

-file: 输入文件路径
-alg: 具体算法
-out: 输出文件路径
-IsDirected: 是否为有向图, 默认为False;

并行编程基础

[Introduction to parallel & distributed algorithms](#)

[Applications of Parallel Computers](#)

1. Louvain社区发现算法

简介: 该算法可以用于探测和挖掘网络中的社区结构。主要思想是定义一个描述一个描述社区划分好坏的模块度函数, 通过最优化模块度找到更显著的社区划分。

FastUnfolding: -file [filepath] -alg [FastUnfolding] -IsDirected[value] -Alpha[value] -PartByIn[value] -OuterIteration[value] -InnerIteration[value] -out [outpath]

-Alpha: 在序列平衡分区 (Sequence-balanced partition) 中使用的alpha值, 默认为-1;

-PartByIn: 是否以入度划分, 默认为False;

-OuterIteration: 算法的外迭代次数;

-InnerIteration: 算法的内迭代次数;

输出为一个节点集合, 代表一个社区

论文:

[Distributed Louvain Algorithm for Graph Community Detection | IEEE Conference Publication | IEEE Xplore](#)

[Scalable distributed Louvain algorithm for community detection in large graphs | SpringerLink](#)

[A Scalable Distributed Louvain Algorithm for Large-Scale Graph Community Detection | IEEE Conference Publication | IEEE Xplore](#)

项目:

[Distributed-louvain-modularity \(sotera.github.io\)](#)

[Spark-Distributed-Louvain-Modularity](#)

2. 标签传播算法

***简介:** 该算法可以用于探测和挖掘网络中的社区结构。主要思想是从已标记的节点的标签信息来预测未标记的节点的标签信息, 利用样本间的关系, 建立完全图模型。*

LPA: -file [filepath] -alg [LPA] -IsDirected[value] -Alpha[value] -PartByIn[value] -Iteration[value] -out [outpath]

-Alpha: 在序列平衡分区 (Sequence-balanced partition) 中使用的alpha值, 默认为-1;

-PartByIn: 是否以入度划分, 默认为False;

-Iteration: 算法的迭代次数;

输出为一个节点集合, 代表一个社区

论文:

[A Degree-based Distributed Label Propagation Algorithm for Community Detection in Networks | Proceedings of the 4th ACM SIGSPATIAL International Workshop on Safety and Resilience](#)

[DHLP 1&2: Giraph based distributed label propagation algorithms on heterogeneous drug-related networks - ScienceDirect](#)

[Distributed Direction-Optimizing Label Propagation for Community Detection \(Conference\) | OSTI.GOV](#)

[LABEL PROPAGATION IN A DISTRIBUTED SYSTEM - Google LLC \(freepatentsonline.com\)](#)

[Scalable Label Propagation Algorithms for Heterogeneous Networks \(biorxiv.org\)](#)

[Shortcutting Label Propagation for Distributed Connected Components](#)

[XtraPuLP](#)

3. 结构图聚类

简介: 该算法可以用于探测和挖掘网络中的社区结构, 还可以发现网络中的离群点。主要思想是基于顶点间的结构相似性度量对顶点进行聚类。

SCAN: -file [filepath] -alg [SCAN] -Size[value] -Seed[value] -Load[value] -out [outpath]

-Size: bitmap的大小;

-Seed: 随机数种子, 默认为666;

-Load: bitmap中非零元素比例, 取值范围为[0,1];

输出为一个节点集合, 代表一个社区

论文:

[DSCAN: Distributed Structural Graph Clustering for Billion-Edge Graphs | SpringerLink](#)

[A Distributed Algorithm for Large-Scale Graph Clustering \(inria.fr\)](#)

4. 局部图聚类

简介: 该算法可以用于发现给定节点的社区结构, 而且只需访问局部的图数据, 而无需访问整个大图。

LocalClustering: -file [filepath] -alg [LocalClustering] -Loss[value] -K[value] -Alpha[value] -MaxSize[value] -out [outpath]

