Graph Mining

Yuwei Zhang
MCDS
CMU
yuweiz1@andrew.cmu.edu

Silun Wang
MCDS
CMU
silunw@andrew.cmu.edu

April 19, 2016

1 Introduction

Many realworld datasets contain valueable information to be discovered. In this paper, we present a graph mining tool – **Graph Miner** which employs simple SQL commands for efficiently mining large datasets through PostgresSQL. On 15 realworld datasets, we present our discoveries about global patterns and anomoly detections.

2 Phase1: Survey

2.1 Papers read by Yuwei Zhang

Evaluating cooperation in communities with k-core structure [6]

- *Problem Definition*: The paper focuses on community detection and evalutation, which means dense connections among some of the nodes. The author make some novel changes to the k-core concepts including updating mtric for evaluating cohesiveness, assigning weights on the edges and other extended experimental evaluation.
- *Summary*:The original k-cores algorithm keeps deleting nodes whose degree are less than k and thus take the number of each set of vertices in the subgraph to do the evalution, which fails in the case where many co-authors have equal weight. This paper improves the method to define a co-authorship edge weight instead and recompute the evalution metrics with restrictions considered. In the expriment stage, testing on an unfiltered graph turns out to be extremely biased while on a filtered one(those co-author a lot) the results seem to be resonable. When weights graph method is applied, the extrem cases where k is too big are ignored and the algorithm gives better results.
- *Shortcomings*:There is no standard metrics to evaluate these algorithms/methods propsed in the paper. Also when we consider the graphs as social networks, where the relationship between two nodes are more than just co-author, for instance we have flollow,

like, dislike, the weighted method should be furthur adjusted and it might be hard to derive the best weighting formula.

Vertexica: your relational friends for graph analytics [7]

- *Problem Definition*: To build a graph analysis tool, Vertexica, on top of a rdb that supports vertex-centric query interface. The system leverage the realational features and enable better graph analysis.
- *Summary*:Vertexica supports user-friendly and high-performance graph analysis by injecting data storage, query processing and query interfaces and supports various kinds of relational database. The system consists of four main components: physical storage to store data, coordinator as the center management driver, worker as the conntainer for the computation programs and vertex computation to process user queries. Vertexica also take several optimization techniques including: table union instead of table join, paralleling workers to work on multicores or multi machines, vertex batching to partition the table and create new tables other than update the orgin information to boost the performance. The paper also includes some use case demonstrations.
- *Shortcomings*:Hand-coded sql implementations give even better performance in the experiments of the paper. Is it possibel to furthur optimize the performance when using the user-friendly vertex-centric query interface?

Visual Exploaration of Collaboration Networks based on Graph Degeneracy [5]

- *Problem Definition*:To build a system that supports visual exploration of collaboration networks based on ranking of the nodes and filtering methods on the edges. It works on DBLP and is suitable for the large-scale networks.
- *Summary*:The idea of graph degeneracy is derived from the conceptsof k-cores, which is introduced in previous paper(the one just summarized). Basically in this system, it extracts the co-authorship graph using the algorithm described in the other paper using filtered weighted egde algorithm, and then partition the graph to f-cores based on the Trim process. Then comes the ranking, by repeatedly performing the Trim procedure to remove more vertexes and in the end stores in the relation databse for furthur query. The system can be useful to demonstrate bibliographic data.
- *Shortcomings*:For huge graphs, the k-core process may be extremly time-consuming. And it will take a long time for the system to reflect the updates in the graph.

2.2 Papers read by Silun Wang

The first paper was the Belief Propogation paper by Wolfgang Gatterbauer, Stephan Gunnemann, Danai Koutra, and Christos Faloutsos [4]

- *Problem Definition*: In big social networks, sometimes we need to infer the labels for particular nodes via transductive inference or semi-supervised learning. The classical belief propagation algorithm is widely used in such scenario, but it does not guarantee

convergence in loopy graphs. In this paper, the authors propose Linearized Belief propagation and Single-pass belief propagation which are based on different restrictions and assumptions and much faster than BP.

- *Summary*: In a nutshell, LBP and SBP have the following advantages over BS:
  1. Have convergence guarantees
  2. Have closed-form solutions, thus reducing computational cost
  3. Can be implemented on standardized SQL
  4. SBP can be updated incrementally

  LBP requires messages are normalized, thus the final belief matrix can be calculated via elegant matrix operations. SBP is based on the assumption that the impact of inference damps with length of paths. To obtain the final belief matrix, each node and each edge only need to be visited once.
- *Shortcomings*: The Daubechies wavelets require a wrap-around setting, which may lead to non-intuitive results.

The second paper was the k-core decomposition paper by Ignacio Alvarez-Hamelin, Luca DallAsta, Alain Barrat, and Alessandro [2]

- *Problem Definition*: To visualize large complex networks is a big challenge, especially when you want clarity of graph and maintaining as much information as possible in the meantime. In this paper, the authors present an effective algorithm k-core decomposition to visualize large complex networks in 2D dimension.
- *Summary*: K-core decomposition introduces several terms: coreness, shell, cluster. It assigns each vertex a polar coordinate, thus visualizing a large complex network in 2D dimension while preserving relative hierarchical structures, connectivity and clustering properties, as well as interrelationship between hierarchies. Whats more, the overall time complexity is only linear as $O(n + e)$.
- *Shortcomings*: In order to obtain a readable layout, we need to tune several parameters. Can we learn these parameters automatically? Also, for huge networks, even a k-core decomposed graph seems to be nasty. Future work might need to combine nodes into a cluster and visualize a cluster via a simplified motif representation.

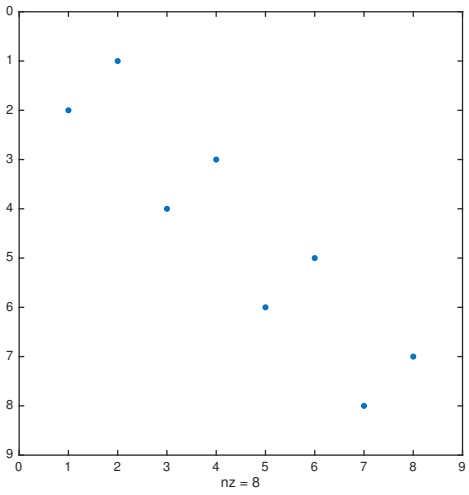
The third paper was the visualization paper by Cody Dunne and Ben Shneiderman [3]

- *Problem Definition*: Big data explosion results in huge and complex networks. In order to understand the relationship between entities and also individual attributes, traditional statistical charts are not applicable. Node-link diagrams are introduced and quickly excels among others. However, some node-link diagrams require relatively large screen space while containing little or repeated information, and optimization for the layout is NP hard. We need a more simplified visualization method which preserves important information.

- *Summary*: Authors of this paper on one hand defines three kinds of motifs: fan, connector and clique. A fan consists of a head node connected to leaf nodes with no other neighbors. A connector motif consists of functionally equivalent span nodes that solely link a set of anchor nodes. A clique motif consists of a set of member nodes in which each pair is connected by at least one link. On the other hand, the author presents an effective algorithm for motif detection with polynomial time complexity. For example, they use the obvious algorithm for detecting fan motifs which has a run time complexity of O(|*G.nodes*|*average neighbor count) and in order to find all cliques they use the Tomita algorithm. After replacing the motifs with more representative glyphs, the graph requires much less screen space and layout effort. It helps us more easily understand the network and even discover some hidden relationships.
- *Shortcomings*: Users need to be trained for a short time to fully understand this new representation. It is ambiguous in choosing clique motifs because they often overlap with each other. Future work could present users with these overlaps and relative confidence on different partitions.

3 Phase1: Unit Tests

3.1 Test Case I



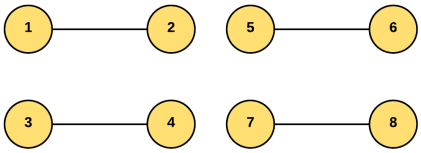


Figure 2: Graph

Figure 1: Adjacency Matrix

```
k core value:
[(1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1)]

Degree distribution:
1 --> 8

Number of connected components: 4
```

3.2 Test Case II

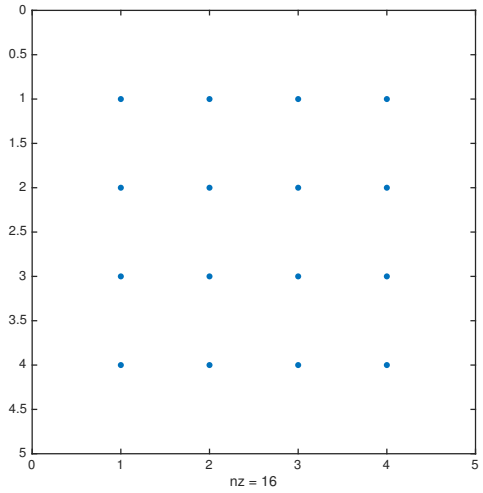


Figure 3: Adjacency Matrix

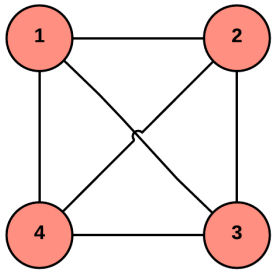


Figure 4: Graph

```
k core value:
[(1, 3), (2, 3), (3, 3), (4, 3)]

Degree distribution:
3 --> 4

Number of connected components: 1
```

3.3 Test Case III

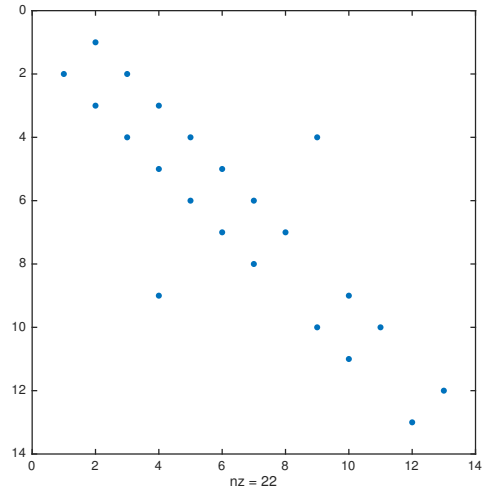


Figure 5: Adjacency Matrix

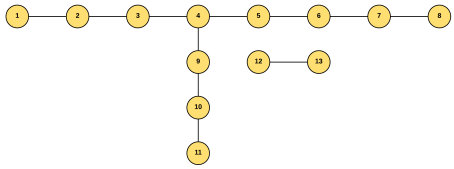


Figure 6: Graph

```
k core value:
[(1, 1), (8, 1), (11, 1), (12, 1), (13, 1), (2, 1), (7, 1),
(10, 1), (3, 1), (6, 1), (9, 1), (5, 1), (4, 1)]

Degree distribution:
1 --> 5
2 --> 7
3 --> 1

Number of connected components: 2
```

