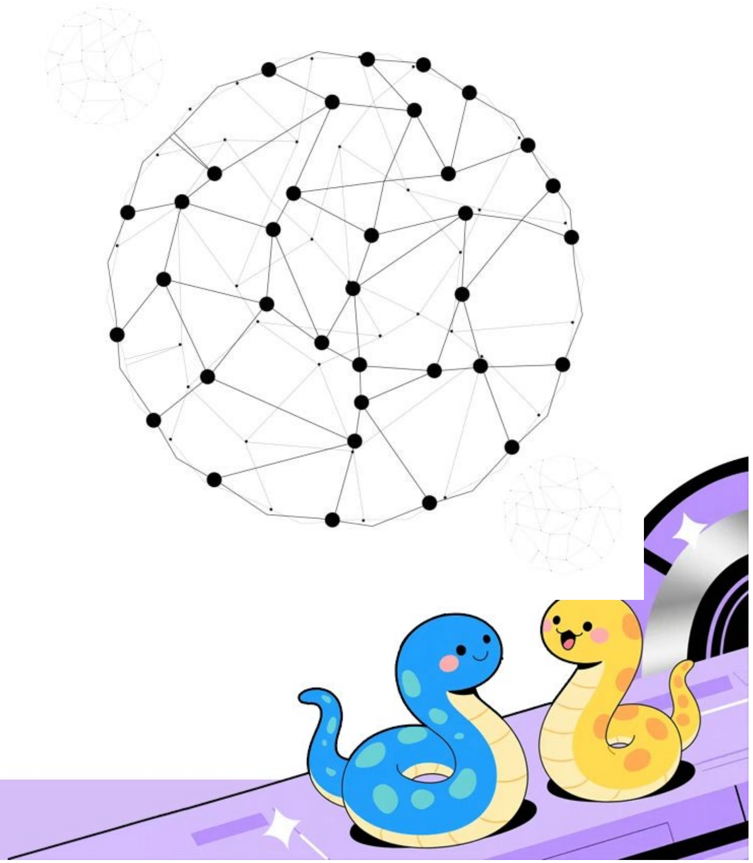
EasyGraph: 面向多学科的高性能网络结构分析工具箱

陈阳
计算与智能创新学院
复旦大学
chenyang@fudan.edu.cn

Content

- ◆ Background & Motivation
- ◆ EasyGraph
- ◆ Features of EasyGraph
- ◆ Ease-of-Use APIs and Use Cases
- ◆ Performance of EasyGraph
- ◆ Key Elements of EasyGraph 2.0
- ◆ Conclusion



Network Data is Everywhere

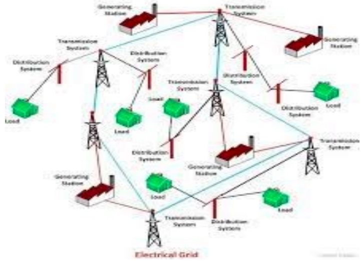
Networks are powerful tools for representing the relationships and interactions between entities in various disciplines

social network

NDUFS5
ENSG00000237495
ENSG00000234651
ENSG00000237431
NME4
ALPL
MACF1
G5
TMEM8A

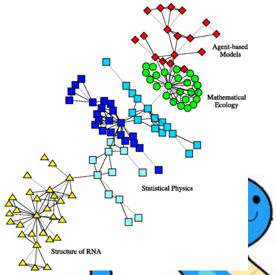
protein interaction network

brain network



power grid network

metro network



co-authorship network

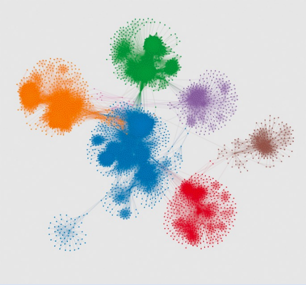
Social Network

- **Social network**, a type of complex network, includes people and their interactions
- **Social Network Analysis (SNA)** is a methodology of analyzing social structures with network analytic techniques

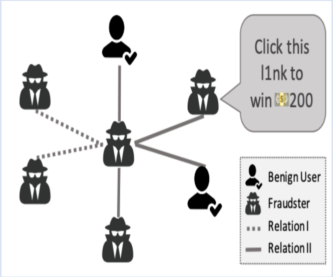
social network platforms

top-10 influencers

viral marketing



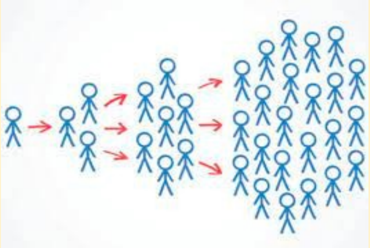
community detection



fraud detection

Structural Hole Spanner Detection

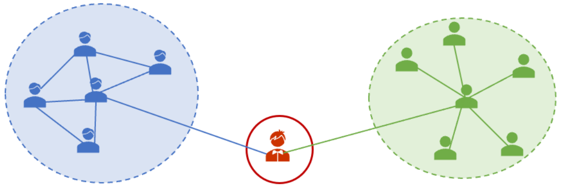
- Structural hole spanners (SH Spanners) refer to the individuals act as brokers among communities
- SH Spanners occupy unique positions, so they facilitate information and resources from different groups



information propogation

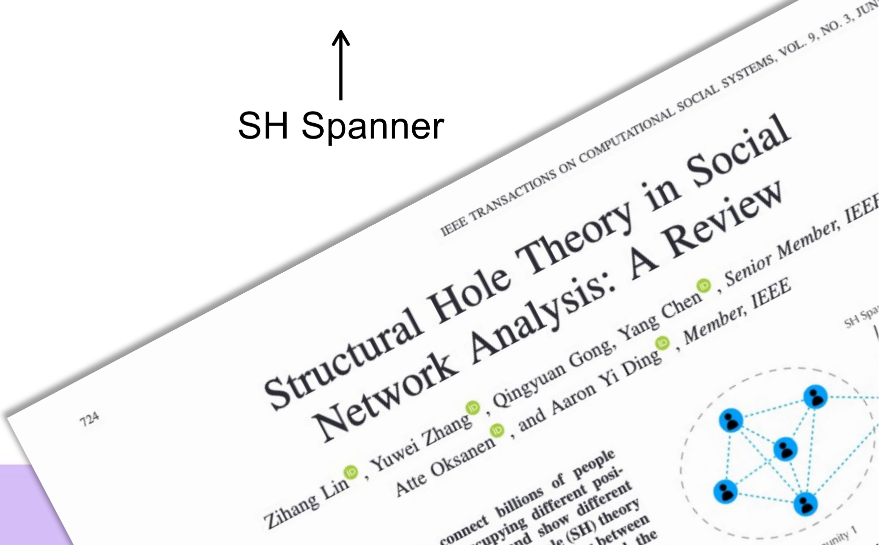
public opinion mining

social innovation



↑

SH Spanner



IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, VOL. 9, NO. 3, JUN

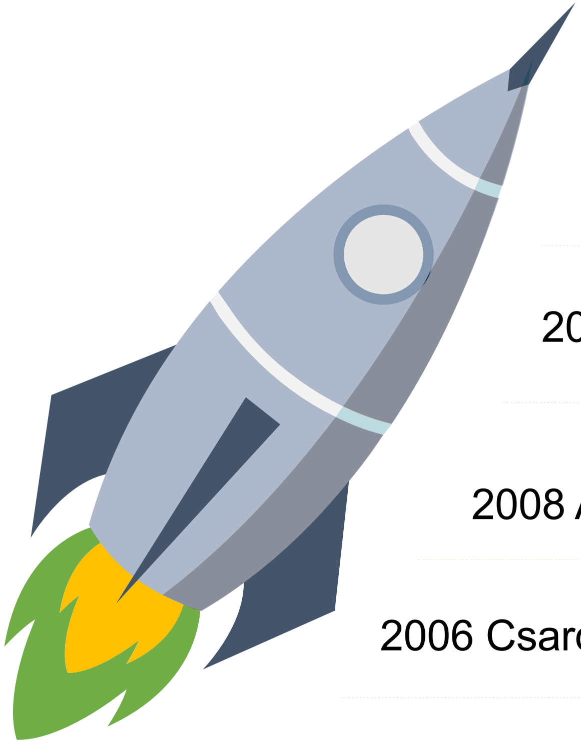
724

Structural Hole Theory in Social Network Analysis: A Review

Zihang Lin, Yuwei Zhang, Qingyuan Gong, Yang Chen, Senior Member, IEEE, Atte Oksanen, and Aaron Yi Ding, Member, IEEE

connect billions of people
occupying different posi-
and show different
le (SH) theory
between
the

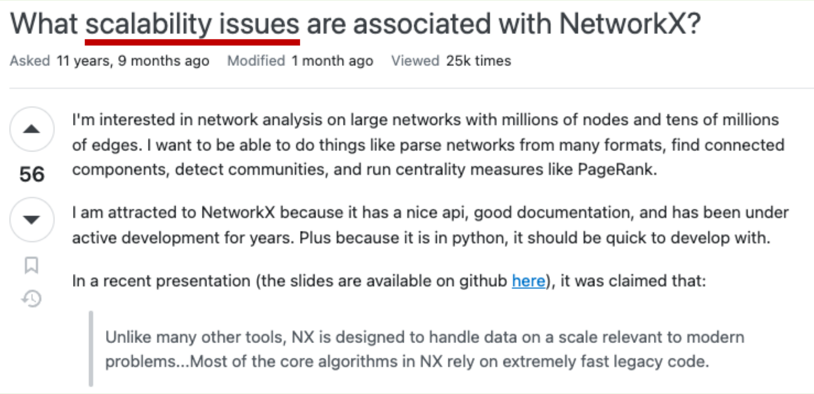
Tools for Network Analysis



2016 Peixoto	graph-tool	**statistical network analys**
2014 Leskovec et al.	SNAP	**efficient network analysis**
2009 Bastian et al.	Gephi makes graphs handy	**network visualization**
2008 Aric et al.	NetworkX Network Analysis in Python	**basic network analysis**
2006 Csardi et al.	igraph	**efficient network analysis**
2003 Shannon et al.	Cytoscape	**network visualization**

Discussions from Online Forums

Stack Overflow



What scalability issues are associated with NetworkX?

Asked 11 years, 9 months ago Modified 1 month ago Viewed 25k times

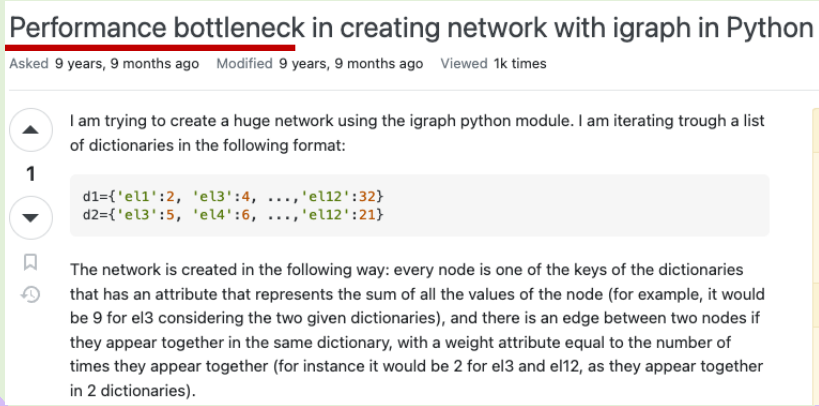
56

I'm interested in network analysis on large networks with millions of nodes and tens of millions of edges. I want to be able to do things like parse networks from many formats, find connected components, detect communities, and run centrality measures like PageRank.

I am attracted to NetworkX because it has a nice api, good documentation, and has been under active development for years. Plus because it is in python, it should be quick to develop with.

In a recent presentation (the slides are available on github here), it was claimed that:

> Unlike many other tools, NX is designed to handle data on a scale relevant to modern problems...Most of the core algorithms in NX rely on extremely fast legacy code.



Performance bottleneck in creating network with igraph in Python

Asked 9 years, 9 months ago Modified 9 years, 9 months ago Viewed 1k times

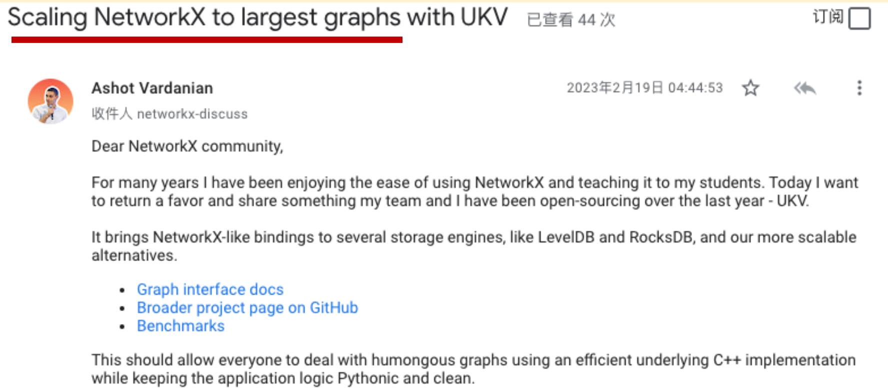
1

I am trying to create a huge network using the igraph python module. I am iterating trough a list of dictionaries in the following format:

```
d1={'el1':2, 'el3':4, ...,'el12':32}
d2={'el3':5, 'el4':6, ...,'el12':21}
```

The network is created in the following way: every node is one of the keys of the dictionaries that has an attribute that represents the sum of all the values of the node (for example, it would be 9 for el3 considering the two given dictionaries), and there is an edge between two nodes if they appear together in the same dictionary, with a weight attribute equal to the number of times they appear together (for instance it would be 2 for el3 and el12, as they appear together in 2 dictionaries).