-Loss: 损失函数, 默认为modularity;

-K: 目标集群数;

-Alpha: 集群大小的惩罚项系数;

-MaxSize: 集群最多顶点数;

输出为一个节点集合, 代表一个社区

论文:

[Scalable local density-based distributed clustering - ScienceDirect](#)

5. K-核

简介: 该算法可以用于社区搜索应用。给定一个节点或者多个节点, 查询这些节点所在社区结构。该算法的主要思想是找到一个子图, 满足子图中每个节点都至少有k个邻居。

KCore: -file [filepath] -alg [KCore] -k [value] -out [outpath]

-k 计算core number大于等于k的点

输出为一个节点集合, 代表一个社区

论文:

[\[1103.5320\] Distributed k-Core Decomposition \(arxiv.org\)](#)

[Distributed --Core View Materialization and Maintenance for Large Dynamic Graphs | IEEE Journals & Magazine | IEEE Xplore](#)

[\(PDF\) Distributed k-Core Decomposition of Dynamic Graphs \(researchgate.net\)](#)

[Distributed k-core algorithm, CoNext '12 quick talk. | Paul Jakma - Academia.edu](#)

6. K-Truss

简介: 该算法的应用场景与k-核类似, 可以用于社区搜索应用。相比于k-核, k-truss所表达的社区结构要更为稠密。该算法的主要思想是社区结构是找到一个子图, 满足子图中每条边都包含在k-2个三角形中。

KTruss: -file [filepath] -alg [KTruss] -k [value] -out [outpath]

-k 计算truss number大于等于k的边

输出为一个节点集合, 代表一个社区

论文:

[K-truss decomposition for Scale-Free Graphs at Scale in Distributed Memory | IEEE Conference Publication | IEEE Xplore](#)

[Distributed algorithms for k-truss decomposition | IEEE Conference Publication | IEEE Xplore](#)

[KTMiner: Distributed k-truss detection in big.graphs | Semantic Scholar](#)

7. 派系过滤算法:

简介: 该算法可以用于挖掘网络中的重叠社区结构。主要思想如下: k -派系表示网络中含有 k 个节点的完全子图, 如果一个 k -派系与另一个 k -派系有 $k-1$ 个节点重叠, 则这两个 k -派系是连通的, 由所有彼此连通的 k -派系构成的集合就是一个 k -派系社区。

CliquePercolation: -file [filepath] -alg [cpct] -k [value] -out [outpath]

-k 派系约束大于等于 $k-1$ 的联通子图

输出为一个节点集合, 代表一个社区

论文:

[Distributed Clique Percolation based community detection on social networks using MapReduce | IEEE Conference Publication | IEEE Xplore](#)

[Clique percolation method: memory efficient almost exact communities | DeepAI](#)

8. 极大团枚举

简介: 该算法可以用于发现网络中得极大完全子图结构。该算法可以用于分析网络中团状的社区结构。

MaximalClique: -file [filepath] -alg [clique] -s [size] -out [outpath]

-s size大于等于 s 的团

输出到文件为若干行文本, 每行数据表示一个极大团

论文:

[Distributed Maximal Clique Computation | IEEE Conference Publication | IEEE Xplore](#)

[Distributed Maximal Clique Computation and Management | Semantic Scholar](#)

[Maximal Clique Based Distributed Coalition Formation for Task Allocation in Large-Scale Multi-agent Systems](#)

[A Distributed Algorithm to Enumerate All Maximal Cliques in MapReduce | Request PDF \(researchgate.net\)](#)

9. PageRank

简介: 该算法可以用于分析网络中节点的重要性。最早由Google创始人提出, 可以用这个算法来体现网页的相关性和重要性。

PageRank: -file [filepath] -alg [PageRank] -PageRankAlpha [value] -PageRankMaxIterations [value] -PageRankConvergence [value] -PageRankDirected [value] -out [outpath]

