New Algorithm for Dense Subgraphs in Multilayer Networks: S-core Methodology

Research Background and Significance

In the study of complex systems, Multilayer Networks (ML) have become an important tool for modeling multidimensional interactions. This model has wide applications in fields such as social networks, biological systems, and financial markets. For example, in social network analysis, interactions between individuals may involve multiple dimensions such as social, professional, and interests. The concept of multilayer networks provides a more comprehensive framework to capture these complex relational structures.

Research Questions and Challenges

Existing research faces the following key challenges when extending the concept of core structures to multilayer networks:

1. **Overly strict constraints**: Forcing nodes to meet degree constraints in predefined subset layers may lead to the neglect of some important dense structures.
2. **High computational complexity**: Existing algorithms often require exponential time complexity, making them difficult to apply to large-scale real-world networks.
3. **Difficulty in integrating interlayer information**: Different types of relationships may have varying importance in practical applications, and effectively integrating this information remains an open problem.

Innovative Method: S-core

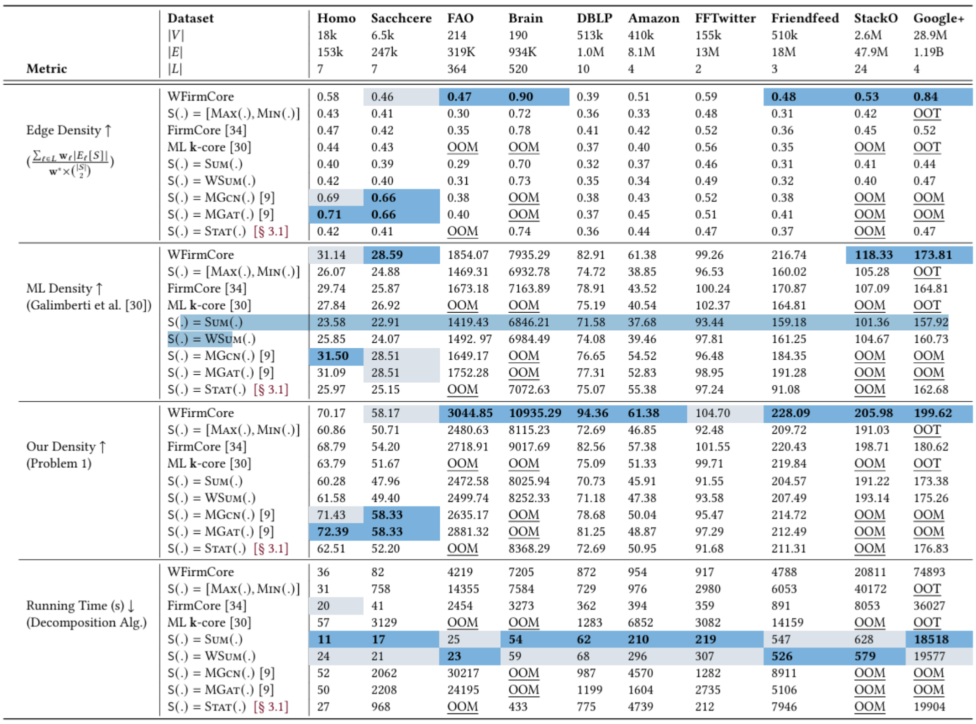
To address the above issues, this study proposes the S-core method, a novel unified framework for dense structure analysis in multilayer networks. The core idea of S-core is to introduce an abstraction function S(.), which maps each node's multidimensional vector to a lower-dimensional space, thereby effectively reducing computational complexity and avoiding overly strict constraints.

Specifically, given a d-dimensional threshold vector k, a (k, S)-core is defined as a maximal subgraph where the summary degree of each node u in the subgraph is at least k. This definition provides greater flexibility, accommodating different density requirements across layers while ensuring the algorithm's efficiency.

Method Evaluation and Application Value

The performance of different S-core variants on various datasets is as follows:

Table 1: The density of the densest different variants of S-cores with respect to edge density, ML density [30], and our density. The best (resp. the second best) result is highlighted in blue (resp. gray). OOT: Time exceeds 24 hours, OOM: Memory exeeds 100 GB.