3.4 Test Case IV

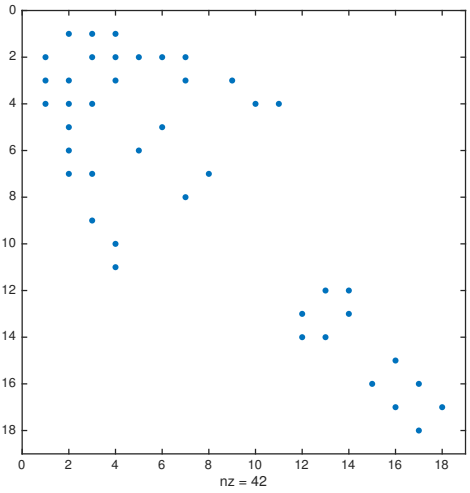


Figure 7: Adjacency Matrix

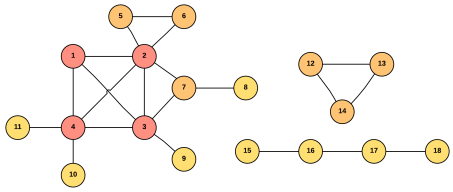


Figure 8: Graph

```
k core value:
[(8, 1), (9, 1), (10, 1), (11, 1), (15, 1), (16, 1), (17, 1), (5, 2),
(6, 2), (12, 2), (13, 2), (14, 2), (7, 2), (4, 3), (1, 3), (2, 3), (3, 3)]

Degree distribution:
1 --> 6
2 --> 7
3 --> 2
5 --> 2
6 --> 1

Number of connected components: 3
```

3.5 Test Case V

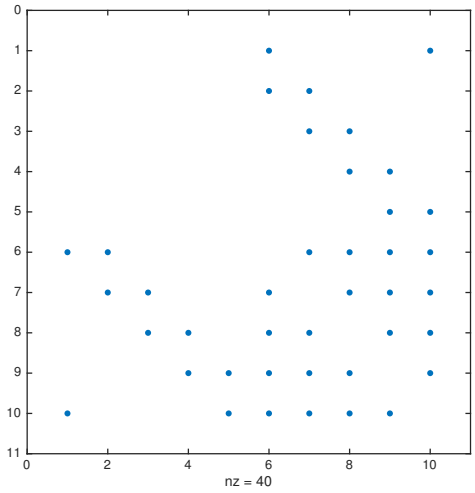


Figure 9: Adjacency Matrix

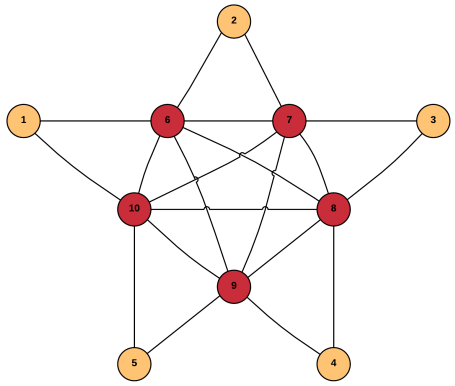


Figure 10: Graph

```
k core value:
[(2, 2), (3, 2), (4, 2), (5, 2), (1, 2), (7, 4), (6, 4), (8, 4), (9, 4), (10, 4)]

Degree distribution:
2 --> 5
6 -- > 5

Number of connected components: 1
```

4 Phase1: Experiment Results

4.1 Core value

dataset	id = 0	id = 17	id = 9422	id = 18475	id = 27763
soc-Slashdot0811	43	43	43	43	15
soc-Epinions1	67	43	2	10	4

4.2 Degeneracy value

soc-Slashdot0811 : 55
soc-Epinions1 : 67

4.3 Core value distribution

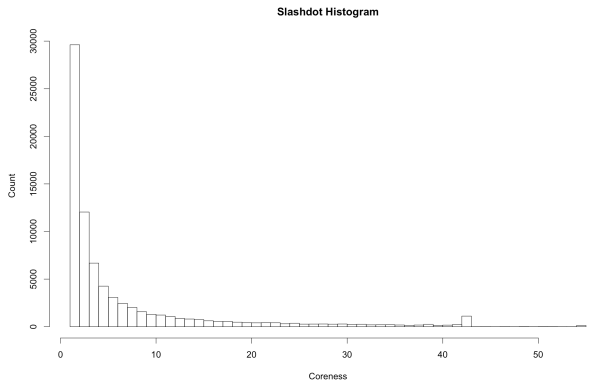


Figure 11: soc-Slashdot0811

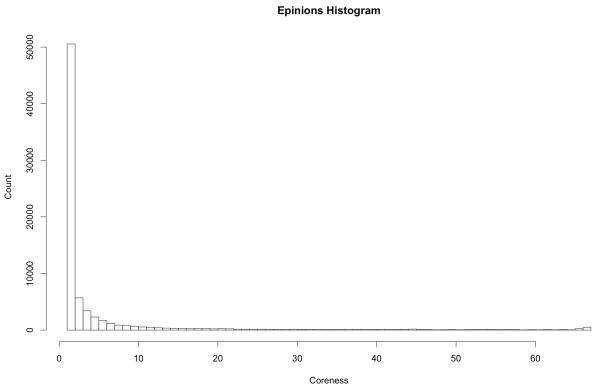


Figure 12: soc-Epinions1

5 Phase2: Indexing

5.1 Dataset

We choose data from Stanford Large Network Dataset Collection, including social network, collaboration network and peer-to-peer network data.

dataset	Type	Nodes	Edges	Description
ego-Facebook	Undirected	4,039	88,234	Social circles from Facebook (anonymized)
wiki-Vote	Directed	7,115	103,689	Wikipedia who-votes-on-whom networt
ca-GrQc	Undirected	5,242	14,496	Collaboration network of Arxiv General Relativity
ca-HepTh	Undirected	9,877	25,998	Collaboration network of Arxiv High Energy Physics Theory
p2p-Gnutella08	Directed	6,301	20,777	Gnutella peer to peer network from August 8 2002

5.2 Performance

In the following sections, we present the performance of different indexing methods {non-clustering index on `source`, clustering index on `source`, composite index on `source` and `destination`} $\times$ different node ordering methods {random ordering, coreness ordering, pagerank ordering}.

5.2.1 Undirected Social Network: Facebook

Facebook dataset has more edges and a long tail in degree distribution, hence it takes relatively longer run-time for our algorithms compared with other networks.

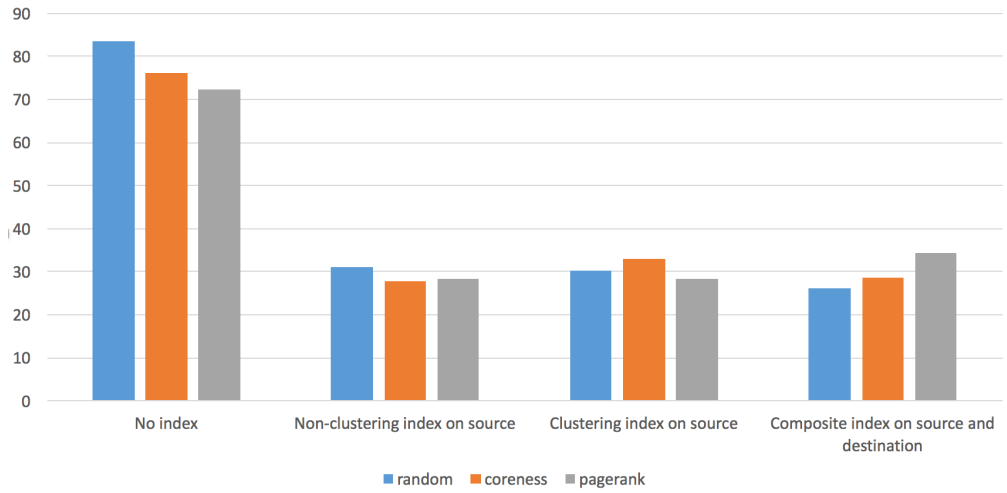


Figure 13: Performance

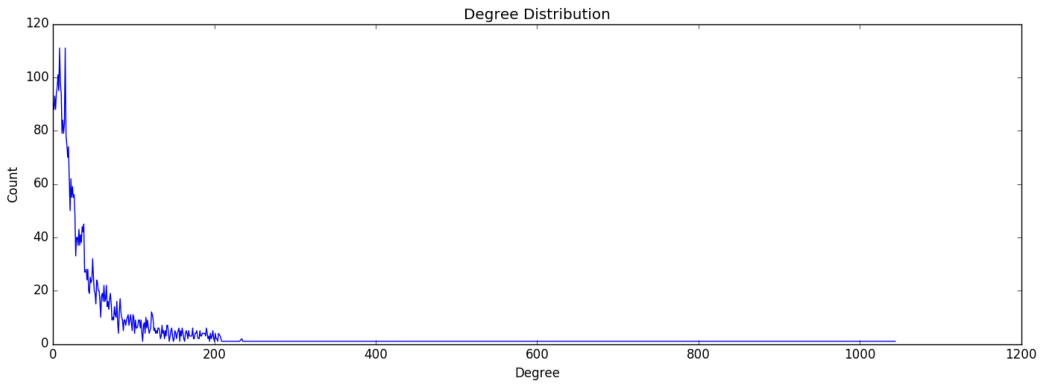


Figure 14: Degree Distribution

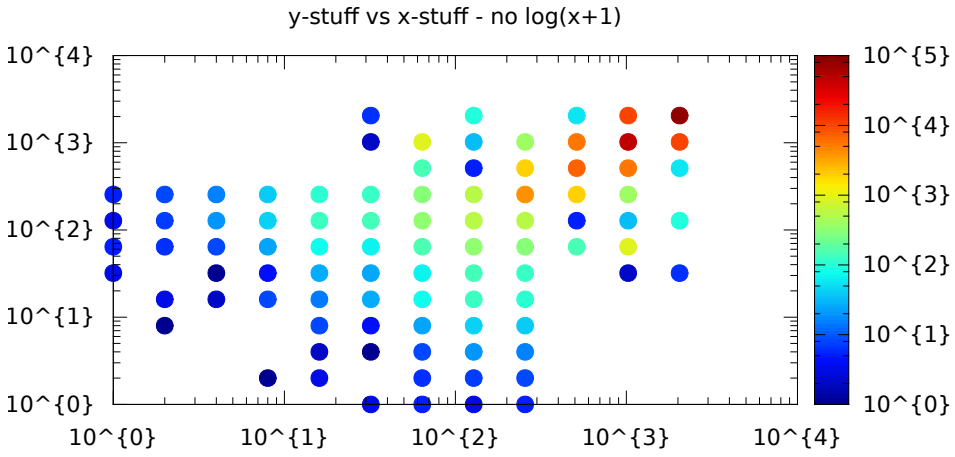


Figure 15: Scatter Plot

5.2.2 Directed Social Network: Wiki Vote

Wiki-Vote also belongs to social network dataset. From its degree distribution, we see it approximates power law and has relatively more edges. Majority of the nodes has small degrees while a few 'popular' has a degree over 1000.