-PageRankAlpha: 非必选参数, 阻尼系数, 在任意时刻, 用户到达某页面后并继续向后浏览的概率, 取值范围 $(0,1)$, 通常默认为0.85;

-PageRankMaxIterations: 非必选参数, 最大迭代次数, 默认为1000;

-PageRankConvergence：非必选参数，收敛精度，取值范围(0,1)，判断迭代是否已经终止的参数，默认为0.0001；

-PageRankDirected：非必选参数，是否考虑边的方向，默认为true；

输出到文件为一个向量，代表了节点（从小到大排序）的PageRank值

论文：

[Distributed PageRank Computation Based on Iterative Aggregation-Disaggregation Methods](#)

[Distributed Pagerank for P2P Systems](#)

[Fast Distributed PageRank Computation | SpringerLink](#)

[Distributed Pagerank: A Distributed Reputation Model for Open Peer-to-Peer Network. | Request PDF \(researchgate.net\)](#)

项目：

[GitHub - sandy14795/Distributed PageRank](#)

10. Closeness中心性

简介：该算法可以用于分析网络中节点到达其它节点的难易程度。节点CC值越高，代表其在网络中处于比较中心的位置。

Closeness Centrality: -file [filepath] -alg [ClosenessCentrality] -ClosenessCentralityDirected [value] -out [outpath]

-ClosenessCentralityDirected：非必选参数，是否考虑边的方向，默认为false；

输出到文件为一个向量，代表了节点（从小到大排序）的Closeness中心性值

论文：

[STREAMER: a Distributed Framework for Incremental Closeness Centrality Computation](#)

[Incremental Closeness Centrality in Distributed Memory](#)

[Distributed Estimation of Closeness Centrality](#)

11. Betweenness中心性

简介：该算法可以用于分析网络中节点对其它“节点对间的最短路径”的重要性。

Betweenness Centrality : -file [filepath] -alg [BetweennessCentrality]

-BetweennessCentralityDirected [value] -out [outpath]

-BetweennessCentralityDirected：非必选参数，是否考虑边的方向，默认为false；

输出到文件为一个向量，代表了节点（从小到大排序）的Betweenness中心性值

论文：

[Distributed estimation of betweenness centrality_ | IEEE Conference Publication | IEEE Xplore](#)

[Distributed and Parallel Algorithm for Computing Betweenness Centrality_ | SpringerLink](#)

12. Eigenvector中心性

简介：该算法可以用于分析网络中节点的重要性。它利用特征向量中心性对节点进行重要性排名。

Eigenvector Centrality: -file [filepath] -alg [EigenvectorCentrality] -Normalization [value] -ClosenessCentralityDirected [value] -out [outpath]

-Normalization：非必选参数，是否将Eigenvector Centrality标准化，默认不进行标准化，

value=max: 把每个Eigenvector Centrality都除以最大的Eigenvector Centrality;

value=norm1: 把每个Eigenvector Centrality标准化，使它们之和为1;

value=norm2: 把每个Eigenvector Centrality都除以所有Eigenvector Centrality平方和的平方根;

- EigenvectorCentralityDirected：非必选参数，是否考虑边的方向，默认为true;

输出到文件为一个向量，代表了节点（从小到大排序）的Eigenvector中心性值

论文：

[Totally asynchronous distributed estimation of eigenvector centrality in digraphs with application to the PageRank problem | Request PDF \(researchgate.net\)](#)

[\(PDF\) Distributed Centrality Analysis of Social Network Data Using MapReduce \(researchgate.net\)](#)

13. HITS算法

简介：该算法可以用于分析网络中节点的重要性。可以用于对于节点进行重要性排名。同时，也可以找到网络中的hub节点。

HITS: -file [filepath] -alg [HITS] -HITSMaxIterations [value] -HITSConvergence [value] -out [outpath]