Google Groups



Scaling NetworkX to largest graphs with UKV

已查看 44 次 订阅 □

Ashot Vardanian 2023年2月19日 04:44:53

收件人 networkx-discuss

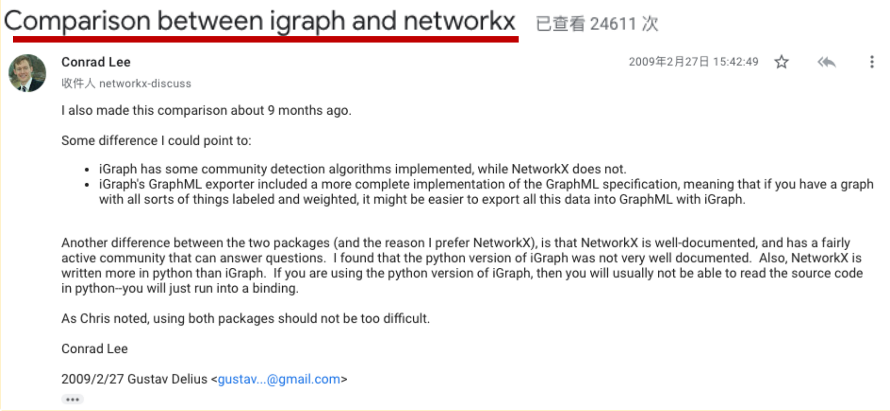
Dear NetworkX community,

For many years I have been enjoying the ease of using NetworkX and teaching it to my students. Today I want to return a favor and share something my team and I have been open-sourcing over the last year - UKV.

It brings NetworkX-like bindings to several storage engines, like LevelDB and RocksDB, and our more scalable alternatives.

- Graph interface docs
- Broader project page on GitHub
- Benchmarks

This should allow everyone to deal with humongous graphs using an efficient underlying C++ implementation while keeping the application logic Pythonic and clean.



Comparison between igraph and networkx

已查看 24611 次

Conrad Lee 2009年2月27日 15:42:49

收件人 networkx-discuss

I also made this comparison about 9 months ago.

Some difference I could point to:

- iGraph has some community detection algorithms implemented, while NetworkX does not.
- iGraph's GraphML exporter included a more complete implementation of the GraphML specification, meaning that if you have a graph with all sorts of things labeled and weighted, it might be easier to export all this data into GraphML with iGraph.

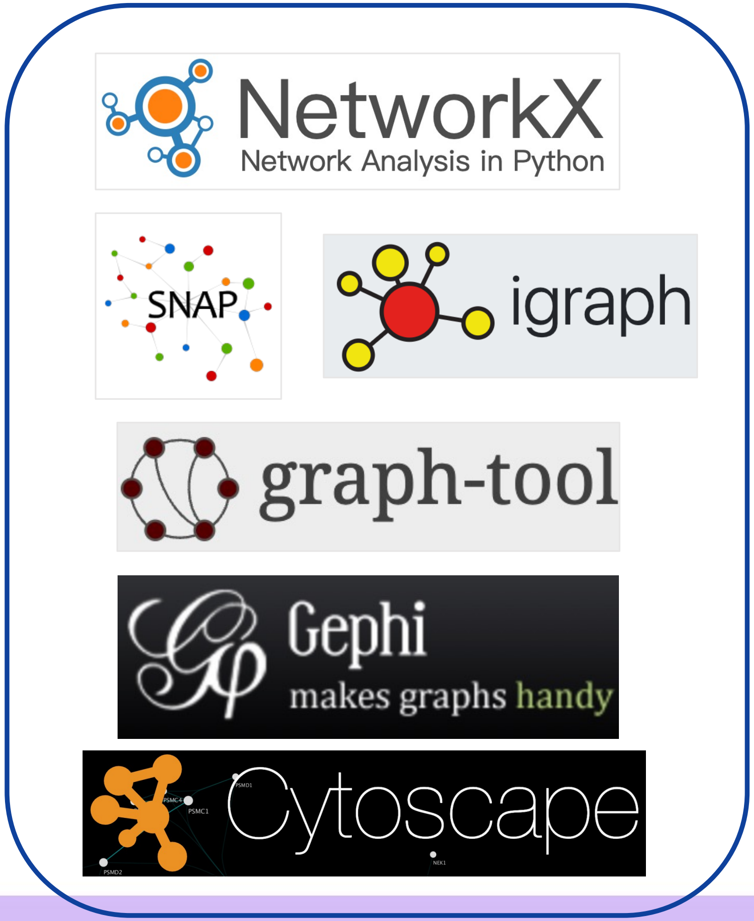
Another difference between the two packages (and the reason I prefer NetworkX), is that NetworkX is well-documented, and has a fairly active community that can answer questions. I found that the python version of iGraph was not very well documented. Also, NetworkX is written more in python than iGraph. If you are using the python version of iGraph, then you will usually not be able to read the source code in python--you will just run into a binding.

As Chris noted, using both packages should not be too difficult.

Conrad Lee

2009/2/27 Gustav Delius <gustav...@gmail.com>

Limitations of Existing Tools for Network Analysis



□ **Inefficient at performing network analysis on large-scale networks**

- **Example: betweenness centrality for 100 thousand nodes**

□ **Overlook some critical functions**

- **Structural hole theory**
- **Motif detection**
- **Network embedding**

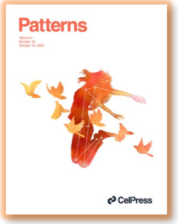
□ **Support only a limited number of network formats**

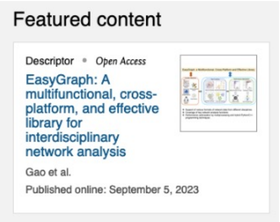
- **NetworkX for 5 types**
- **SNAP for 2 types**

EasyGraph

Python 3.8 | 3.9 | 3.10 | 3.11 | 3.12 | 3.13 ☆ 443 stars Downloads 892k

- **An open-source network analysis library**
- **Code website:** *https://github.com/easy-graph/Easy-Graph*
- **Documentation website:** *https://easy-graph.github.io/docs/index.html*
- **Video tutorial website:** *https://easy-graph.github.io/docs/videos.html*



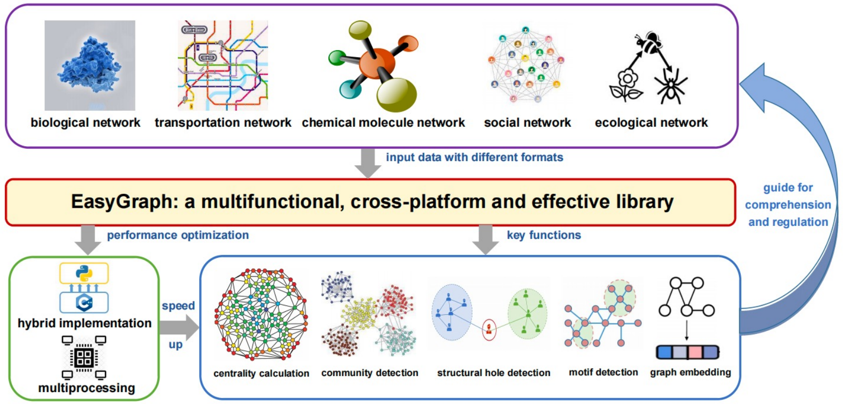

Featured content

Descriptor • *Open Access*

EasyGraph: A multifunctional, cross-platform, and effective library for interdisciplinary network analysis

Gao et al.

Published online: September 5, 2023



Framework of EasyGraph

Patterns

CellPress OPEN ACCESS

Descriptor

EasyGraph: A Multifunctional, Cross-Platform, and Effective Library for Interdisciplinary Network Analysis

Min Gao,[1] Zheng Li,[1] Ruichen Li,[1] Chenhao Cui,[1] Xinyuan Chen,[1] Bodian Ye,[1] Yupeng Li,[2] Weiwei Gu,[3] Qingyuan Gong,[1] Xin Wang,[1] and Yang Chen[1,4,*]

[1]Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai, China

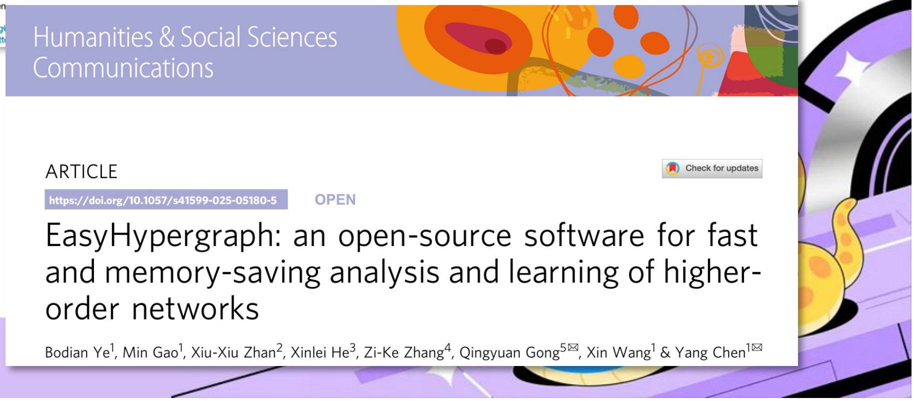
[2]Department of Interactive Media, Hong Kong Baptist University, Hong Kong, China

[3]College of Information Scien

[4]Lead contact

*Correspondence: chenyang

https://doi.org/10.1016/j.patt


Humanities & Social Sciences Communications

ARTICLE

Check for updates

https://doi.org/10.1057/s41599-025-05180-5 OPEN

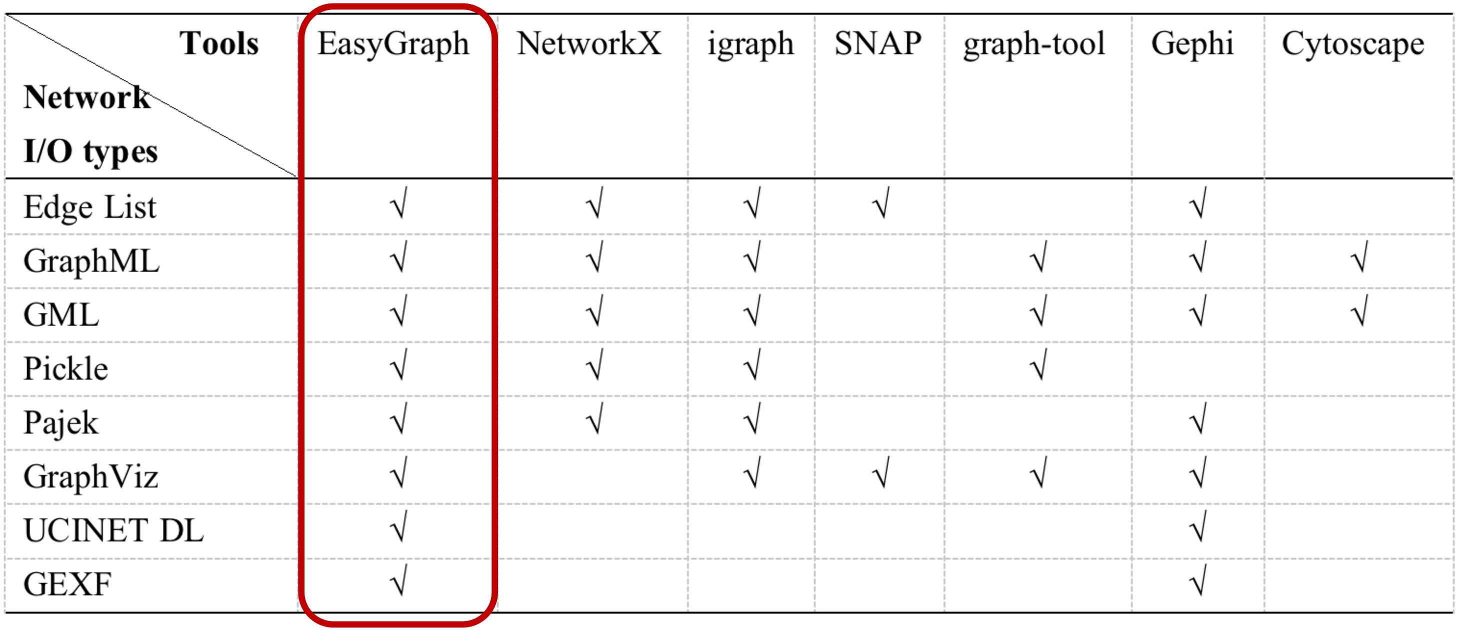
EasyHypergraph: an open-source software for fast and memory-saving analysis and learning of higher-order networks

Bodian Ye[1], Min Gao[1], Xiu-Xiu Zhan[2], Xinlei He[3], Zi-Ke Zhang[4], Qingyuan Gong[5✉], Xin Wang[1] & Yang Chen[1✉]

