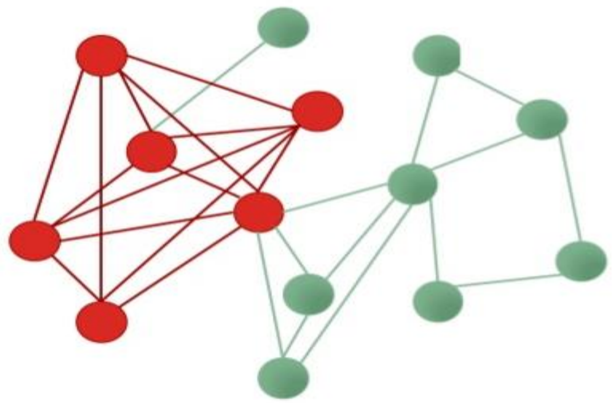
Tensor:

Is a multidimensional array
- 1-way tensor: vector
- 2-way tensor: matrix
- N-way: higher orders
- Size of tensor is number of slices (n)
- m is size of the densest



Graph :

An undirected graph G is a set of vertices V and edges E
- At each vertex and on edge, there is a weight
- Size of graph: number of vertices (n)
- m is size of the densest



The general densest-subgraph(subtensor) problem

Tensor

- Given a N-Way Tensor T
and a density measure $f : 2^V \rightarrow R$

- Find set of slices $Q \subseteq V$
that maximizes f(S)

Graph

- Given an undirected graph $G = (V; E)$
and a density measure $f : 2^V \rightarrow R$

- Find set of vertices $S \subseteq V$
that maximizes f(S)

Applications of finding dense subgraphs (subtensors)

Communities and spam link farms [1]

Graph visualization [2]

Real-time story identification [3]

Regulatory motif detection in DNA [4]

Finding correlated genes [5]

Epilepsy prediction [6]

Anomaly/Fraud Detection [7-11]

And Others

Source: https://users.ics.aalto.fi/gionis/dense.pdf

[1] Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. (1999). Trawling the Web for emerging cyber-communities. *Computer Networks*, 31(11–16):1481–1493.

[2] Alvarez-Hamelin, J. I., Dall'Asta, L., Barrat, A., and Vespignani, A. (2005). Large scale networks fingerprinting and visualization using the *k*-core decomposition. In *NIPS.*

[3] Angel, A., Koudas, N., Sarkas, N., and Srivastava, D. (2012). Dense Subgraph Maintenance under Streaming Edge Weight Updates for Real-time Story Identification.

[4] Fratkin, E., Naughton, B. T., Brutlag, D. L., and Batzoglou, S. (2006). MotifCut: regulatory motifs finding with maximum density subgraphs. *Bioinformatics*, 22(14) .

[5] Zhang, B. and Horvath, S. (2005). A general framework for weighted gene co-expression network analysis.*Statistical applications in genetics and molecular biology*, 4(1):1128.

[6] Lasemidis, L. D., Shiau, D.-S., Chaovalitwongse, W. A., Sackellares,J. C., Pardalos, P. M., Principe, J. C., Carney, P. R., Prasad, A.,Veeramani, B., and Tsakalis, K. (2003). Adaptive epileptic seizure prediction system. *IEEE Transactions on Biomedical Engineering*, 50(5).

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

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

[9] Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos. 2017. DenseAlert: Incremental Dense-Subtensor Detection in Tensor Streams. In *KDD*. 1057–1066.

[10] Hooi, B., Song, H.A., Beutel, A., Shah, N., Shin, K., Faloutsos, C.: Fraudar: bounding graph fraud in the face of camouflage. In: KDD (2016).

[11] Yikun , Xin Liu , Ling Huang, Yitao Duan, Xue Liu , Wei Xu: No Place to Hide: Catching Fraudulent Entities in Tensors. In WWW (2019).

State-of-the-art algorithms for densest subgraph mining

1. Goldberg's algorithm [1]

- To find the densest subgraph: generally Np-hard problem

- Is Polynomial algorithm, highly complexity

- $O(nm)$ time for one min-cut computation

- Not scalable for large graphs (millions of vertices / edges)

2. Charikar's algorithm [2]

- Faster algorithm

- Greedy and simple to implement

- Approximation algorithm : factor-**2** approximation algorithm

- For a polynomial problem but faster and easier to implement than the exact algorithm

[1] Goldberg, A. V. (1984). Finding a maximum density subgraph. Technical report.
[2] Charikar, M. (2000). Greedy approximation algorithms for finding dense components in a graph. In *APPROX*.

Motivation

1. Most of the existing applications adapt Charikar's algorithm:

- To give an approximation algorithm
- For a specific application
- With a guarantee as in Charikar's algorithm (**2**-approximation)

2. No work for a better approximation (>2) (to the best of my knowledge)

3. Question: can we give a better approximation guarantee (>2) with linear time complexity as in Charikar's?

- Focus on providing a better guarantee
- Is a theoretical work, not aim at any specific application
- Give guarantee on both tensor and graph data

So Far

- What I have done: I use the same mechanism as Charikar's but I try to prove something new

Characteristic	Goldberg	Charikar's Based Algorithms	Ours
Tensor support			☑
Graph support	☑	☑	☑
Complexity	Polynomial	Near Linear Time	Near Linear Time
Type	Exact	Approximation	Approximation
α- approximation (tensor)		$\frac{1}{N}$	$\text{Max}(\frac{1}{N}(1+\frac{N-1}{\sqrt{n}}), \frac{1}{N}(1+\frac{N-1}{m}))$ or $\frac{1}{N}(1+\frac{N-1}{\min(m,\sqrt{n})})$
α- approximation (graph)	1	$\frac{1}{2}$	$\frac{2m+x}{2(m+x)} \geq \frac{1}{2}(1+\frac{1}{n})$

- Now the improvement is not much
- A possible way is to: find a new proof or new cut mechanism for a better α