-HITSMaxIterations：非必选参数，最大迭代次数，默认为1000;

-HITSConvergence：非必选参数，收敛精度，取值范围(0,1)，判断迭代是否已经终止的参数，默认为0.0001;

输出到文件为一个向量，代表了节点（从小到大排序）的HITS值

论文：

[Improvement on HITS Algorithm](#)

[A Scalable Parallel HITS Algorithm for Page Ranking_| Request PDF \(researchgate.net\)](#)

项目：

[hits-algorithm · GitHub Topics · GitHub](#)

14. 广度优先搜索算法

简介：适用于图遍历的应用场景。例如搜索该节点可以到达的节点。

BFS: -file [filepath] -alg [BFS] -out [outpath] -BFSRoot [value]

-BFSRoot：必选参数，bfs的起始顶点id

输出到文件多行，每行一个顶点ID，为从root节点开始能遍历到的所有节点

论文：

[Parallel Breadth-First Search on Distributed Memory Systems](#)

[A Work-Efficient Parallel Breadth-First Search Algorithm \(or How to Cope with the Nondeterminism of Reducers\)](#)

[Large-Scale Parallel Breadth-First Search \(aaai.org\)](#)

[BFS and Coloring-based Parallel Algorithms for Strongly Connected Components and Related Problems](#)

[A Work-Efficient Parallel Breadth-First Search Algorithm \(or How to Cope with the Nondeterminism of Reducer Hyperobjects\) \(mit.edu\)](#)

Evaluating Parallel Breadth-First Search Algorithms for Multiprocessor Systems

[Parallel and Distributed Breadth-First Search](#)

[Distributed Breadth-First Search with 2-D Partitioning*](#)

[Distributed BFS algorithms \(mit.edu\)](#)

[Distributed Memory Breadth-First Search Revisited: Enabling Bottom-Up Search](#)

[Parallel Breadth-First Search on Distributed Memory Systems](#)

15. 最短路径算法

简介：可以用于计算两个节点间最短路径的应用。

SSSP: -file [filepath] -alg [SSSP] -out [outpath] -SSSPRoot [value]

-SSSPRoot：必选参数，SSSP的起始顶点id

输出到文件n行，n是图中顶点个数，第i行为从root点到id为i-1这个点的最短路径长度

论文：

[\[1711.01364\] A Faster Distributed Single-Source Shortest Paths Algorithm \(arxiv.org\)](#)

[Performance Analysis of Single-source Shortest Path Algorithms on Distributed-memory Systems improved-distributed-stoc18.pdf \(q3r.github.io\)](#)

[A Faster Distributed Single-Source Shortest Paths Algorithm \(ieee-focs.org\)](#)

[Parallel Implementation of the Single Source Shortest Path Algorithm on CPU-GPU Based Hybrid System | Journal of Computer Science IJCSIS - Academia.edu](#)

[DSMR: A Parallel Algorithm for Single-Source Shortest Path Problem](#)

项目:

[GitHub - PepperoniBlvd/Distributed-SSSP](#)

16. 连通分量算法

简介: 可以用于发现图的连通块。

CC: -file [filepath] -alg [CC] -out [outpath]

输出到文件, 共包含n行, n是图中顶点个数, 每行代表一个顶点和它所属的社区, 由 < nodeID, CommunityID >的二元组数据组成。nodeID表示顶点id为nodeID, CommunityID表示id为nodeID的顶点经过CC计算后最终所属的社区ID (社区ID为该社区中某一顶点的ID)

论文:

[\(PDF\) Connected Components on Distributed Memory Machines \(researchgate.net\)](#)

[Communication-efficient Massively Distributed Connected Components | IEEE Conference Publication | IEEE Xplore](#)

17. 强连通分量算法

简介: 可以用于发现强连通图。

SCC: -file [filepath] -alg [SCC] -out [outpath]