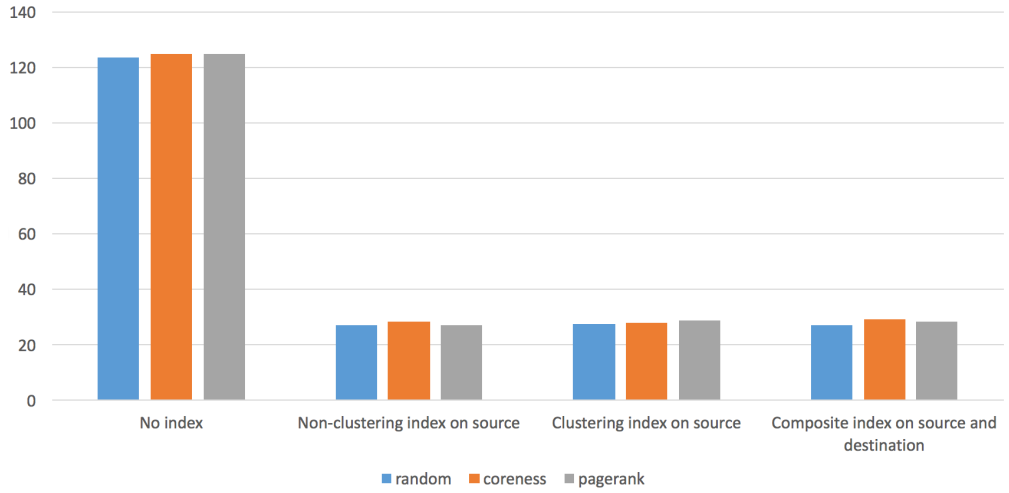


Figure 16: Performance

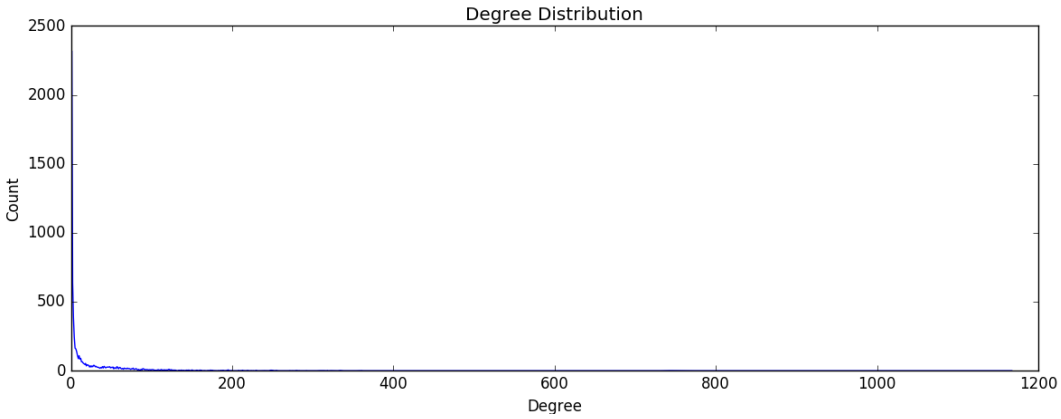


Figure 17: Degree Distribution

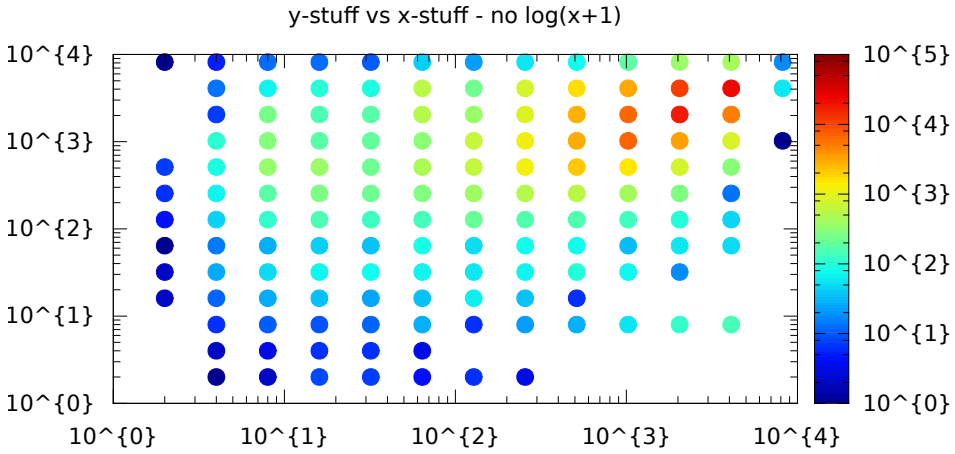


Figure 18: Scatter Plot

5.2.3 Collaboration Network

General Relativity and Quantum Cosmology collaboration network

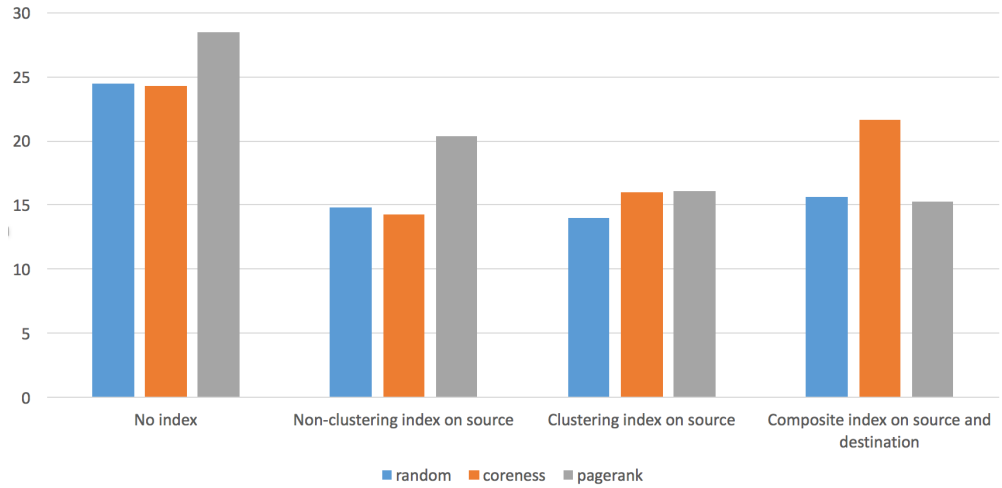


Figure 19: Performance

High Energy Physics - Theory collaboration network

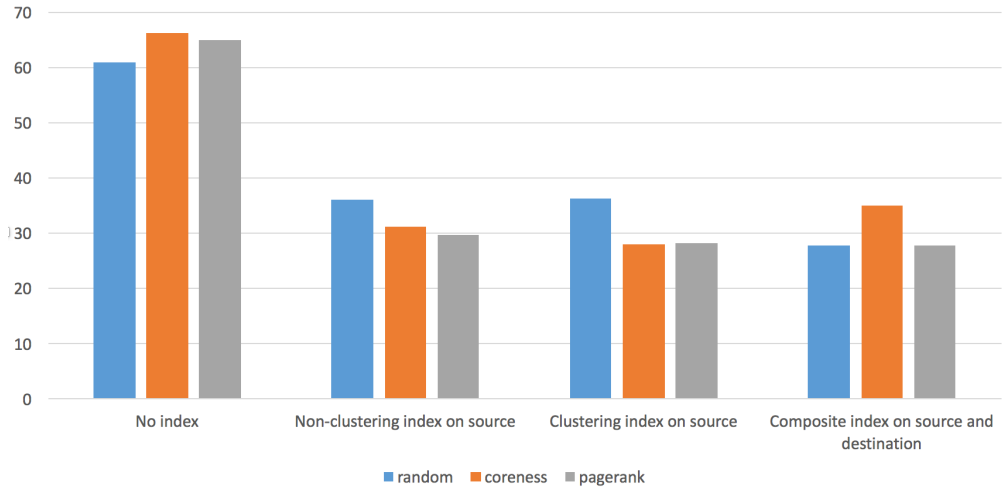


Figure 20: Performance

Due to limited pages, we only list the degree distribution and scatter plot of General Relativity and Quantum Cosmology collaboration network as follows. They are quite similar.