Features of EasyGraph

❖Multiple Types of Network I/O

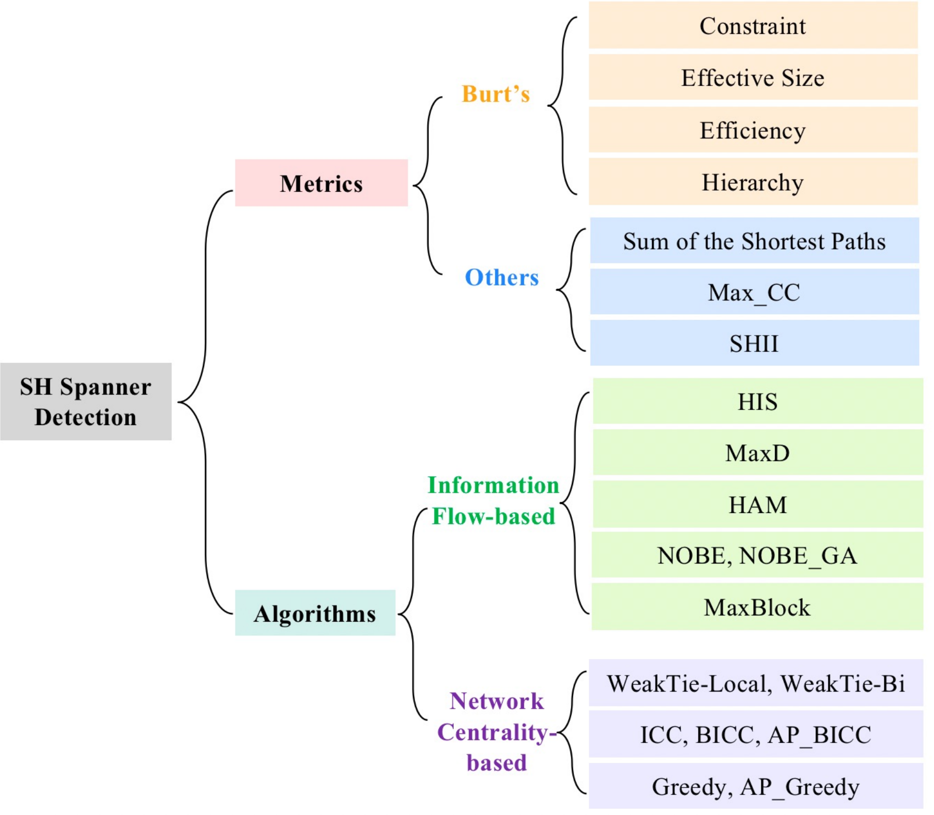
Comparison of network analysis tools in terms of supporting different network I/O types



Tools **Network I/O types**	EasyGraph	NetworkX	igraph	SNAP	graph-tool	Gephi	Cytoscape
Edge List	√	√	√	√		√	
GraphML	√	√	√		√	√	√
GML	√	√	√		√	√	√
Pickle	√	√	√		√		
Pajek	√	√	√			√	
GraphViz	√		√	√	√	√	
UCINET DL	√					√	
GEXF	√					√	

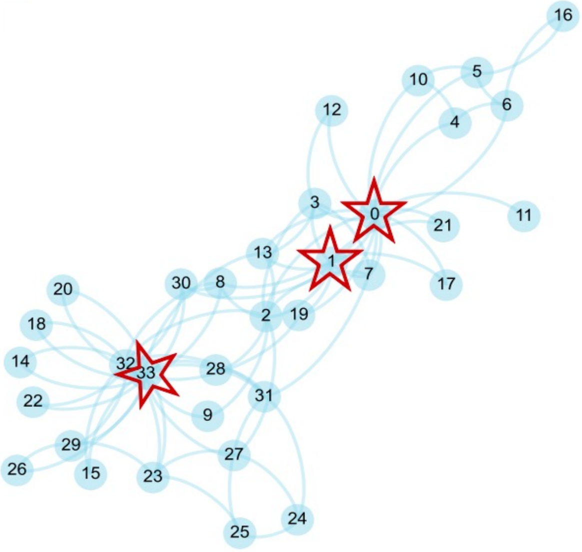
Features of EasyGraph (cont.)

❖Algorithms of SH Spanner Detection



SH Spanner Detection

- Metrics
  - Burt's
    - Constraint
    - Effective Size
    - Efficiency
    - Hierarchy
  - Others
    - Sum of the Shortest Paths
    - Max_CC
    - SHII
- Algorithms
  - Information Flow-based
    - HIS
    - MaxD
    - HAM
    - NOBE, NOBE_GA
    - MaxBlock
  - Network Centrality-based
    - WeakTie-Local, WeakTie-Bi
    - ICC, BICC, AP_BICC
    - Greedy, AP_Greedy

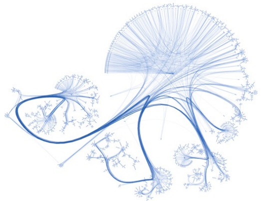


SH spanner detection by AP_Greedy

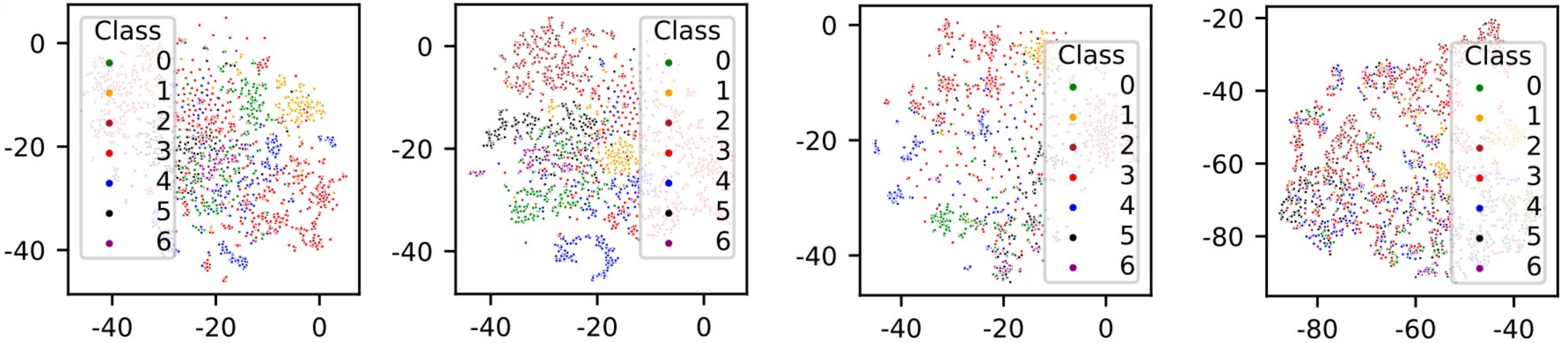
Features of EasyGraph (cont.)

❖ **Methods for Network Embedding:** DeepWalk, node2vec, LINE, and SDNE

- ❑ **The Cora dataset:** 2,708 scientific publications and 5,429 citation links
- ❑ **Each publication:** 1,433 dimensional features, 1 of 7 categories



Cora network



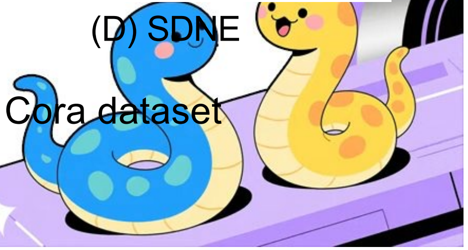
(A) DeepWalk

(B) node2vec

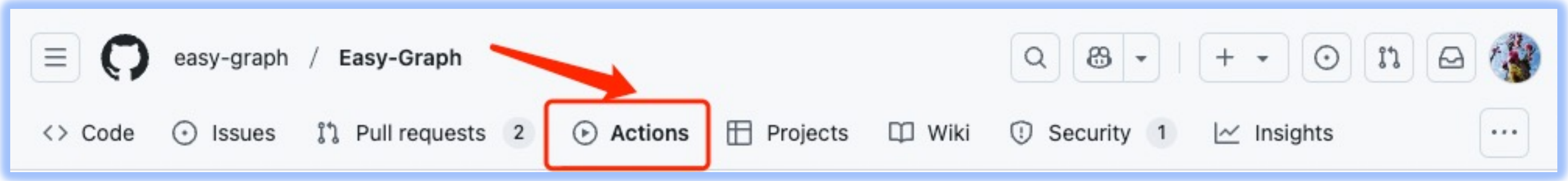
(C) LINE

(D) SDNE

Visualization of the results of four network embedding algorithms using t-SNE for Cora dataset



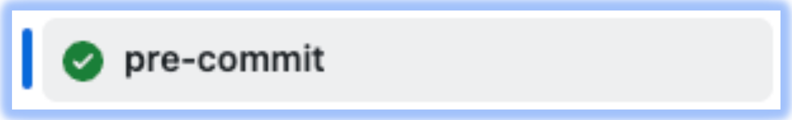
Automated Testing and Releasing



GitHub Actions

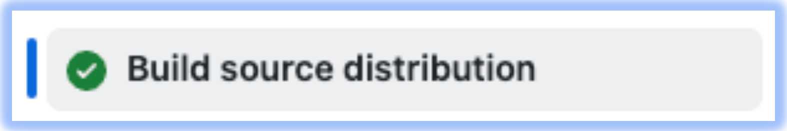
A continuous integration and continuous delivery (CI/CD) platform

- EasyGraph leverages smart pre-commit hooks for code formatting, linting and testing