输出到文件, 共包含n行, n是图中顶点个数, 每行代表一个顶点和它所属的社区, 由 < nodeID, CommunityID >的二元组数据组成。nodeID表示顶点id为nodeID, CommunityID表示id为nodeID的顶点经过SCC计算后最终所属的社区ID

论文:

[Finding strongly connected components in distributed graphs - ScienceDirect](#)

[Identifying Strongly Connected Components on Distributed Networks](#)

18. 三角形计数算法

简介: 用于计算图中的三角形个数。

TriangleCounting: -file [filepath] -alg [TriangleCounting] -out [outpath]

输出TriangleCounting发现的三角形个数

论文:

[DistTC: High Performance Distributed Triangle Counting](#)

[Distributed, Shared-Memory Parallel Triangle Counting \(nsf.gov\)](#)

[\(PDF\) Tri-Fly: Distributed Estimation of Global and Local Triangle Counts in Graph Streams \(researchgate.net\)](#)

[Asynchronous Distributed-Memory Triangle Counting and LCC with RMA Caching | IEEE Conference Publication | IEEE Xplore](#)

19. 聚类系数算法

简介: 用于度量网络的聚类系数。

ClusterCoefficient: -file [filepath] -alg [ClusterCoefficient] -out [outpath]

输出到文件，共包含n+1行，n是图中顶点个数。第一行是全局图聚类系数，第i行 (i>1) 表示顶点id为i-2的点的局部聚类系数

论文:

[Parallel Clustering Coefficient Computation using GPU](#)

[\(PDF\) Efficient Parallel Hierarchical Clustering Algorithms. \(researchgate.net\)](#)

20. 链接预测算法

简介: 该算法可以用于分析和预测任意两个没有连边的节点是否会形成边。主要是链接预测旨在预测图中丢失的边, 或者未来可能会出现的边。算法主要用于判断相邻的两个节点之间的亲密程度。通常亲密程度越大的节点之间的亲密分值越高。

LinkPrediction: -file [filepath] -alg [ClusterCoefficient] -s[value] -t[value] -out [outpath]

-s 起始节点ID

-t 终止节点ID

输出到文件，包含一个值，表示s节点到t节点的链接预测值。

论文:

[Distributed Real Time Link Prediction on Graph Streams | IEEE Conference Publication | IEEE Xplore](#)

[LinkPred: a high performance library for link prediction in complex networks - PubMed \(nih.gov\)](#)

[Parallel Link Prediction in Complex Network Using MapReduce \(researchgate.net\)](#)

项目:

[Distributed Link Prediction — DGL 0.10 documentation](#)

[Parallel-FNN/link_prediction.cpp at main · grvndnz/Parallel-FNN · GitHub](#)

21. k阶度算法

简介：计算图中节点的k步邻居个数。

kHopDegree: -file [filepath] -alg [kHopDegree] -out [outpath] -kHopDegreeRoot [value] -kHopDegreeK [value]

-kHopDegreeRoot：必选参数，要查询的顶点ID

-kHopDegreeK：必选参数，要查询的邻居的度数

输出到文件为若干行文本，每行数据表示一个k度邻居的顶点id

论文：

[ESTI: Efficient k-Hop Reachability Querying over Large General Directed Graphs | SpringerLink](#)

22. 子图匹配算法

简介：该算法可以用于子图匹配查询的应用。给定一个模式子图，查找大图数据中与该模式子图同构的子图。

SubgrapMatching: -file [filepath] -alg [SubMatching] -qGraph [value] -out [outpath]

-qGraph: 必选参数，要匹配的模式图

输出到文件为若干行文本，每行数据表示一个匹配的子图集合

论文：

[Efficient distributed subgraph similarity matching | SpringerLink](#)

[Distributed Top-k Subgraph Matching in A Big Graph \(Journal Article\) | NSF PAGES](#)

[\(PDF\) Optimized Distributed Subgraph Matching Algorithm Based on Partition Replication \(researchgate.net\)](#)