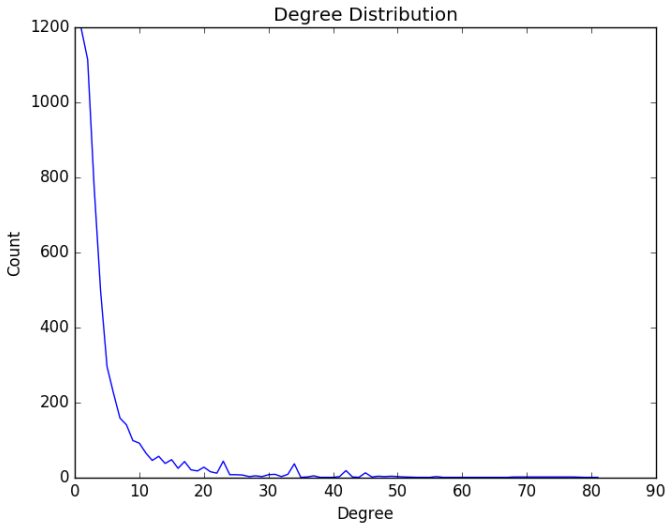


Figure 21: Degree Distribution

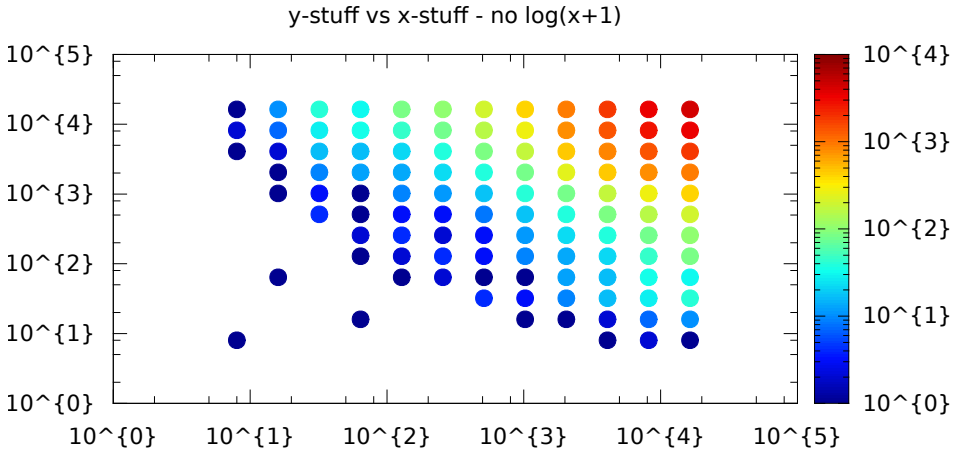


Figure 22: Scatter Plot

5.2.4 P2P Network

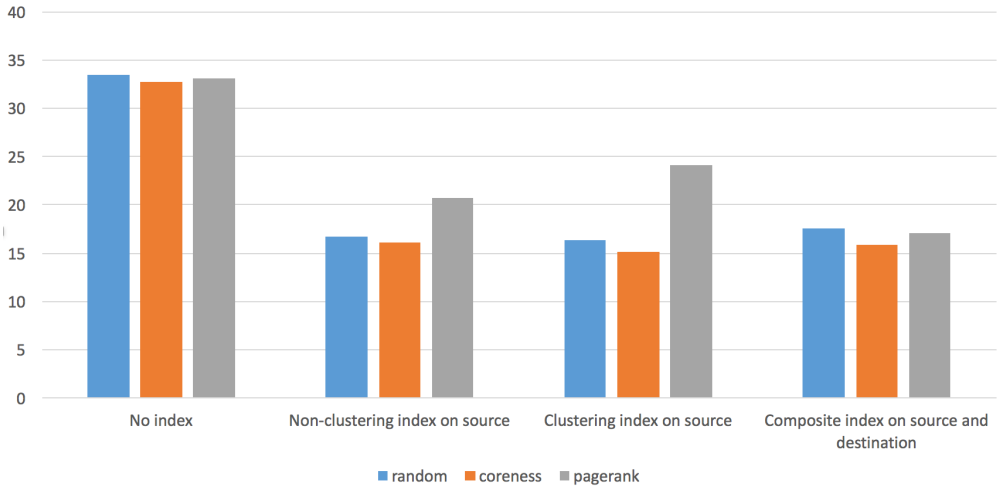


Figure 23: Performance

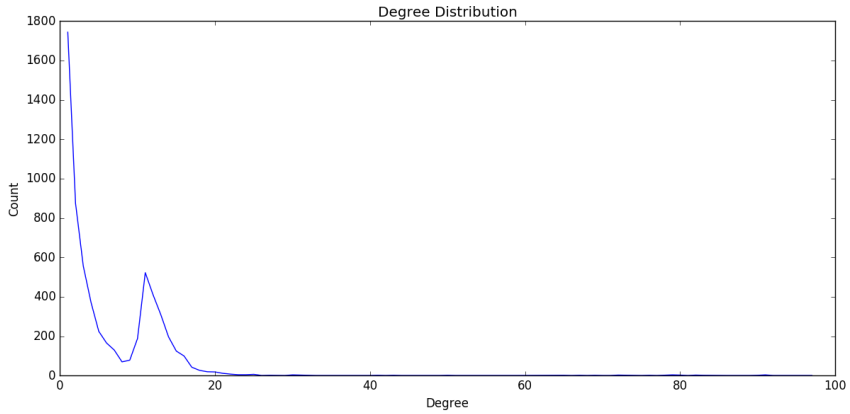


Figure 24: Degree Distribution

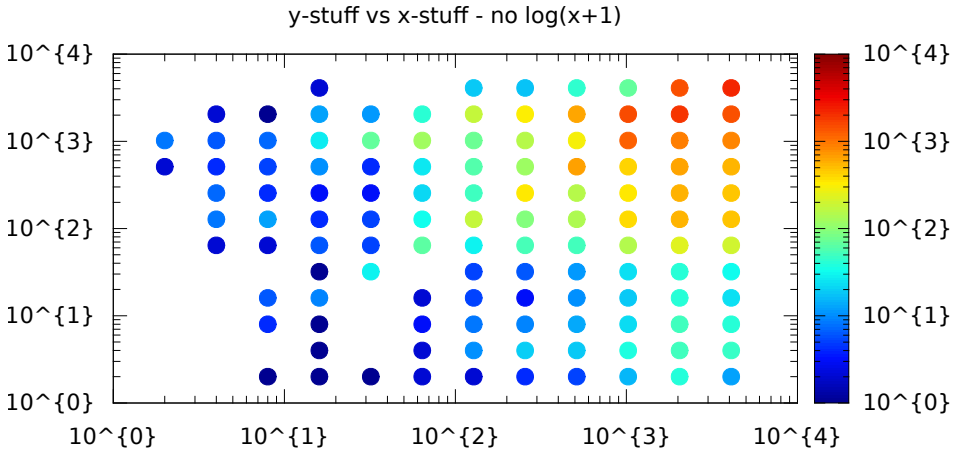


Figure 25: Scatter Plot

5.3 Observation

The most obvious result we can see from the experiment is that the running time get 2-4 times faster when using index. Degree of improvement depends on dataset types.

As for different types of index (clustering on source, non-clustering on source and composite index on source and destination), they all improve the running time and their impacts are similar. In some dataset e.g. Facebook, data with composite indexes run slightly faster. In other dataset like wiki-Vote ranked by coreness, clustering index gives a slightly better performance. So the performance of different indexes is dependent on dataset.

Clustering index stores actual rows on the disc and search for actual rows while non-clustered index store pointers and pointers point to where actual rows are stored. Generally clustering index would provide a faster way of searching, but in our experiment, we not only

have searching, but also a huge amount of work of deleting, which may slow down using clustering index. [1]

We also experiment on preprocessing the data, ranking the source node id in the initial dataset by coreness and page rank. And we find out there is not much difference in running time when we reorder the dataset first. Adding indexes to the reordered dataset also show similar trend as the original random data.

Another interesting finding is that the five datasets we use have different number of nodes and edges, but the running time of the algorithm is not proportional to the number of nodes. For example, ca-HepTh dataset has 9877 nodes while Facebook has 4024 nodes; however, the raw running time of Facebook is 83s while ca-HepTh is 61s. But when we look at their edges, we find that the edges in Facebook is 88234, almost 4 times the edges in ca-HepTh. So we may say that the number of edges determines the running time. In addition, as we look at dataset wiki-Vote, it has the largest number of edges: 103,689, and we see the raw running time is longer than the other datasets. Surprisingly, after adding index to it, the running time decreases dramatically from 123s to 27s, even shorter than the other datasets.

6 Phase3: Discovery

6.1 User-manual documentation on k-degeneracy

This section describes the idea of k-core and the implementation of analyzing the degeneracy of of a graph using SQL.

First, we look at the definition of coreness: (definitions come from project writeup) A node in an undirected graph has coreness k, if it has k or more neighbors that have coreness k or higher, and k is the maximum such integer for node n. After we know the coreness of each node in the graph, we can get the degeneracy of a graph. The formal definition is: Degeneracy D of a graph is the highest coreness among the nodes of thegraph.

Now we have seen the definition of degeneracy, so the general idea of finding the coreness of each node is described below: iterate from k = 1, and we want to find nodes whose coreness = k; for each k, find those nodes who has neighbors less than or equal to k, so those numbers are of coreness k; delete those nodes from the graph; repeat until we can not find any of the node who has neighbors $\leq$ k, all the nodes left must have coreness greater than k, so we increase k by 1; terminate when there is no node in the graph.

Part of the SQL code are provided below:

```
-- 1. start from k = 1, loop when we can still find nodes with neighbors <= k
--    create a temp table to store all nodes that we find with neighbors <= k
INSERT INTO TMP_TABLE
SELECT src_id, COUNT(*) AS neighbor FROM GM_TABLE
GROUP BY src_id HAVING count(*) <= k;

-- 2. check if there is no nodes satisfying the conditions, increase k by 1 and
   continue from the start of the loop
```

```
SELECT COUNT(*) FROM TMP_TABLE;
-- if count is 0, k += 1 and continue

-- 3. save those nodes to a permanent table with their coreness
INSERT INTO GM_KCORE
SELECT src_id, k AS coreness FROM TMP_TABLE;

-- 4. delete those nodes from the original table
DELETE FROM GM_TABLE
WHERE src_id IN (SELECT src_id FROM TMP_TABLE);
DELETE FROM GM_TABLE
WHERE dst_id IN (SELECT src_id FROM TMP_TABLE);

-- 5. terminate if there is no node in the original table
SELECT COUNT(*) FROM GM_TABLE;
-- if count is 0, break the loop

-- 6. calculate the degeneracy of the graph, which is the largest coreness among
    all nodes
SELECT MAX(coreness) FROM GM_KCORE;
```

6.2 Dataset

Our datasets are adopted from SNAP (Stanford Large Network Dataset Collection) `http://snap.stanford.edu/data/index.html`.