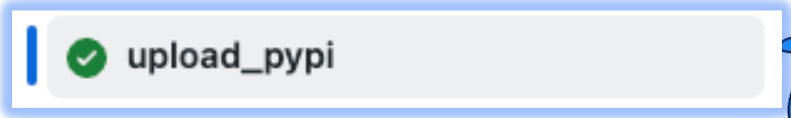


✓ pre-commit

- EasyGraph defines workflows to build and upload wheels



✓ Build source distribution



✓ upload_pypi

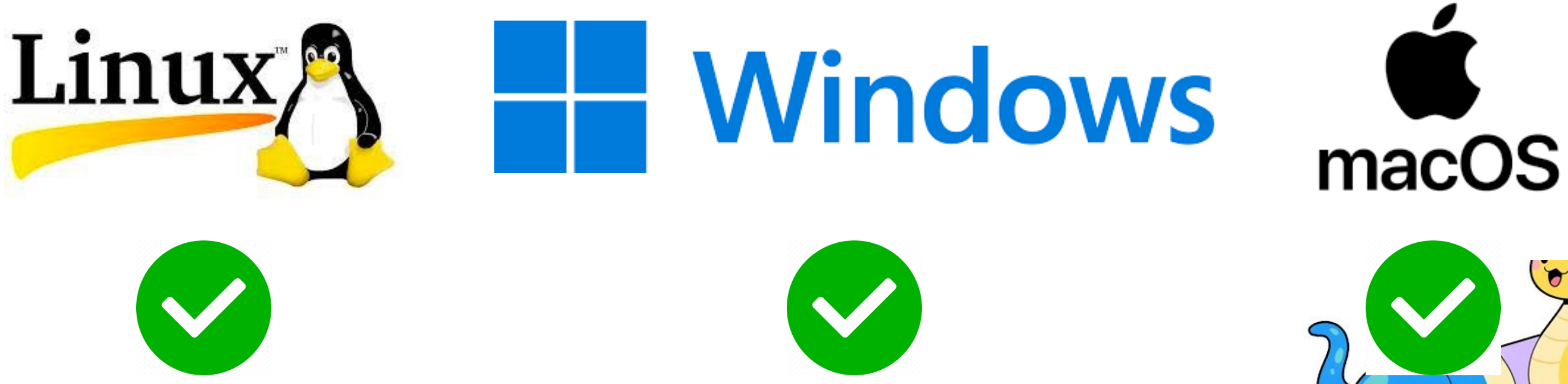
● ● ●

Installation

- ❖ **Easy Installation**

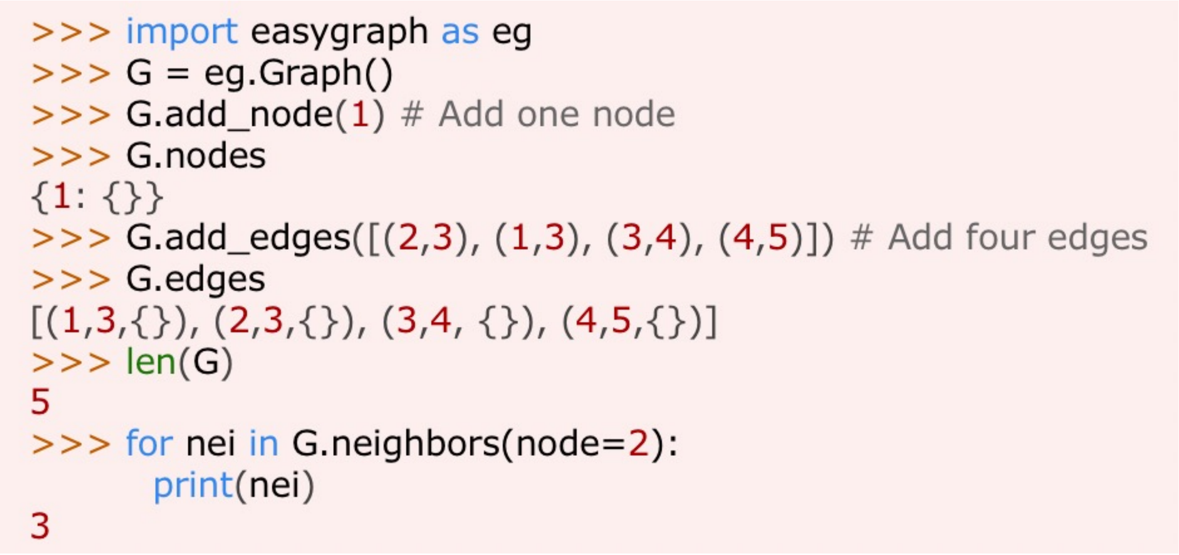
```
pip install Python-EasyGraph
```

- ❖ **Cross-Platform Compatibility**

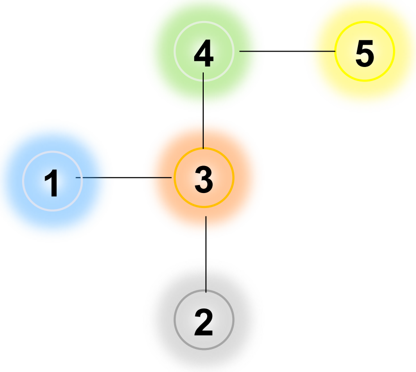


Ease-of-Use APIs and Use Cases

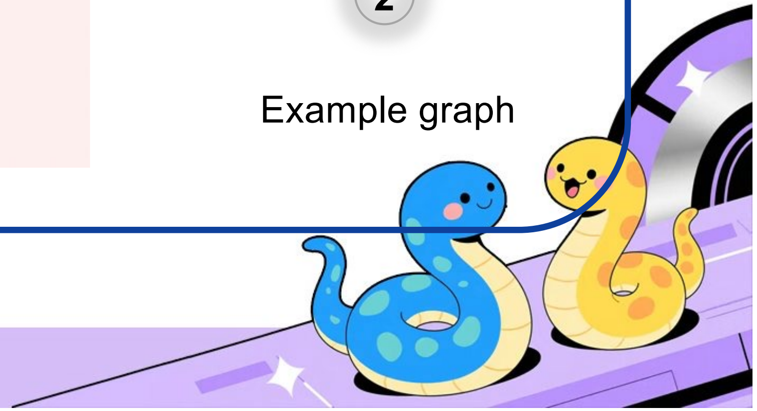
❖ **Definition & Operations:** add_nodes, add_edges, edges, len, neighbors



```
>>> import easygraph as eg
>>> G = eg.Graph()
>>> G.add_node(1) # Add one node
>>> G.nodes
{1: {}}
>>> G.add_edges([(2,3), (1,3), (3,4), (4,5)]) # Add four edges
>>> G.edges
[(1,3,{}), (2,3,{}), (3,4, {}), (4,5,{})]
>>> len(G)
5
>>> for nei in G.neighbors(node=2):
       print(nei)
3
```

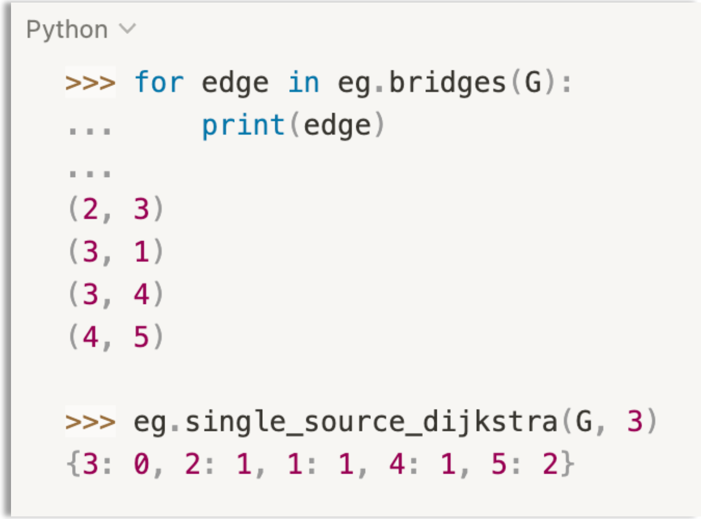


Example graph



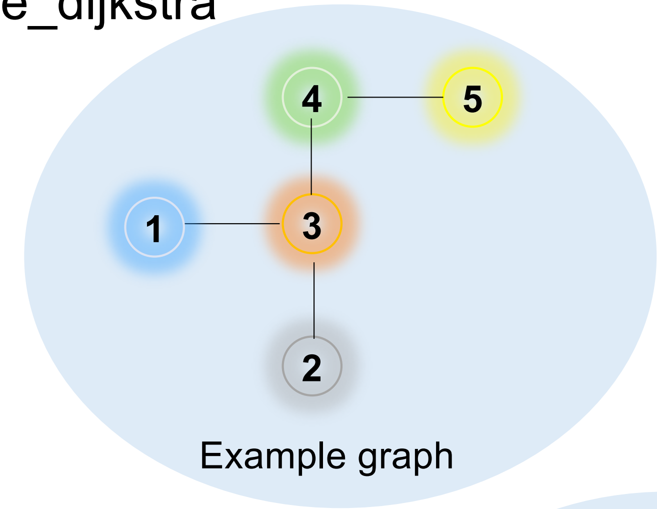
Ease-of-Use APIs and Use Cases

- **Functions:** bridges, single_source_dijkstra



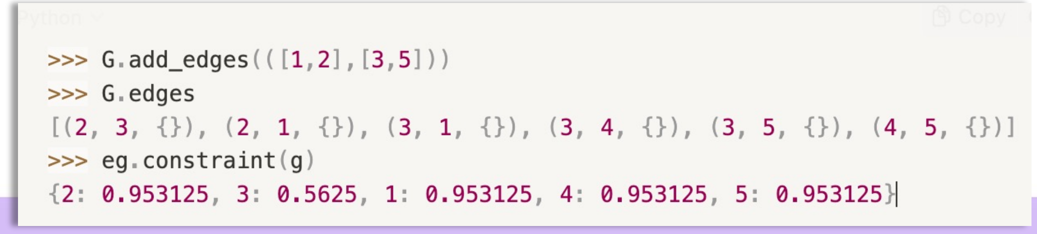
```
>>> for edge in eg.bridges(G):
...     print(edge)
...
(2, 3)
(3, 1)
(3, 4)
(4, 5)

>>> eg.single_source_dijkstra(G, 3)
{3: 0, 2: 1, 1: 1, 4: 1, 5: 2}
```

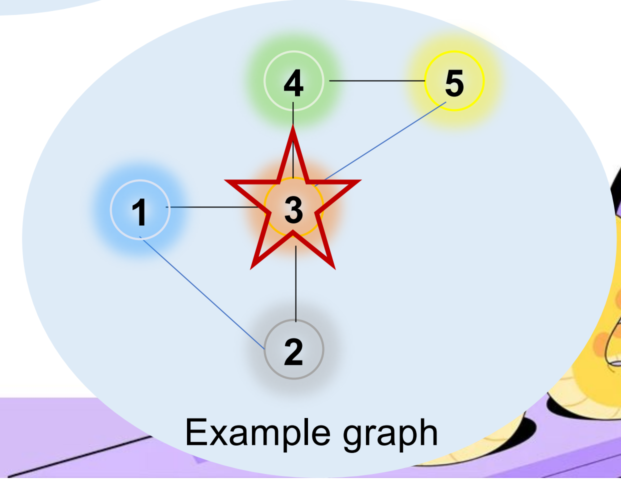


Example graph

- **Functions:** constraint (a SH spanner metric)



```
>>> G.add_edges(([1,2],[3,5]))
>>> G.edges
[(2, 3, {}), (2, 1, {}), (3, 1, {}), (3, 4, {}), (3, 5, {}), (4, 5, {})]
>>> eg.constraint(g)
{2: 0.953125, 3: 0.5625, 1: 0.953125, 4: 0.953125, 5: 0.953125}
```



Example graph

Real-world Applications of the Structural Hole Theory

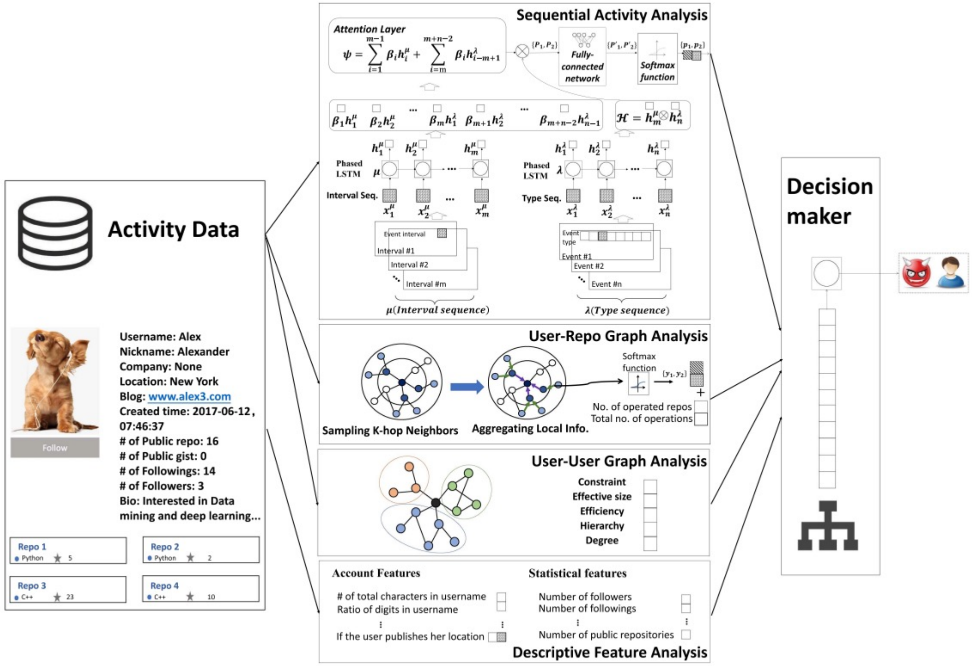
❖ Applications

Influence maximization

Innovation

Information propagation

❖ Case: Malicious User Detection on GitHub



Username: Alex
Nickname: Alexander
Company: None
Location: New York
Blog: www.alex3.com
Created time: 2017-06-12, 07:46:37
# of Public repo: 16
# of Public gist: 0
# of Followings: 14
# of Followers: 3
Bio: Interested in Data mining and deep learning...

Repo 1 Python ★ 5	Repo 2 Python ★ 2
Repo 3 C++ ★ 23	Repo 4 C++ ★ 10

# of total characters in username
Ratio of digits in username
⋮
If the user publishes her location

Number of followers
Number of followings
⋮
Number of public repositories

Phased LSTM + Attention Mechanism + SH Spanner Metrics

IEEE Transactions on Knowledge and Data Engineering 2023

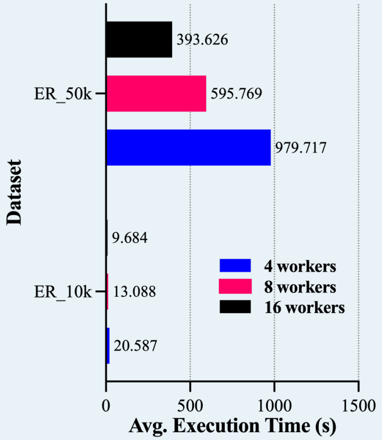
Performance of EasyGraph

❖ Enhancement by Multiprocessing

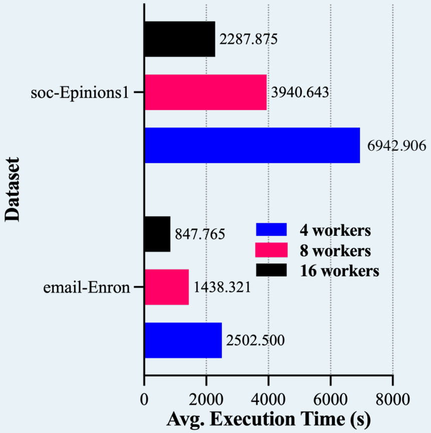
Network	#Nodes	#Edges	Avg. Degree	is directed
ER_10k	10,000	20,000	4.0	False
ER_50k	50,000	100,000	4.0	False
Email-Enron	36,692	183,831	10.0	True
Soc-Epinions1	75,879	508,837	13.4	True

Experiment Settings

- Ubuntu 22.04 LTS
- Python: 3.11
- Python-EasyGraph:1.0
- Function: betweenness centrality
- #Cores: 4,8,16