	Dataset	**Homo**	**Sacchcere**	**FAO**	**Brain**	**DBLP**	**Amazon**	**FFTwitter**	**Friendfeed**	**StackO**	**Google+**
	$\lvert V\rvert$	18k	6.5k	214	190	513k	410k	155k	510k	2.6M	28.9M
	$\lvert E\rvert$	153k	247k	319K	934K	1.0M	8.1M	13M	18M	47.9M	1.19B
Metric	$\lvert L\rvert$	7	7	364	520	10	4	2	3	24	4
Edge Density ↑ $(\frac{\sum_{\ell \in L} \mathbf{w}_\ell \lvert E_\ell[S]\rvert}{\mathbf{w}^* \times \binom{\lvert S\rvert}{2}})$	WFirmCore	0.58	0.46	**0.47**	**0.90**	0.39	0.51	0.59	**0.48**	**0.53**	**0.84**
	S(.) = [Max(.), Min(.)]	0.43	0.41	0.30	0.72	0.36	0.33	0.48	0.31	0.42	OOT
	FirmCore [34]	0.47	0.42	0.35	0.78	0.41	0.42	0.52	0.36	0.45	0.52
	ML **k**-core [30]	0.44	0.43	OOM	OOM	0.37	0.40	0.56	0.35	OOM	OOT
	S(.) = Sum(.)	0.40	0.39	0.29	0.70	0.32	0.37	0.46	0.31	0.41	0.44
	S(.) = WSum(.)	0.42	0.40	0.31	0.73	0.35	0.34	0.49	0.32	0.40	0.47
	S(.) = MGcn(.) [9]	0.69	**0.66**	0.38	OOM	0.38	0.43	0.52	0.38	OOM	OOM
	S(.) = MGat(.) [9]	**0.71**	**0.66**	0.40	OOM	0.37	0.45	0.51	0.41	OOM	OOM
	S(.) = Stat(.) [§ 3.1]	0.42	0.41	OOM	0.74	0.36	0.44	0.47	0.37	OOM	0.47
ML Density ↑ (Galimberti et al. [30])	WFirmCore	31.14	**28.59**	1854.07	7935.29	82.91	61.38	99.26	216.74	**118.33**	**173.81**
	S(.) = [Max(.), Min(.)]	26.07	24.88	1469.31	6932.78	74.72	38.85	96.53	160.02	105.28	OOT
	FirmCore [34]	29.74	25.87	1673.18	7163.89	78.91	43.52	100.24	170.87	107.09	164.81
	ML **k**-core [30]	27.84	26.92	OOM	OOM	75.19	40.54	102.37	164.81	OOM	OOT
	S(.) = Sum(.)	23.58	22.91	1419.43	6846.21	71.58	37.68	93.44	159.18	101.36	157.92
	S(.) = WSum(.)	25.85	24.07	1492. 97	6984.49	74.08	39.46	97.81	161.25	104.67	160.73
	S(.) = MGcn(.) [9]	**31.50**	28.51	1649.17	OOM	76.65	54.52	96.48	184.35	OOM	OOM
	S(.) = MGat(.) [9]	31.09	28.51	1752.28	OOM	77.31	52.83	98.95	191.28	OOM	OOM
	S(.) = Stat(.) [§ 3.1]	25.97	25.15	OOM	7072.63	75.07	55.38	97.24	91.08	OOM	162.68
Our Density ↑ (Problem 1)	WFirmCore	70.17	58.17	**3044.85**	**10935.29**	**94.36**	**61.38**	104.70	**228.09**	**205.98**	**199.62**
	S(.) = [Max(.), Min(.)]	60.86	50.71	2480.63	8115.23	72.69	46.85	92.48	209.72	191.03	OOT
	FirmCore [34]	68.79	54.20	2718.91	9017.69	82.56	57.38	101.55	220.43	198.71	180.62
	ML **k**-core [30]	63.79	51.67	OOM	OOM	75.09	51.33	99.71	219.84	OOM	OOT
	S(.) = Sum(.)	60.28	47.96	2472.58	8025.94	70.73	45.91	91.55	204.57	191.22	173.38
	S(.) = WSum(.)	61.58	49.40	2499.74	8252.33	71.18	47.38	93.58	207.49	193.14	175.26
	S(.) = MGcn(.) [9]	71.43	**58.33**	2635.17	OOM	78.68	50.04	95.47	214.72	OOM	OOM
	S(.) = MGat(.) [9]	**72.39**	**58.33**	2881.32	OOM	81.25	48.87	97.29	212.49	OOM	OOM
	S(.) = Stat(.) [§ 3.1]	62.51	52.20	OOM	8368.29	72.69	50.95	91.68	211.31	OOM	176.83
Running Time (s) ↓ (Decomposition Alg.)	WFirmCore	36	82	4219	7205	872	954	917	4788	20811	74893
	S(.) = [Max(.), Min(.)]	31	758	14355	7584	729	976	2980	6053	40172	OOT
	FirmCore [34]	20	41	2454	3273	362	394	359	891	8053	36027
	ML **k**-core [30]	57	3129	OOM	OOM	1283	6852	3082	14159	OOM	OOT
	S(.) = Sum(.)	**11**	**17**	25	**54**	**62**	**210**	**219**	547	628	**18518**
	S(.) = WSum(.)	24	21	**23**	59	68	296	307	**526**	**579**	19577
	S(.) = MGcn(.) [9]	52	2062	30217	OOM	987	4570	1282	8911	OOM	OOM
	S(.) = MGat(.) [9]	50	2208	24195	OOM	1199	1604	2735	5106	OOM	OOM
	S(.) = Stat(.) [§ 3.1]	27	968	OOM	433	775	4739	212	7946	OOM	19904