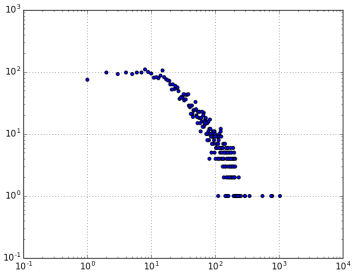
dataset	Type	Nodes	Edges	Description
Facebook	Undirected	4,039	88,234	**Social network** Social circles from Facebook (anonymized)
soc-Slashdot0811	Directed	77,360	905,468	**Social network** Slashdot social network from November 2008
soc-Slashdot0922	Directed	82,168	948,464	**Social network** Slashdot social network from February 2009
soc-Epinions1	Directed	75,879	508,837	**Social network** Who-trusts-whom network of Epinions.com
wiki-Vote	Directed	7,115	103,689	**Wikipedia social network** Wikipedia who-votes-on-whom network
email-Enron	Undirected	36,692	183,831	**Communication network** Email communication network from Enron
ca-HepTh	Undirected	9,877	25,998	**Collaboration network** of Arxiv High Energy Physics Theory
ca-GrQc	Undirected	5,242	14,496	**Collaboration network** of Arxiv General Relativity
ca-AstroPh	Undirected	18,772	198,110	**Collaboration network** of Arxiv Astro Physics
p2p-Gnutella24	Directed	26,518	65,369	**P2P Network** Gnutella peer to peer network from August 24 2002
p2p-Gnutella25	Directed	22,687	54,705	**P2P Network** Gnutella peer to peer network from August 25 2002
oregon1-010331	Undirected	10,670	22,002	**Autonomous system** peering information inferred from Oregon route-views from March 31
oregon1-010519	Undirected	11,051	22,724	**Autonomous system** peering information inferred from Oregon route-views from May 19
cit-HepPh	Directed	34,546	421,578	**Citation network** Arxiv High Energy Physics paper citation network
cit-HepTh	Directed	27,770	352,807	**Citation network** Arxiv High Energy Physics paper citation network

6.3 Results

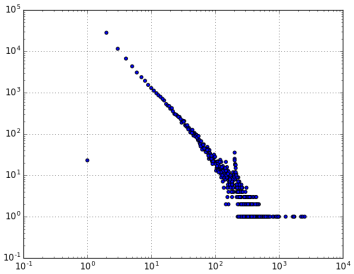
All the graphs shown below are in log-log scale.

6.3.1 Degree Distribution

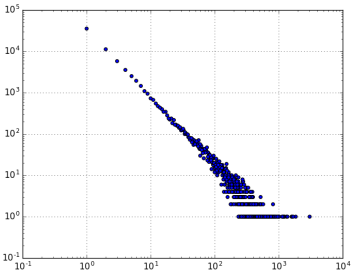
Scatter plot x-axis: degree y-axis: count



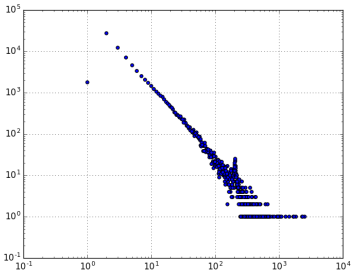
Facebook



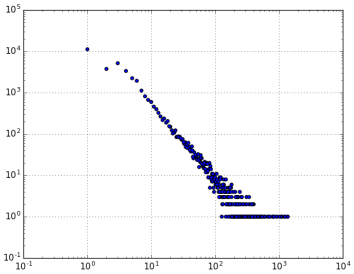
soc-Slashdot0811



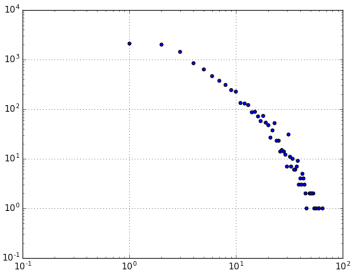
soc-Epinions1



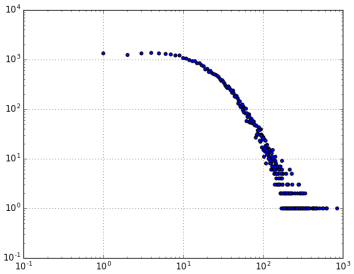
soc-Slashdot0922



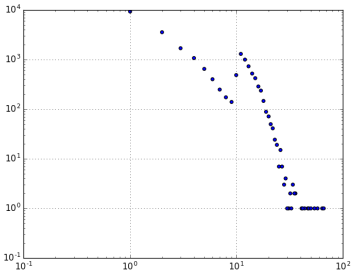
email-Enron



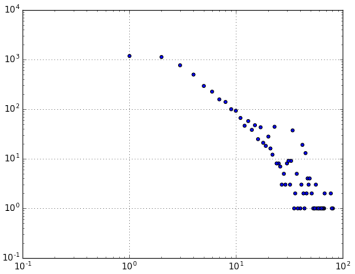
ca-HepTh



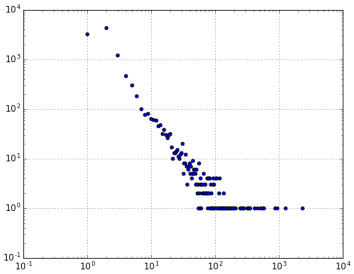
cit-HepPh



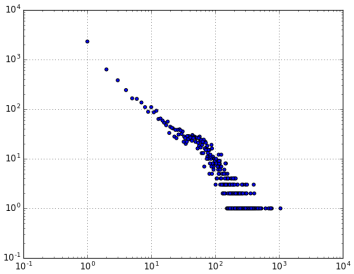
p2p-Gnutella25



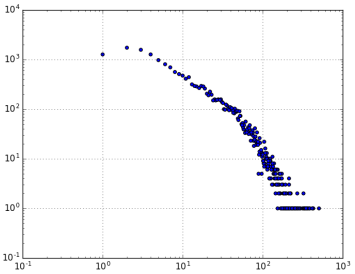
ca-GrQc



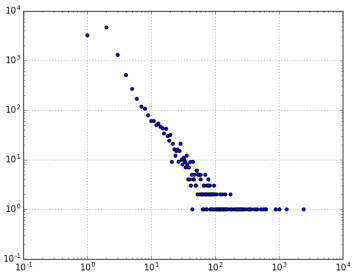
Oregon1-010331



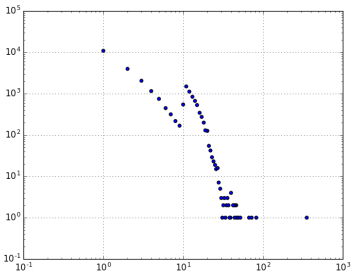
wiki-Vote



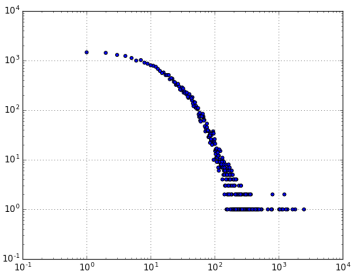
ca-Astro



Oregon1-010519 Observations



p2p-Gnutella24

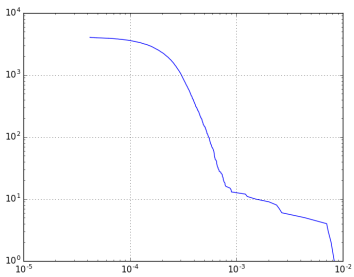


cit-HepTh

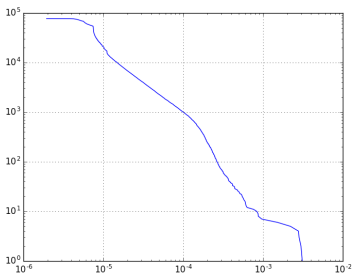
- Most datasets exhibit a strong power law relation between degree and count. A large fraction of nodes (users) have very low degrees or very few connections.
- **P2P networks** (p2p-Gnutella24, pep-Gnutella25) exhibits a spike around degree = 10, implying that people tend to connect with 10 peers.
- Some networks such as Facebook and **citation networks** do not follow a power law in degree range [1 - 10]. They look more like log-logistic. Taking Facebook for example, there are a lot of people with 1-10 friends, but the number of people with 1 friend and 10 friend does not differ much. Speaking in a formal way, nodes with a degree between 1 - 10 have similar counts.
- We can also spot some outliers with only one connection in {soc-Slashdot0811, soc-Slashdot0922}, which probably means newly registered accounts in Slashdot Zoo, but they are less likely to get only one friend compared to the normal trend of other networks. Another outlier can be observed in dataset {p2p-Gnutella 24} where one node has a much higher degree = 200, probably because it functions as the central server in p2p network.

6.3.2 Pagerank Distribution

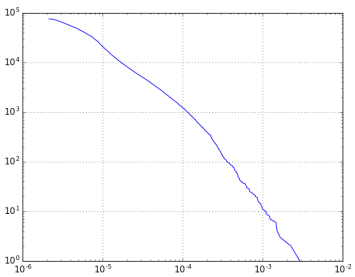
CCDF x-axis: pagerank score y-axis: count($\geq$ score)