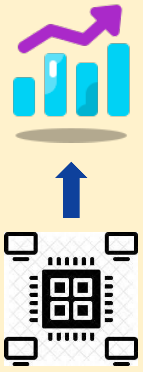


betweenness centrality



betweenness centrality

performance



multiprocessing

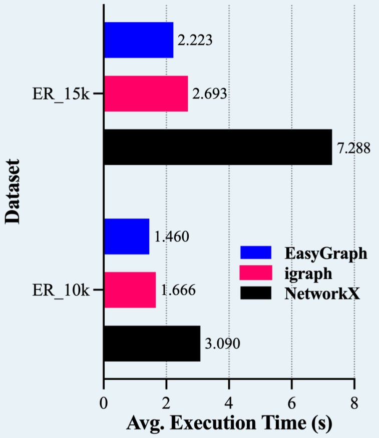
Performance of EasyGraph

❖ Comparison among EasyGraph, igraph and

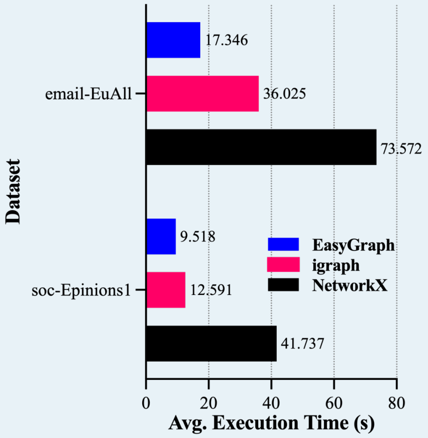
Network	#Nodes	#Edges	Avg. Degree	is directed
ER_10k	10,000	20,000	4.0	True
ER_15k	15,000	30,000	4.1	True
Email-Enron	36,692	183,831	10.0	True
Soc-Epinions1	75,879	508,837	13.4	True

Experiment Settings

- Ubuntu 22.04 LTS
- Python: 3.11
- Python-EasyGraph:1.0
- igraph: 0.10.6
- NetworkX: 3.1
- Function: Finding paths

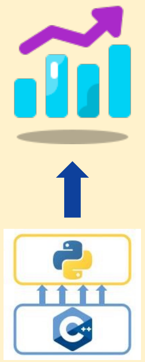


finding the shortest paths

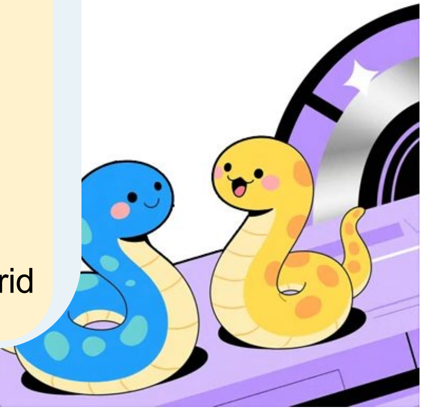


finding the shortest paths

performance



Python/C++ hybrid programming



Optimization Techniques

❖ **Dirty flag design pattern (Xu et al., 2019)**

We use the dirty flag to **track the state** of variables during network analysis

❖ **The singly-linked list data structure (Berdine et al., 2007)**

We apply this data structure to **store information of adjacent nodes** in a network

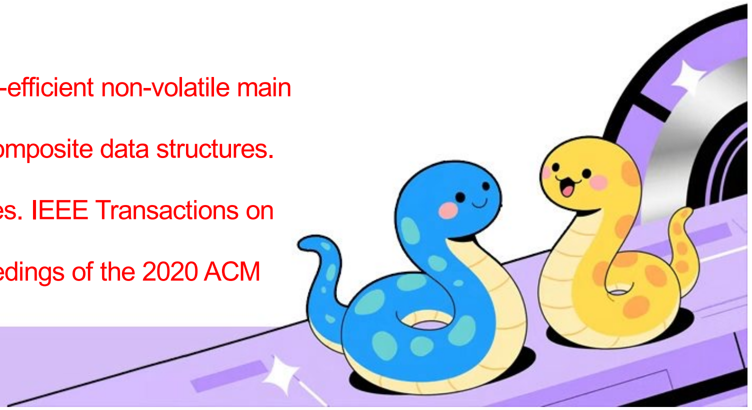
❖ **The segment tree data structure (Xu et al., 2020)**

We introduce a segment tree **in Dijkstra's algorithm** to implement the identification of the **shortest paths**

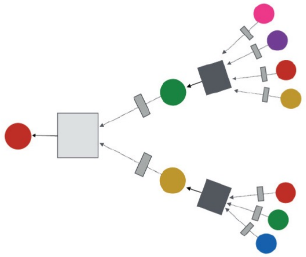
❖ **The Radix sorting algorithm (Kristo et al., 2020)**

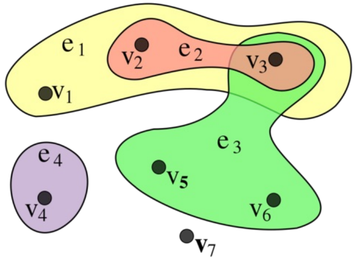
We choose this algorithm to **optimize the k-core function**

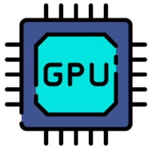
- Xu, J., Feng, D., Hua, Y., Tong, W., Liu, J., Li, C., Xu G., & Chen, Y. Adaptive granularity encoding for energy-efficient non-volatile main memory. Proceedings of the 56th Annual Design Automation Conference, 2019, 1-6.
- Berdine, J., Calcagno, C., Cook, B., Distefano, D., O'hearn, P. W., Wies, T., & Yang, H. Shape analysis for composite data structures. Proceedings of the 19th International Conference on Computer Aided Verification, 2007, 178-192.
- Xu, Y., Tong, Y., Shi, Y., Tao, Q., Xu, K., & Li, W. An efficient insertion operator in dynamic ridesharing services. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(8):3583-3596.
- Kristo, A., Vaidya, K., Çetintemel, U., Misra, S., & Kraska, T. The case for a learned sorting algorithm. Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, 2020, 1001-1016.



Key Elements of EasyGraph 2.0 Overview





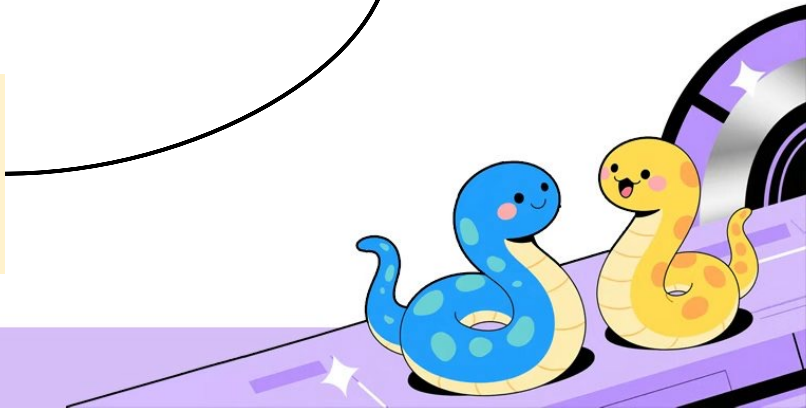


EasyGNN

EasyHypergraph

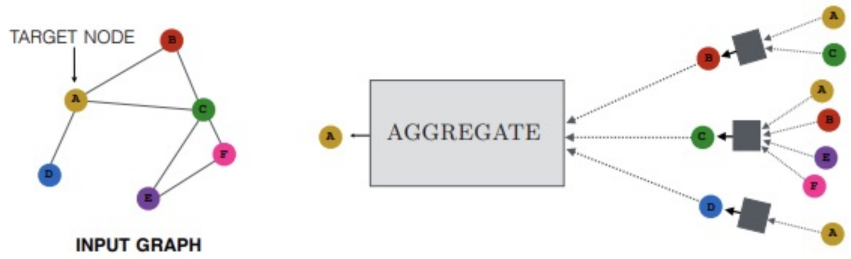
EGGPU

EasyGraph (Core)



Ongoing Work #1: EasyGNN

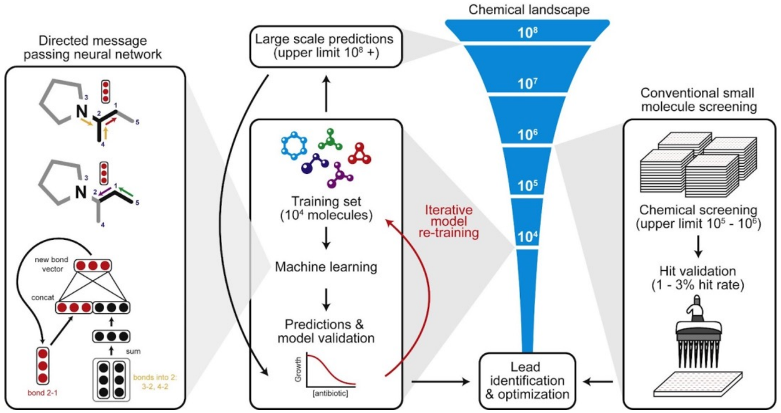
❖ GNN Models with Message Passing



❖ Representative GNN Models

- GCN
- GAT
- GraphSAGE
- GRAND
- Graph Transformer
- ……

❖ GNN Applications


Directed message passing neural network
N
new bond vector
concat
sum
bond 2-1
bonds into 2: 3-2, 4-2
Large scale predictions (upper limit 10^8 +)
Training set (10^4 molecules)
Machine learning
Predictions & model validation
Growth
[antibiotic]
Iterative model re-training
Chemical landscape
10^8
10^7
10^6
10^5
10^4
Conventional small molecule screening
Chemical screening (upper limit 10^5 - 10^6)
Hit validation (1 - 3% hit rate)
Lead identification & optimization

Drug discovery and synthesize chemical compounds

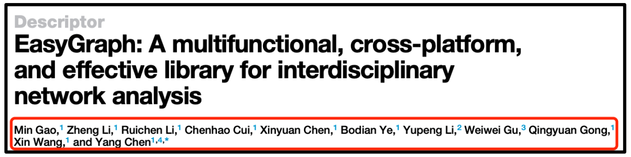
Social influence prediction

Recommendation systems

Ongoing Work #2: Group Interaction & Hypergraph Analysis

❖ Group Interactions are Everywhere

Collaborations of researchers

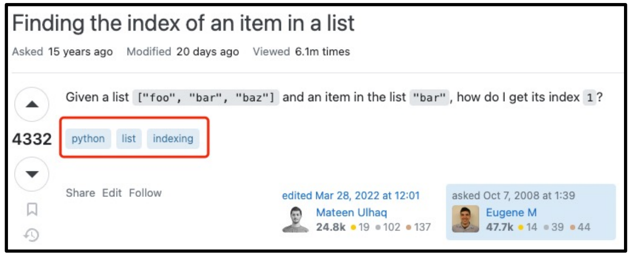


Descriptor

EasyGraph: A multifunctional, cross-platform, and effective library for interdisciplinary network analysis

Min Gao,[1] Zheng Li,[1] Ruichen Li,[1] Chenhao Cui,[1] Xinyuan Chen,[1] Bodian Ye,[1] Yupeng Li,[2] Weiwei Gu,[3] Qingyuan Gong,[1] Xin Wang,[1] and Yang Chen[1,4,*]

Tags in online Q&A sites



Finding the index of an item in a list

Asked 15 years ago Modified 20 days ago Viewed 6.1m times

Given a list `["foo", "bar", "baz"]` and an item in the list `"bar"`, how do I get its index `1`?

4332

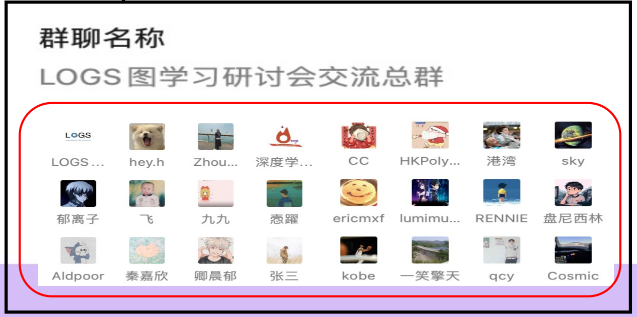
python list indexing

Share Edit Follow

edited Mar 28, 2022 at 12:01
Mateen Ulhaq
24.8k ● 19 ● 102 ● 137

asked Oct 7, 2008 at 1:39
Eugene M
47.7k ● 14 ● 39 ● 44

Group chats in WeChat

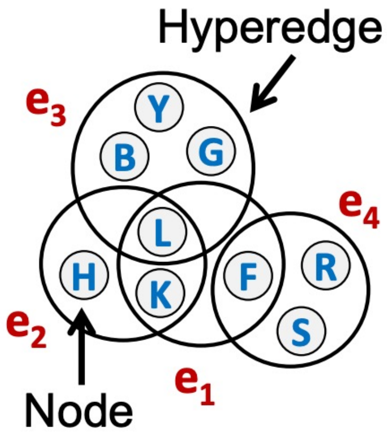


群聊名称

LOGS图学习研讨会交流总群

LOGS ...	hey.h	Zhou...	深度学...	CC	HKPoly...	港湾	sky
郁离子	飞	九九	悫躍	ericmxf	lumimu...	RENNIE	盘尼西林
Aldpoor	秦嘉欣	卿晨郁	张三	kobe	一笑擎天	qcy	Cosmic

❖ Hypergraphs Model Group Interactions



Hypergraph analysis

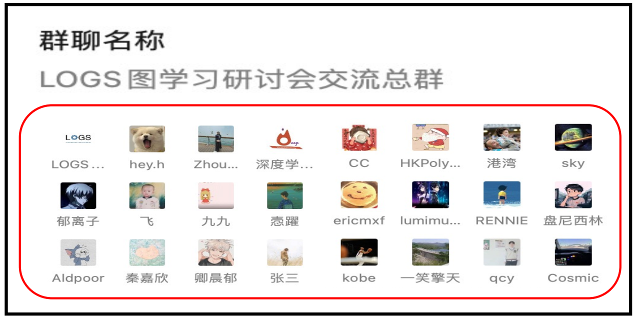
- ➢ structural metrics: degree, distance, etc
- ➢ hypergraph generators: random, uniform, etc
- ➢ hypergraph learning: HGNN [AAAI'19]
- ➢ ......