The main advantages of the S-core method are:

1. **Unification**: It can unify and extend various existing dense subgraph models.
2. **Computational efficiency**: The introduction of the summary function significantly reduces computational complexity.
3. **Model flexibility**: It can more accurately capture interaction patterns in multilayer networks.

The S-core method excels in identifying the densest subgraphs and modeling user engagement, offering high computational efficiency and the ability to more precisely reflect the structural characteristics of complex networks.

Comparative Analysis of Dense Subgraph Algorithms

1. Traditional Single-Layer Network Dense Subgraph Algorithms

1.1 k-core Decomposition

- **Definition**: A k-core is a maximal subgraph in which each vertex has a degree of at least k.
- **Advantages**: High computational efficiency with time complexity O(m+n), where m is the number of edges and n is the number of nodes. It reveals the hierarchical structure of the network.
- **Disadvantages**: Considers only the degree, possibly overlooking other important structural features.

1.2 k-truss Decomposition

- **Definition**: A k-truss is a maximal subgraph in which each edge participates in at least k-2 triangles.
- **Advantages**: Better captures local dense structures than k-core. It can discover higher-order network structures.
- **Disadvantages**: Higher computational complexity, typically O(m^1.5).

1.3 Maximum Clique Problem

- **Definition**: Finding the largest complete subgraph in the network.
- **Advantages**: Provides the strictest definition of a dense subgraph.
- **Disadvantages**: NP-hard problem with extremely high computational cost in large-scale networks. Sensitive to noise, making it unsuitable for real-world sparse networks.

2. Multilayer Network Dense Subgraph Algorithms

2.1 Vector Threshold Method

- **Definition**: Extends the k-core concept to multilayer networks, requiring each node to meet specific degree thresholds in each layer.
- **Advantages**: Intuitively extends the single-layer network concept.
- **Disadvantages**: High computational complexity, typically requiring exponential time. Overly strict constraints may miss some important dense structures.

2.2 Multilayer k-core Decomposition

- **Definition**: Finds maximal subgraphs in multilayer networks that meet multidimensional constraints.
- **Advantages**: Can capture complex structures in multilayer networks.
- **Disadvantages**: High algorithmic complexity, making it difficult to apply to large-scale networks. Inflexible handling of the importance of different layers.

2.3 S-core Method (Proposed in this study)

- **Definition**: Introduces an abstraction function S(.) to map nodes' multidimensional vectors to lower-dimensional space and defines dense subgraphs based on this.
- **Advantages**: High computational efficiency, avoiding exponential time complexity. Provides a more flexible definition of dense structures, accommodating the importance of different layers. Unifies various existing dense subgraph models.
- **Disadvantages**: The choice of the summary function may affect the quality of the results. Specific applications may require parameter adjustments.

Research Prospects

The proposal of the S-core method opens up new research directions for multilayer network analysis. Future work can focus on the following aspects:

1. **Optimization of the abstraction function**: Explore more types of abstraction functions to meet the needs of different application scenarios.

2. **Dynamic network analysis**: Extend S-core to the analysis of temporal multilayer networks.
3. **Large-scale parallel algorithms**: Develop distributed algorithms based on S-core to address the challenges of ultra-large-scale networks.
4. **Interdisciplinary applications**: Explore the potential applications of S-core in fields such as social sciences and bioinformatics.

In summary, the S-core method provides a powerful and flexible tool for complex network analysis, promising significant impacts across multiple fields. This study not only advances the theory of multilayer networks but also provides new analytical perspectives and methodological support for practical applications.

```
[Paper link](https://arxiv.org/abs/2406.13734)
```