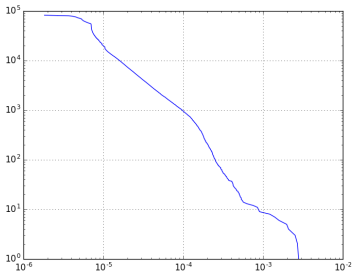
Facebook



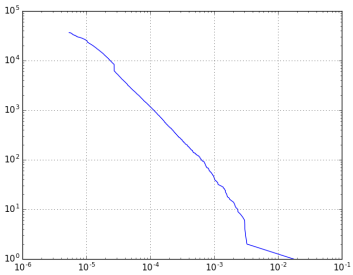
soc-Slashdot0811



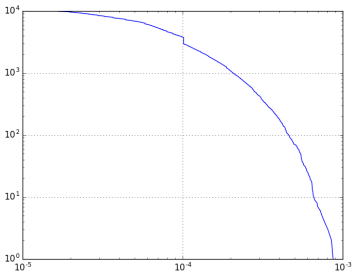
soc-Epinions1



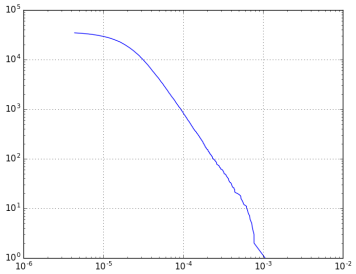
soc-Slashdot0922



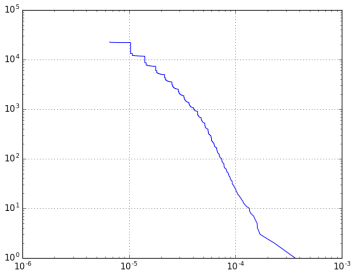
email-Enron



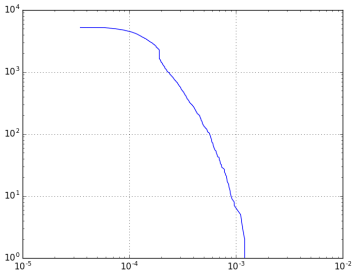
ca-HepTh



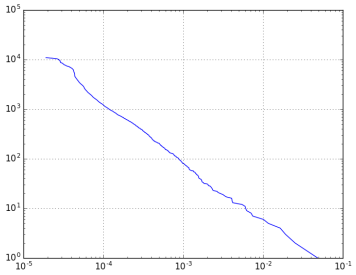
cit-HepPh



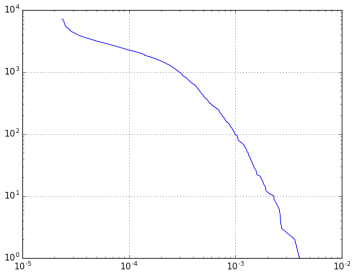
p2p-Gnutella25



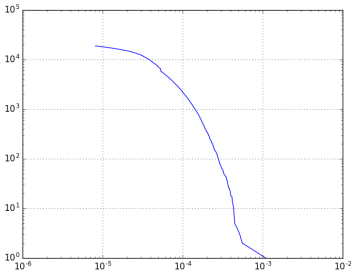
ca-GrQc



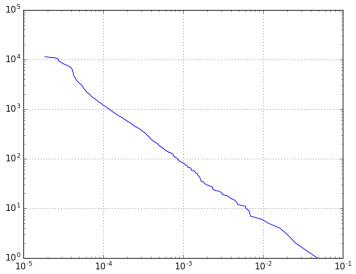
Oregon1-010331



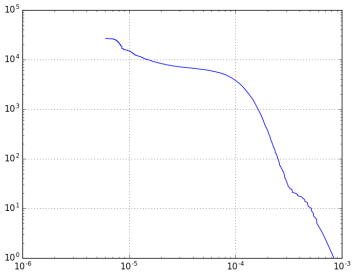
wiki-Vote



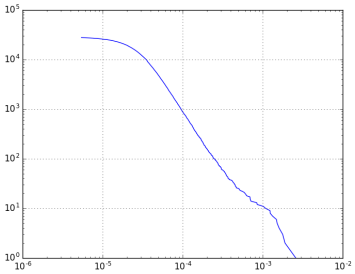
ca-Astro



Oregon1-010519



p2p-Gnutella24



cit-HepTh

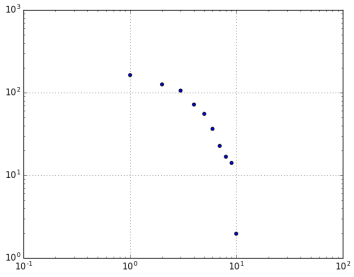
Observations

- Via CCDF plots, we see many datasets follow power law between pagerank score and counts. This implies that only a few `'famous'` users own large impacts, while most `'ordinary'` users have small pagerank score $\sim [10^{-5}, 10^{-4}]$

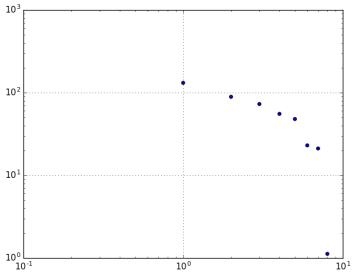
- However, for facebook dataset, the CCDF plot is not quite linear. This can be interpreted that a majority of Facebook users have low pagerank score and it is very hard to get pagerank score above a certain threshold.

6.3.3 Eigen Value Distribution

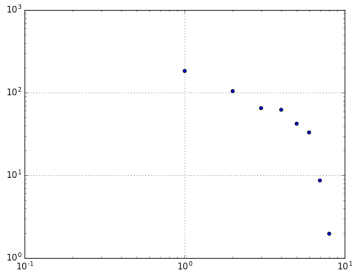
Scatter plot x-axis: rank y-axis: abs(eigen value)



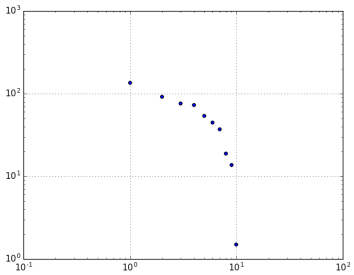
Facebook



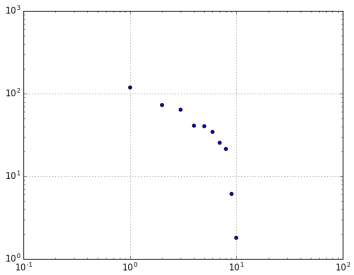
soc-Slashdot0811



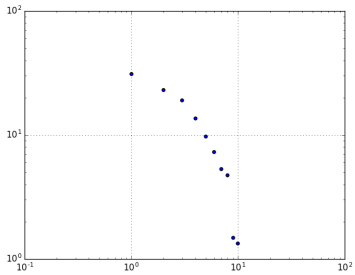
soc-Epinions1



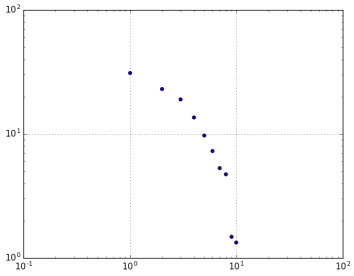
soc-Slashdot0922



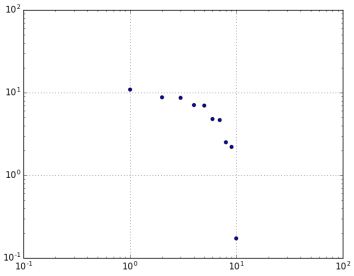
email-Enron



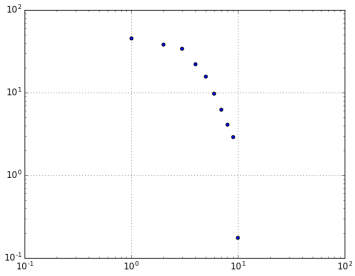
ca-HepTh



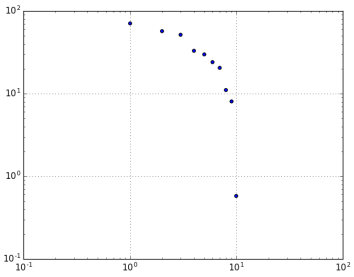
cit-HepPh



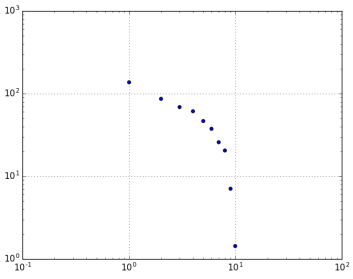
p2p-Gnutella25



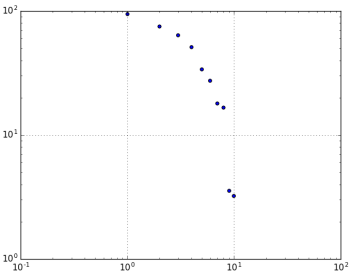
ca-GrQc



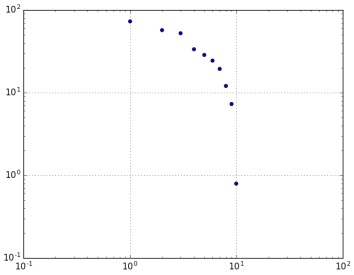
Oregon1-010331



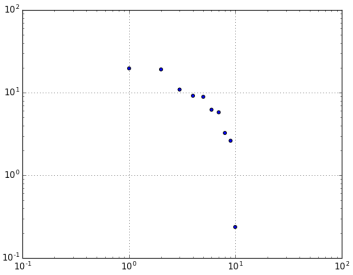
wiki-Vote



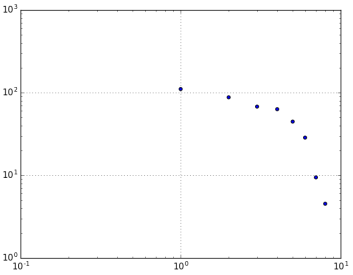
ca-Astro



Oregon1-010519



p2p-Gnutella24



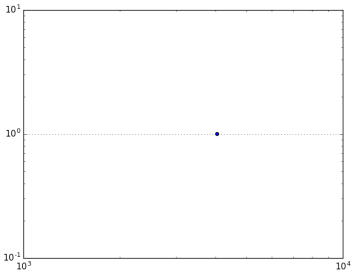
cit-HepTh

Observations

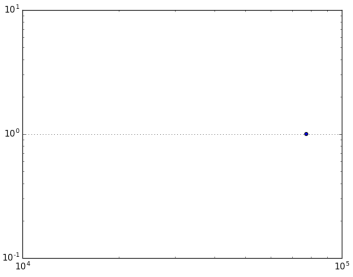
- We draw the top 10 eigen values of each dataset. It can be seen that {`eigen value` λ_i `- rank` i} follows power law for most datasets.
- Eigen value drops dramatically at rank 8, rank 9 and rank 10. It takes much longer time to converge for the calculation of these eigen values and their values are $\sim [0.1, 1]$. This implies that we can actually compress many realworld datasets with top 8 eigen values and eigen vectors without losing much information.