- Feng, Y., You, H., Zhang, Z., Ji, R., & Gao, Y. Hypergraph neural networks. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33(1): 3558-3565.
- Zhang, Y., Lucas, M., & Battiston, F. Higher-order interactions shape collective dynamics differently in hypergraphs and simplicial complexes. Nature Communications, 2023, 14(1): 1605.

Ongoing Work #2: Group Interaction & Hypergraph Analysis

- **A Higher-order network** describes the interaction involving multiple entities.
- **Hypergraph** is one of the most typical representations of higher-order relationships.

Group chats on WeChat



群聊名称

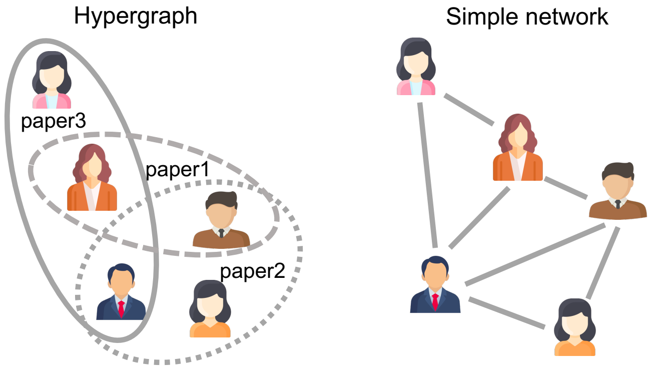
LOGS图学习研讨会交流总群

LOGS ... hey.h Zhou... 深度学... CC HKPoly... 港湾 sky

郁离子 飞 九九 悫躍 ericmxf lumimu... RENNIE 盘尼西林

Aldpoor 秦嘉欣 卿晨郁 张三 kobe 一笑擎天 qcy Cosmic

WeChat groups



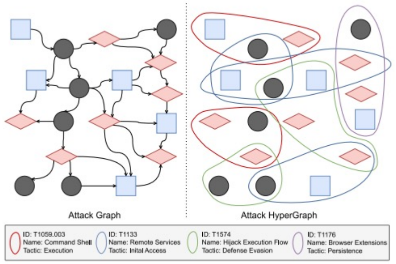
Co-authorship network

Ongoing Work #2: Group Interaction & Hypergraph Analysis

➢ Hypergraph analysis

The quantization and characterization of hypergraph structures at multiple scales

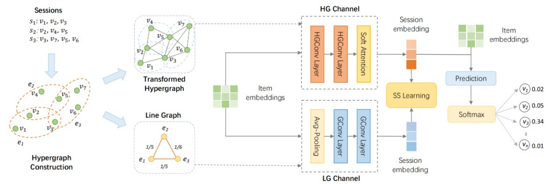
- ➢ Social science[1], Biochemistry [2] , Cybersecurity[3]

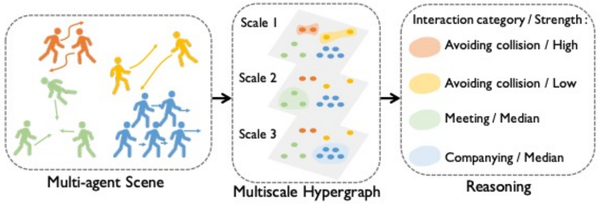


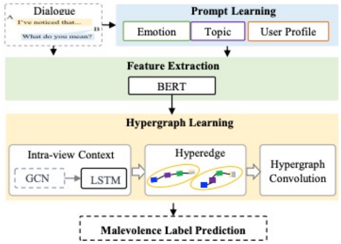
➢ Hypergraph learning

The node representation learning through HNNs

- ➢ Recommendation system[4] , Trajectory prediction[5], Natural Language Processing[6]







[1] Cheng K, Cheng X, Wang W. The determinants influencing bilingual instruction in Chinese higher education: a complex network analysis[J]. Humanities and Socıal Sciences Communications, 2024, 11(1): 1-12.

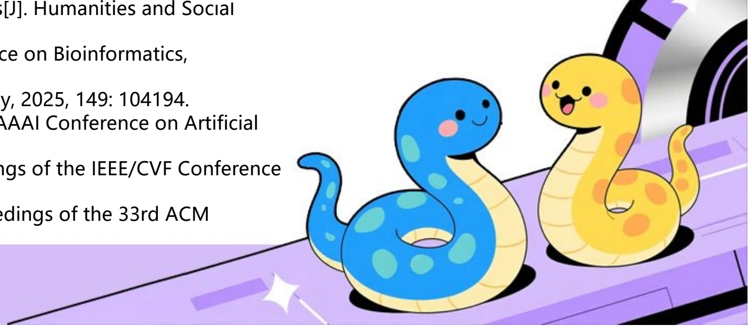
[2] Gallagher S R, Goldberg D S. Clustering coefficients in protein interaction hypernetworks[C]//Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics. 2013: 552-560.

[3] Jia J, Yang L, Wang Y, et al. Hyper attack graph: Constructing a hypergraph for cyber threat intelligence analysis[J]. Computers & Security, 2025, 149: 104194.

[4] Xia X, Yin H, Yu J, et al. Self-supervised hypergraph convolutional networks for session-based recommendation[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(5): 4503-4511.

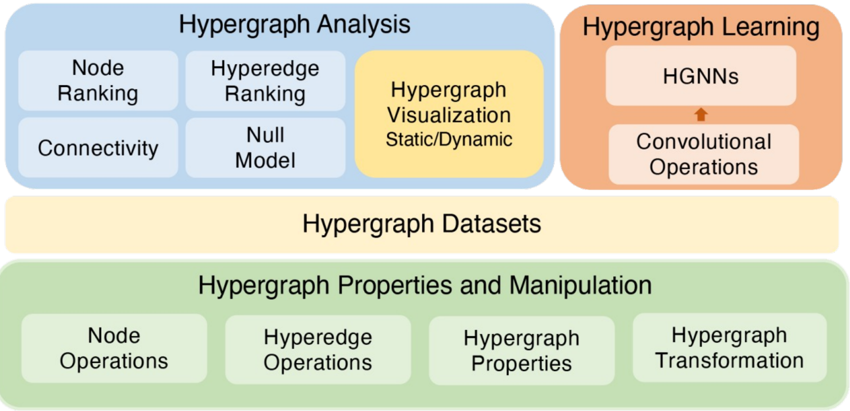
[5] Xu C, Li M, Ni Z, et al. Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 6498-6507.

[6] Xu B, Qiao X, Lin H, et al. MPHDetect: Multi-View Prompting and Hypergraph Fusion for Malevolence Detection in Dialogues[C]//Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. 2024: 4133-4137.

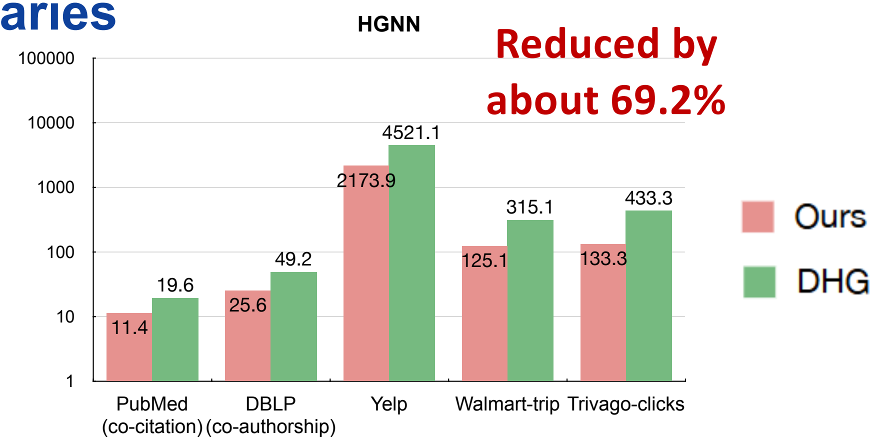


Ongoing Work #2: Group Interaction & Hypergraph Analysis

❖ Hypergraph function framework



❖ Running time of EasyGraph and other libraries

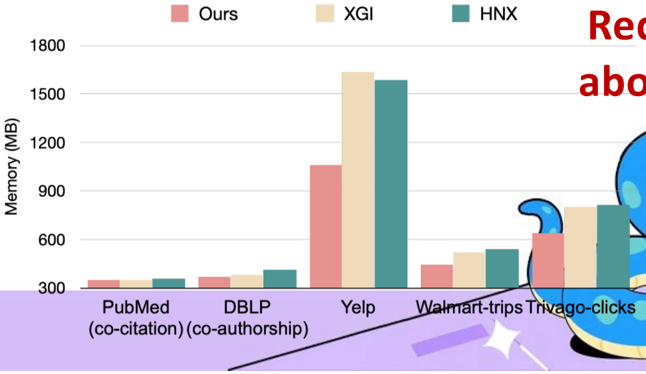


Reduced by about 69.2%

❖ Hypergraph function comparison

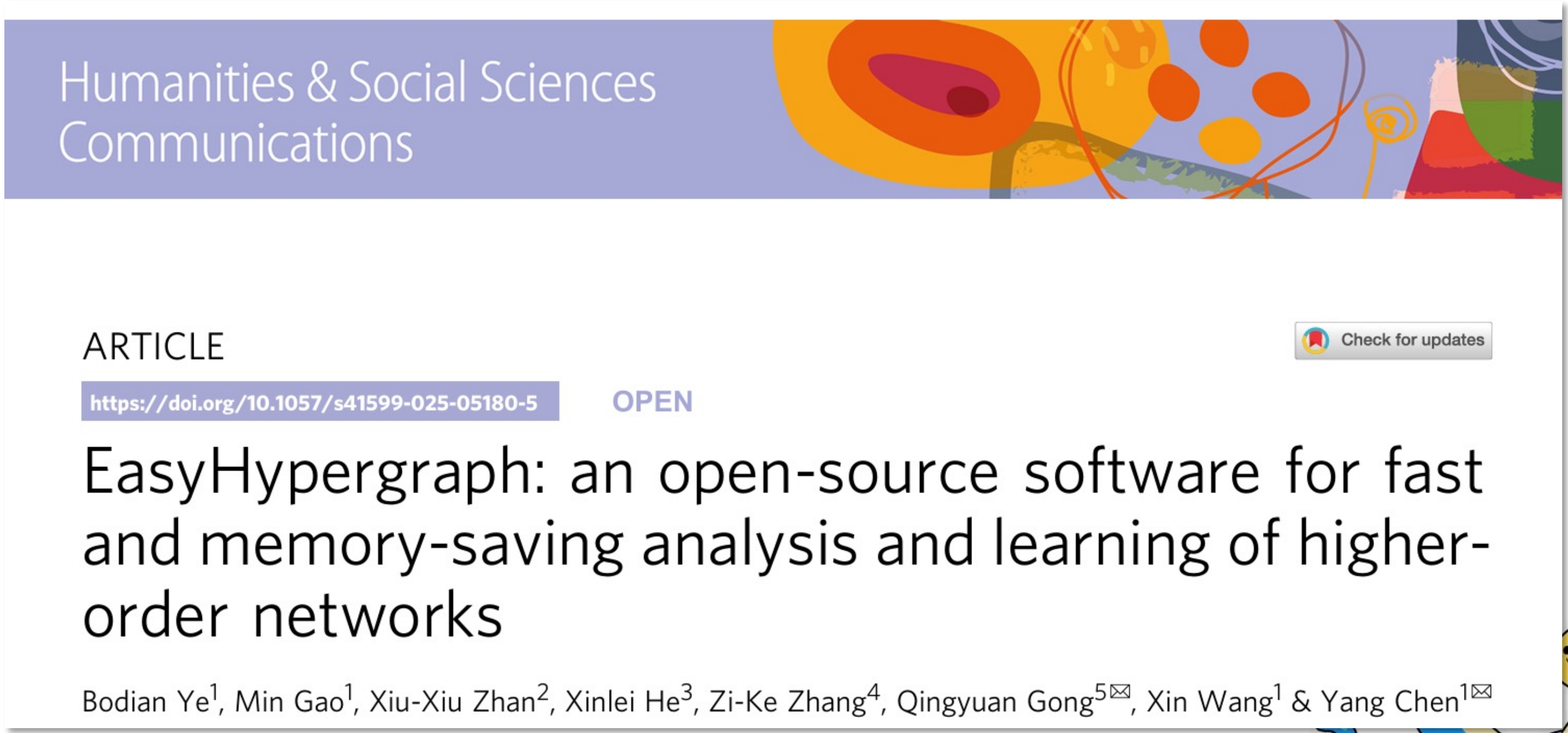
Library	Hypergraph Analysis	Hypergraph Learning	Structure Visualization	Continuous Development
XGI	✓	-	✓	✓
HNX	✓	-	✓	✓
HGX	✓	-	✓	✓
Reticula	✓	-	-	✓
HyperX	✓	-	-	-
Halp	✓	-	-	-
DHG	-	✓	✓	✓
Ours	✓	✓	✓	✓

❖ Memory cost of EasyGraph and other libraries



Reduced by about 35.3%

Ongoing Work #2: Group Interaction & Hypergraph Analysis



Humanities & Social Sciences Communications

ARTICLE

Check for updates

https://doi.org/10.1057/s41599-025-05180-5

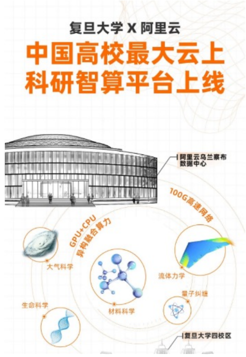
OPEN

EasyHypergraph: an open-source software for fast and memory-saving analysis and learning of higher-order networks

Bodian Ye[1], Min Gao[1], Xiu-Xiu Zhan[2], Xinlei He[3], Zi-Ke Zhang[4], Qingyuan Gong[5]✉, Xin Wang[1] & Yang Chen[1]✉

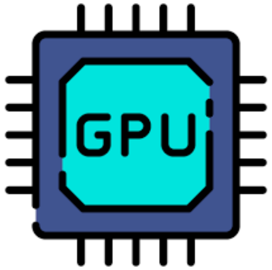
Ongoing Work #3: GPU-Enhanced Network Analysis

❖ GPU-based Performance Optimization



复旦大学 X 阿里云

中国高校最大云上
科研智算平台上线



Selected Network Analysis Metrics

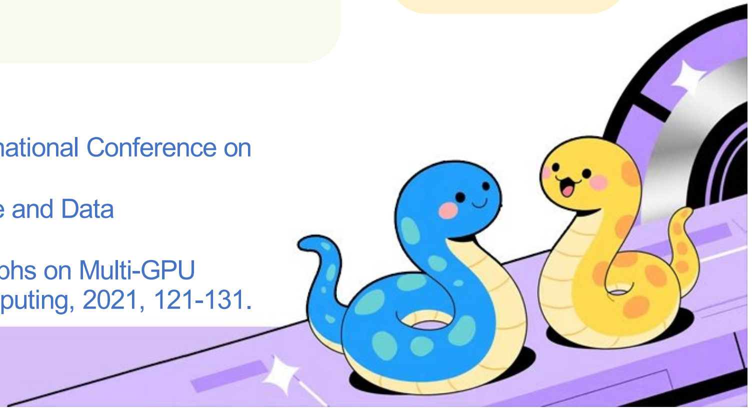
- closeness centrality
- betweenness centrality
- k-core centrality

Selected Network Analysis Algorithms

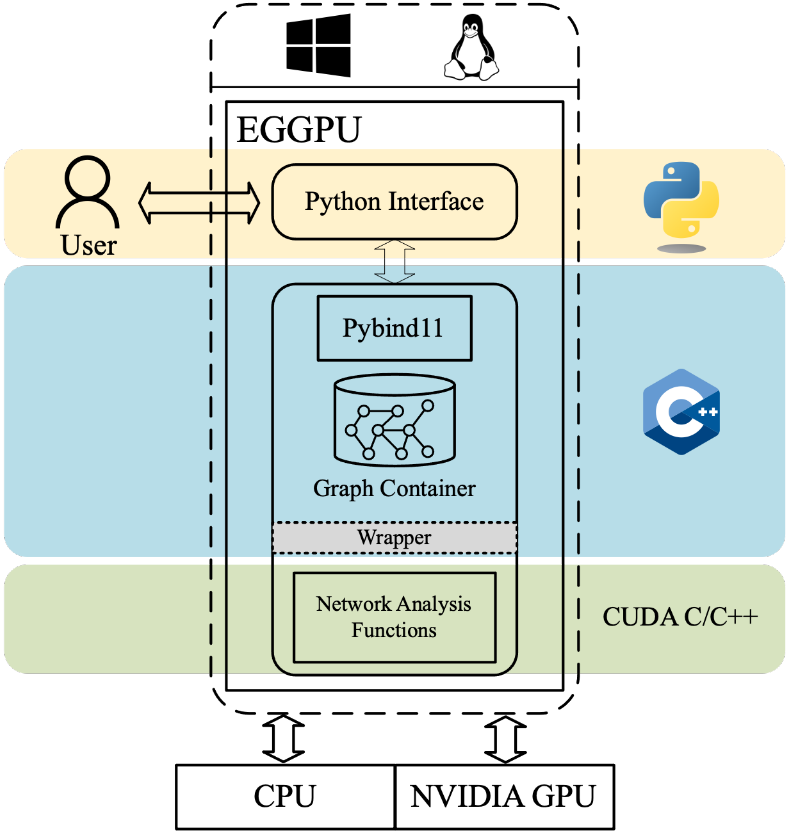
- graph traversal
- graph sampling

performance

- Sha, M., Li, Y., & Tan, K. L. GPU-based graph traversal on compressed graphs. Proceedings of the 2019 International Conference on Management of Data, 2019, 775-792.
- Guo, W., Li, Y., & Tan, K. L. Exploiting reuse for GPU subgraph enumeration. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(9), 4231-4244.
- Sao, P., Lu, H., Kannan, R., Thakkar, V., Vuduc, R., & Potok, T. Scalable All-pairs Shortest Paths for Huge Graphs on Multi-GPU Clusters. Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing, 2021, 121-131.



Ongoing Work #3: GPU-Enhanced Network Analysis



EGGPU Framework

User Interface Layer

- Developed in Python
- Provides Python API
- Relays commands to the following layer

Middleware Layer

- Built with C++
- Intermediary between the two layers
- Includes a graph container responsible for the loading, storage, and transformation of graphs

Computation Layer

- Powered by CUDA C/C++
- Supports various network analysis functions, including betweenness centrality, k-core centrality, and SSSP

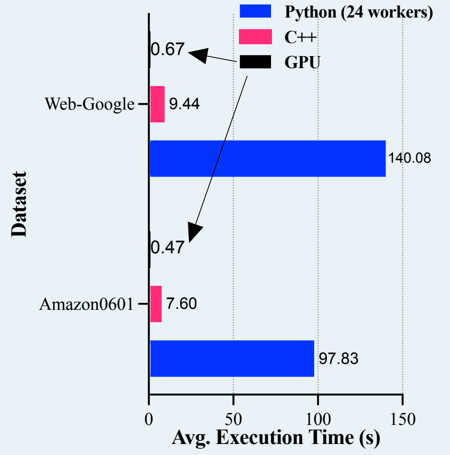
Ongoing Work #3: GPU-Enhanced Network Analysis

❖ GPU-based Performance Optimization

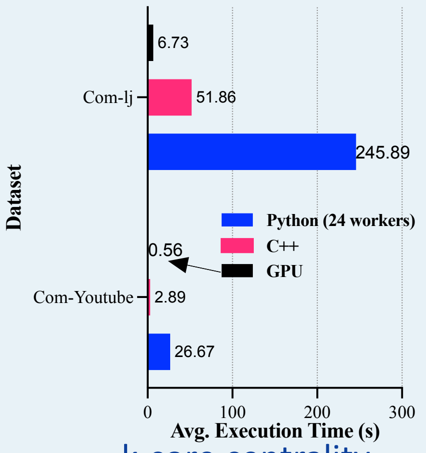
Network	#Nodes	#Edges	Avg. Degree	Is directed
Amazon0601	403,394	2,443,408	9.87	True
web-Google	875,713	4,322,051	12.11	True
com-Youtube	1,134,890	2,987,624	5.27	False
com-LiveJournal	3,997,962	34,681,189	17.35	False

Experiment Settings

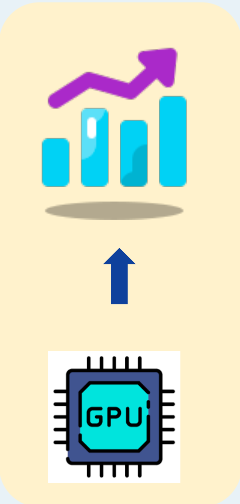
- Ubuntu 23.10 LTS
- Python: 3.10
- Python-EasyGraph:1.2
- Function: betweenness and k-core centrality
- GPU: GeForce RTX 4090



betweenness centrality



k-core centrality

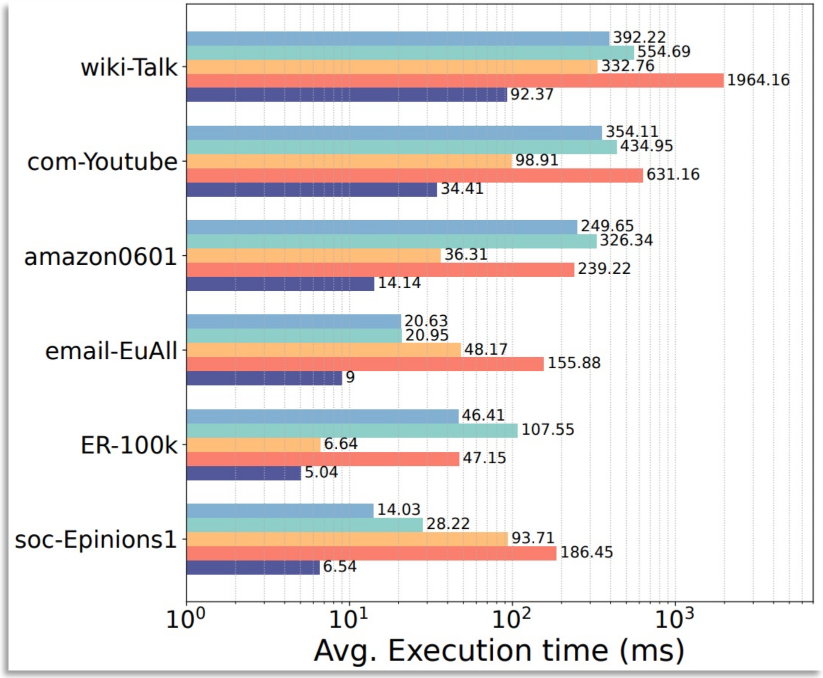


Reduced by about 99.5%

Ongoing Work #3: GPU-Enhanced Network Analysis

cuGraph

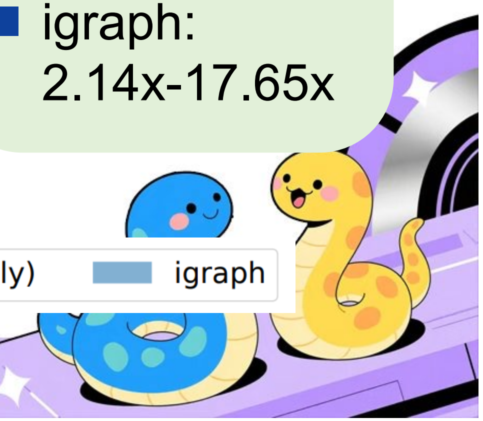
Gunrock



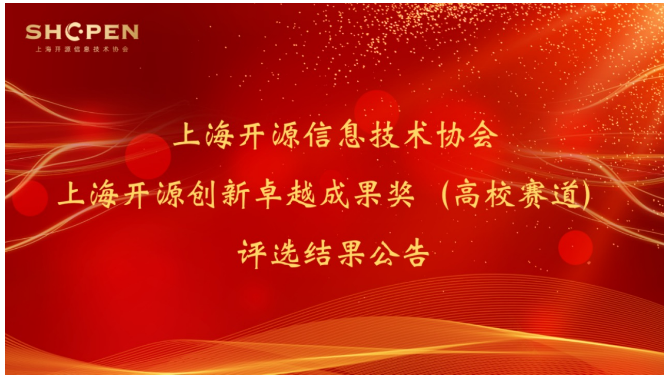
K-core centrality

Performance Comparison

- nx_cugraph: 9.36x-28.49x
- Gunrock: 1.31x-14.32x
- EG(CPU only): 2.23x-23.07x
- igraph: 2.14x-17.65x