6.3.4 Connected-component-size distribution

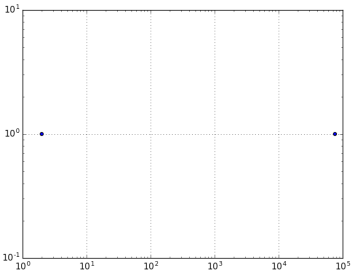
Scatter plot x-axis: component size y-axis: count



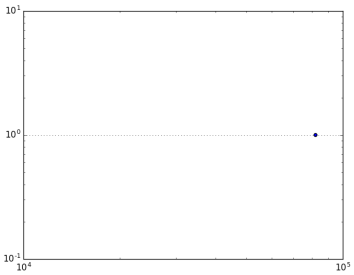
Facebook



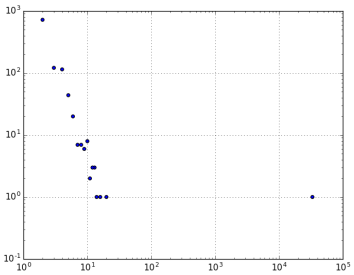
soc-Slashdot0811



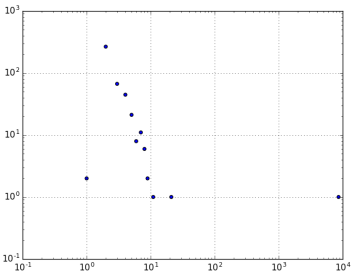
soc-Epinions1



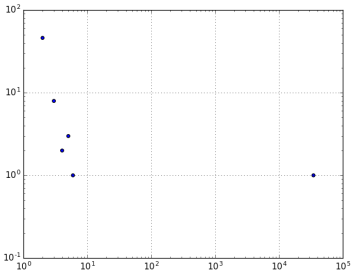
soc-Slashdot0922



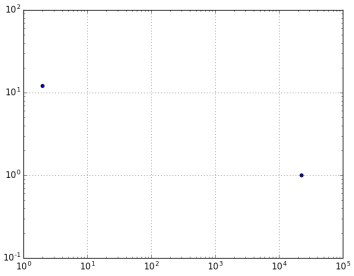
email-Enron



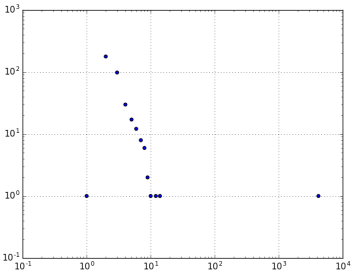
ca-HepTh



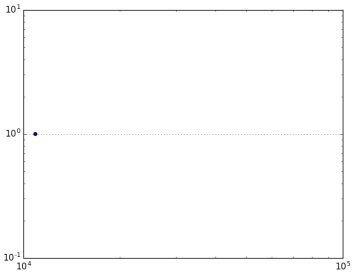
cit-HepPh



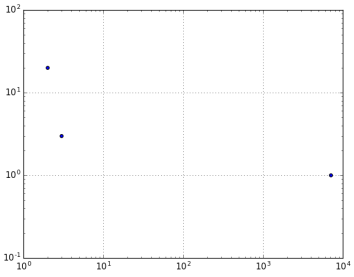
p2p-Gnutella25



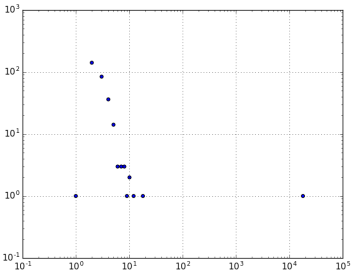
ca-GrQc



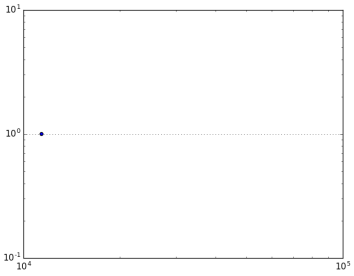
Oregon1-010331



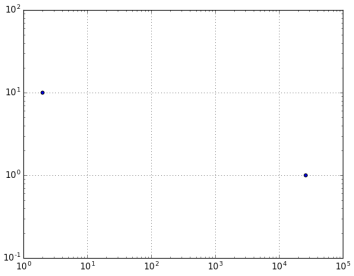
wiki-Vote



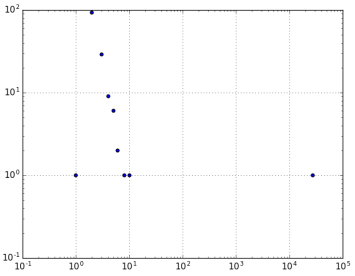
ca-Astro



Oregon1-010519



p2p-Gnutella24



cit-HepTh

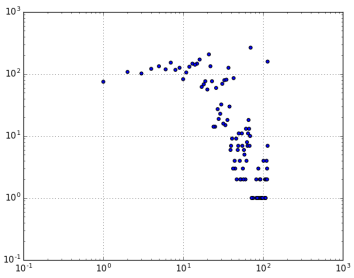
Observations

- **Social networks** (Facebook, soc-Slashdot, soc-Epinions), **P2P networks** (p2p-Gnutella) and **Autonomous networks** (Oregon1) are very well and fully connected, resulting in only one or two connected components.

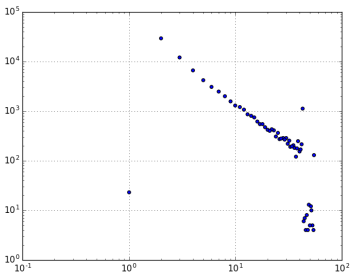
- Other networks such as **Collaboration networks** and **Citation networks** exhibits a power law relation for components of small & medium sizes. For these networks, there exists one large component whose size is greater than sum of the rest.

6.3.5 Coreness Value Distribution

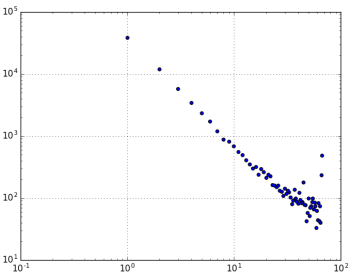
Scatter plot x-axis: k-core value y-axis: count



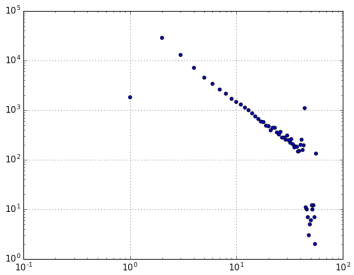
Facebook



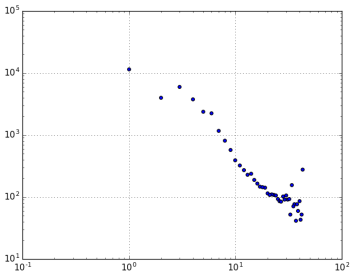
soc-Slashdot0811



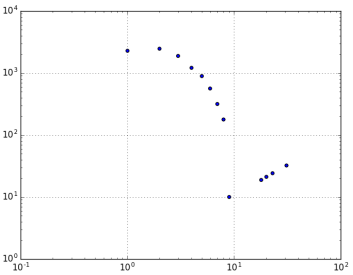
soc-Epinions1



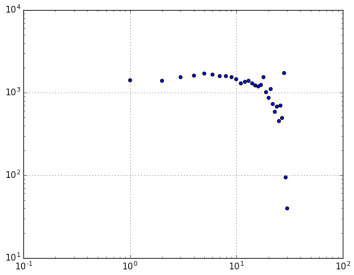
soc-Slashdot0922



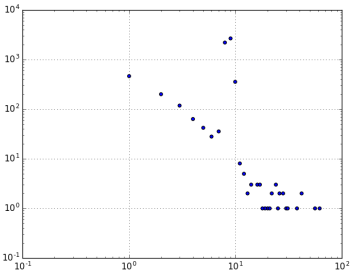
email-Enron



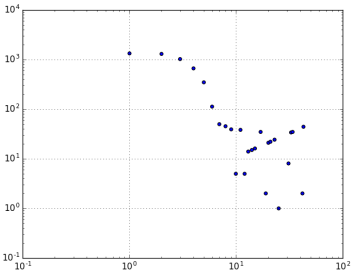
ca-HepTh



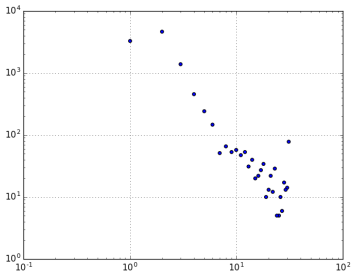
cit-HepPh



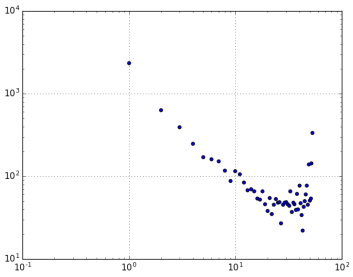
p2p-Gnutella25



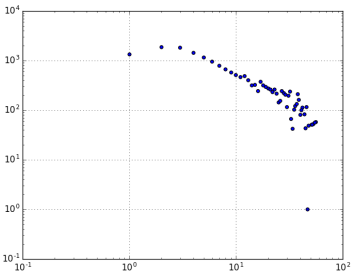
ca-GrQc



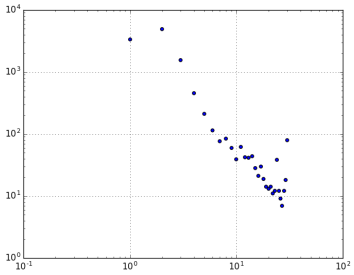
Oregon1-010331



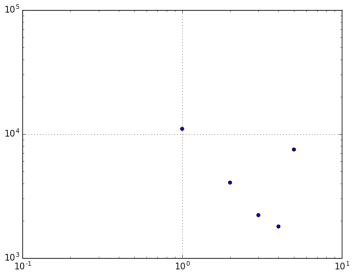
wiki-Vote



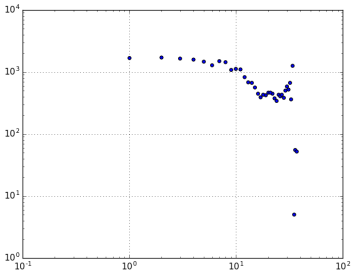
ca-Astro



Oregon1-010519



p2p-Gnutella24



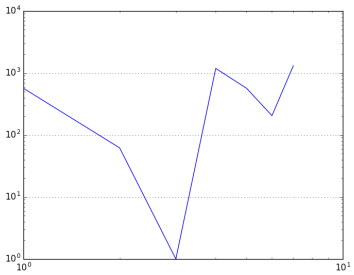
cit-HepTh

Observations

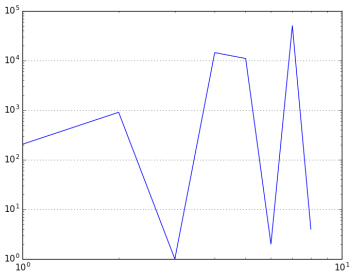
- Most datasets follow power law between `kcore value - count`.
- It is easy to spot some outliers in dataset soc-Slashdot. soc-Slashdot graph has several dozen nodes whose k-core value are 1, loosely connected with other nodes.