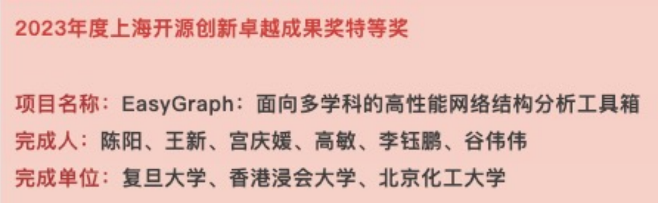


Awards and Media Coverage



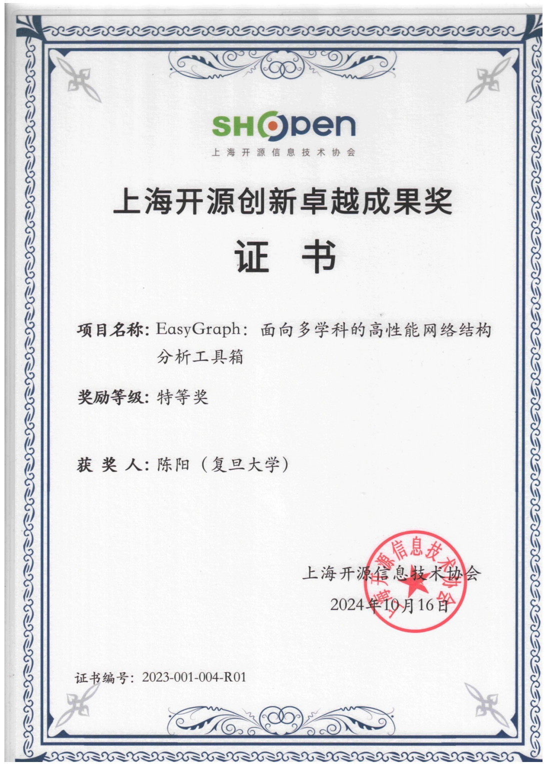
SHOPEN
上海开源信息技术协会

上海开源信息技术协会
上海开源创新卓越成果奖（高校赛道）
评选结果公告



2023年度上海开源创新卓越成果奖特等奖

项目名称：EasyGraph：面向多学科的高性能网络结构分析工具箱
完成人：陈阳、王新、宫庆媛、高敏、李钰鹏、谷伟伟
完成单位：复旦大学、香港浸会大学、北京化工大学



SHOpen
上海开源信息技术协会

上海开源创新卓越成果奖
证书

项目名称：EasyGraph：面向多学科的高性能网络结构分析工具箱

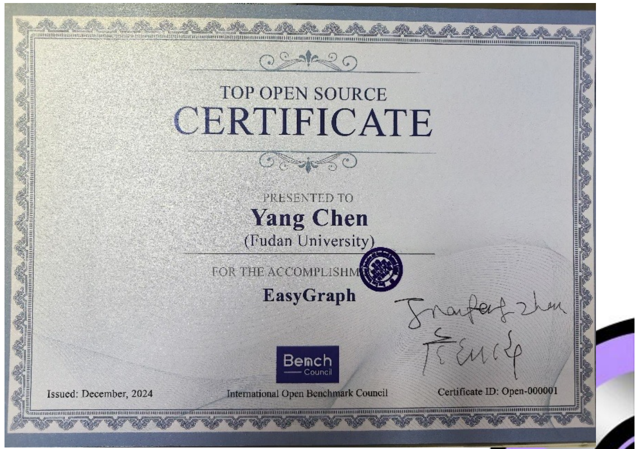
奖励等级：特等奖

获奖人：陈阳（复旦大学）

上海开源信息技术协会
2024年10月16日

上海开源信息技术协会

证书编号：2023-001-004-R01



TOP OPEN SOURCE
CERTIFICATE

PRESENTED TO
Yang Chen
(Fudan University)

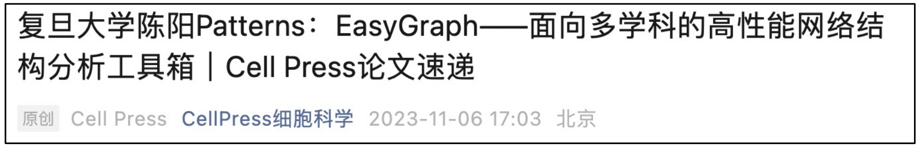
FOR THE ACCOMPLISHM
EasyGraph

Bench Council

Issued: December, 2024
International Open Benchmark Council
Certificate ID: Open-000001

Shanghai Open Source Innovation
Outstanding Achievement Award (Grand Prize)

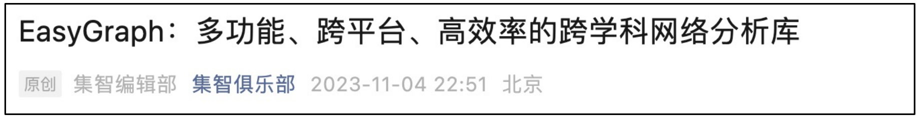
Awards and Media Coverage (cont.)



复旦大学陈阳Patterns：EasyGraph——面向多学科的高性能网络结构分析工具箱｜Cell Press论文速递

原创 Cell Press CellPress细胞科学 2023-11-06 17:03 北京

WeChat Official Account: Cell Press



EasyGraph：多功能、跨平台、高效率的跨学科网络分析库

原创 集智编辑部 集智俱乐部 2023-11-04 22:51 北京

WeChat Official Account: Swarma Club

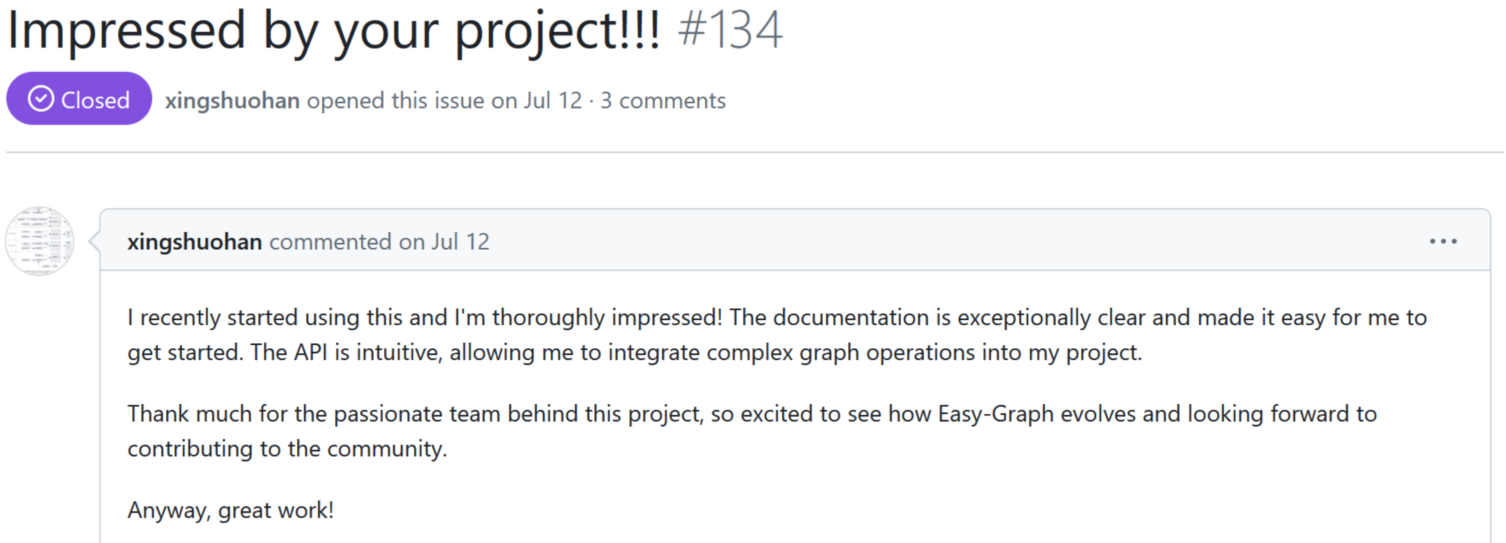
陈阳
segmentfault
开源社
kaiyuanshe

2024
中国开源先锋 33 人
心尖上的开源人物

陈 阳

复旦大学计算机科学技术学院教授，开源软件 EasyGraph 创始人 / 负责人，上海开源信息技术协会理事 / 大数据专委会主任

Feedbacks



Impressed by your project!!! #134

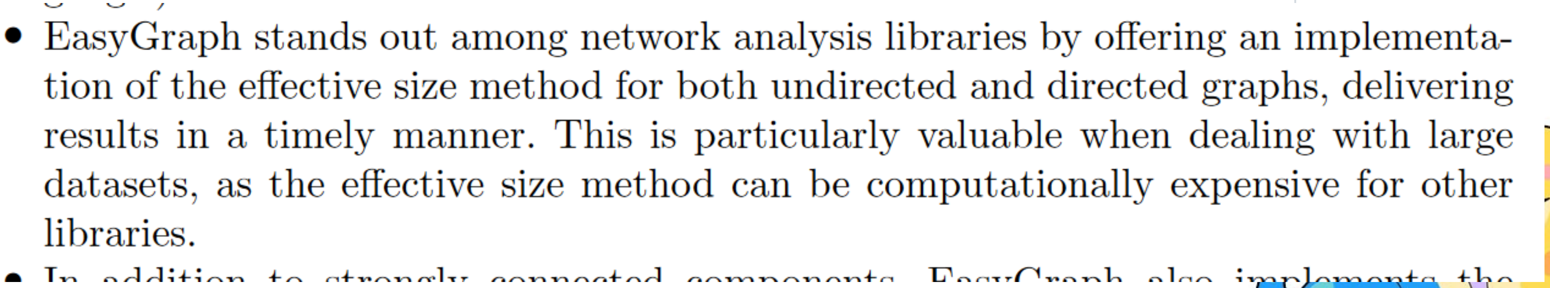
Closed xingshuohan opened this issue on Jul 12 · 3 comments

xingshuohan commented on Jul 12

I recently started using this and I'm thoroughly impressed! The documentation is exceptionally clear and made it easy for me to get started. The API is intuitive, allowing me to integrate complex graph operations into my project.

Thank much for the passionate team behind this project, so excited to see how Easy-Graph evolves and looking forward to contributing to the community.

Anyway, great work!

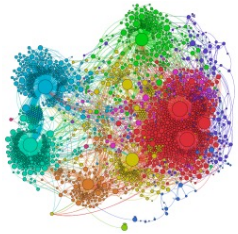


- EasyGraph stands out among network analysis libraries by offering an implementation of the effective size method for both undirected and directed graphs, delivering results in a timely manner. This is particularly valuable when dealing with large datasets, as the effective size method can be computationally expensive for other libraries.
- In addition to strongly connected components, EasyGraph also implements the

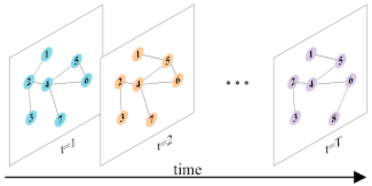
Limitations

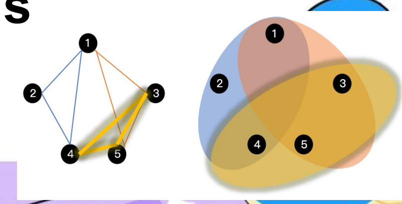
- **EasyGraph does not currently support some less-popular functions such as is_homorphic, LFR_benchmark_graph, and hypercube_graph, which NetworkX has already implemented**

- **EasyGraph does not currently focus on functions for comprehensive visualization, which are provided by Gephi and Cytoscape**



- **EasyGraph has not yet incorporated some advanced network types such as dynamic networks, bipartite networks, and heterogeneous networks. For might want to use other libraries for these networks.**





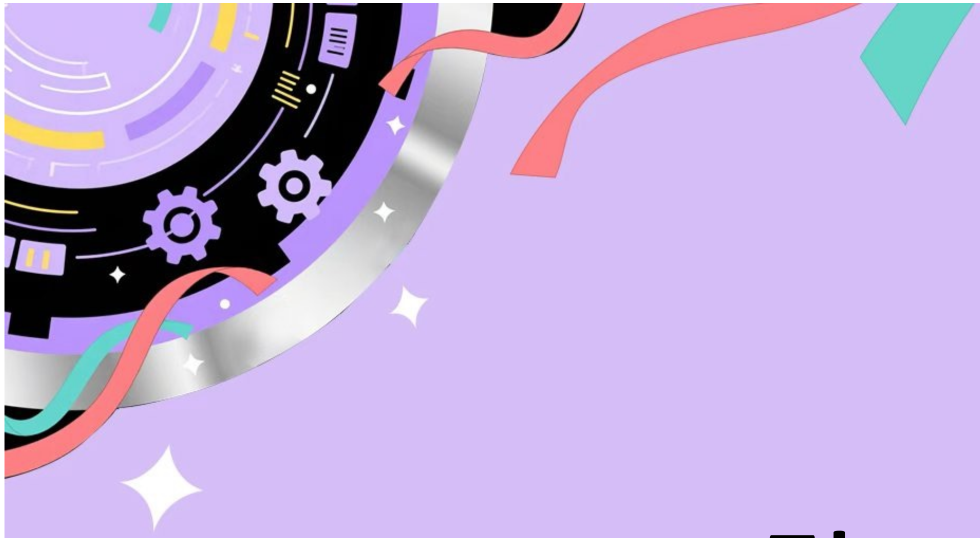
Conclusions and Future Work

Conclusions

- ✓ EasyGraph supports more formats of network data than other tools
- ✓ EasyGraph offers important specialized functions such as SHS detection and graph embedding
- ✓ For networks that contain more than thousands of nodes, EasyGraph outperforms other tools on several important network analysis tasks

Future Work

- ➢ Extend EasyGraph to support **more types of networks**
- ➢ Track and implement **more important algorithms** for SNA
- ➢ Deploy EasyGraph as a **cloud-based web service**



Thank you!

Yang Chen
https://chenyang03.wordpress.com/
Big Data and Networking (DataNET) Group
Fudan University