6.3.6 Radius Distribution

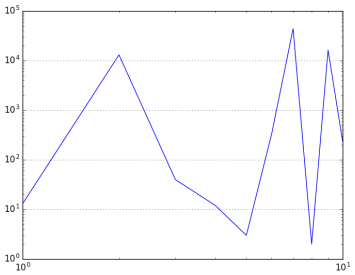
PDF x-axis: raidus y-axis: count



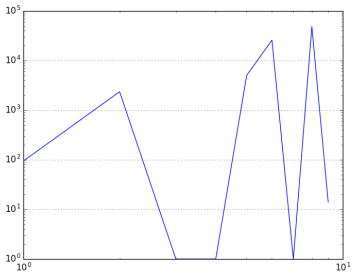
Facebook



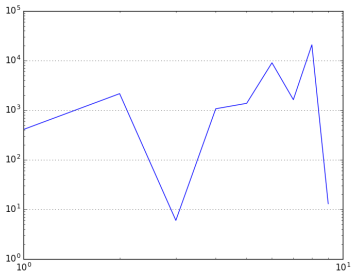
soc-Slashdot0811



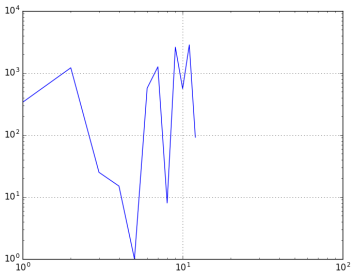
soc-Epinions1



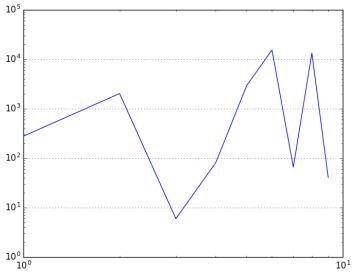
soc-Slashdot0922



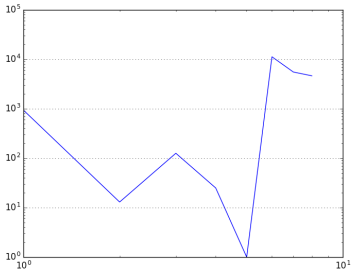
email-Enron



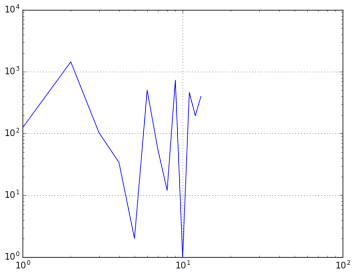
ca-HepTh



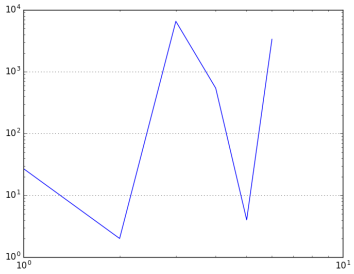
cit-HepPh



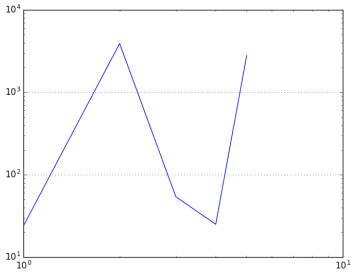
p2p-Gnutella25



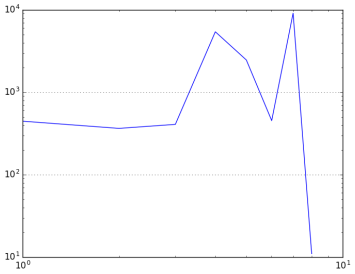
ca-GrQc



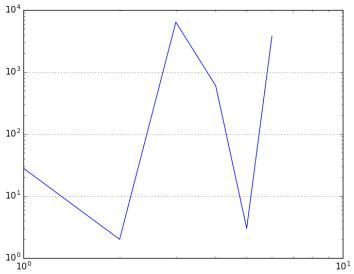
Oregon1-010331



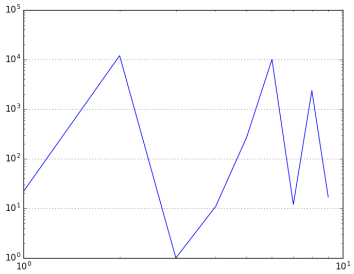
wiki-Vote



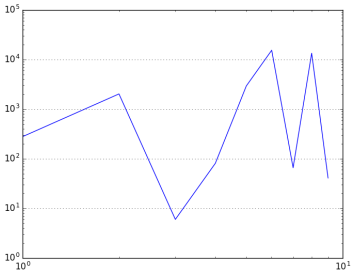
ca-Astro



Oregon1-010519



p2p-Gnutella24



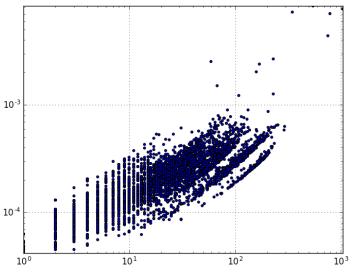
cit-HepTh

Observations

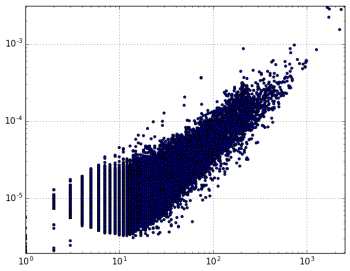
- Realworld graphs have small diameters. Most nodes follow **six degree principle** (their effective radius are 5-7).
- We see that radius distribution (in PDF graph) is multimodal.

6.3.7 Degree VS PageRank Distribution

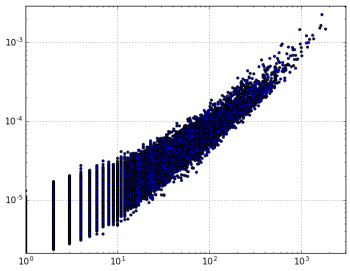
Scatter plot a-axis: degree y-axis: pagerank score



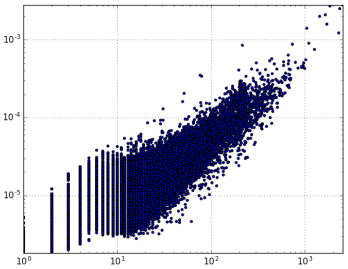
Facebook



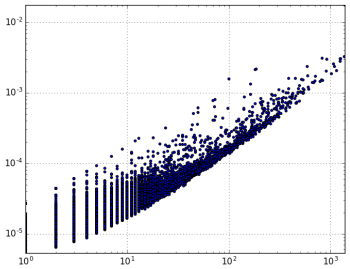
soc-Slashdot0811



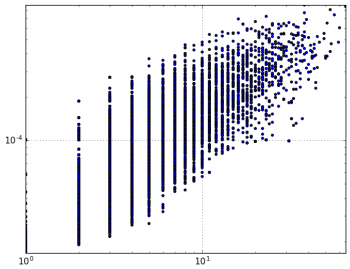
soc-Epinions1



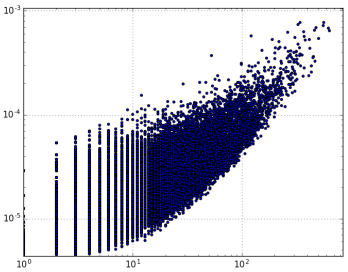
soc-Slashdot0922



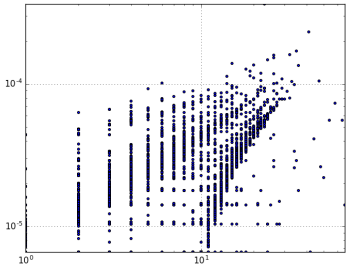
email-Enron



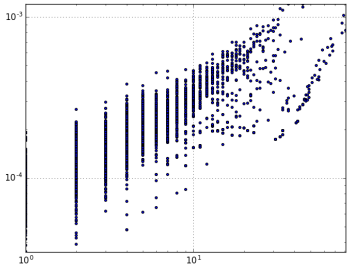
ca-HepTh



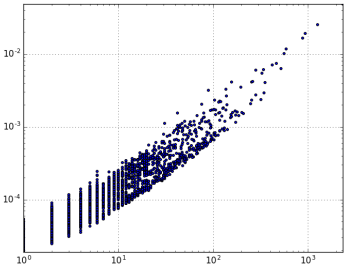
cit-HepPh



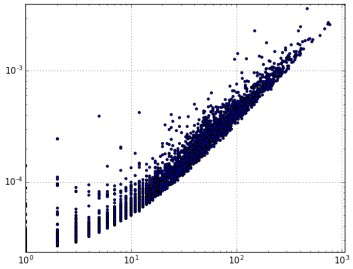
p2p-Gnutella25



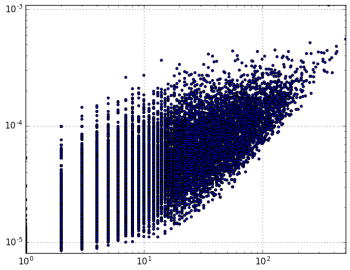
ca-GrQc



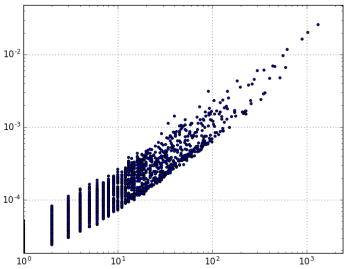
Oregon1-010331



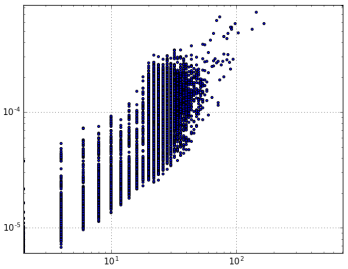
wiki-Vote



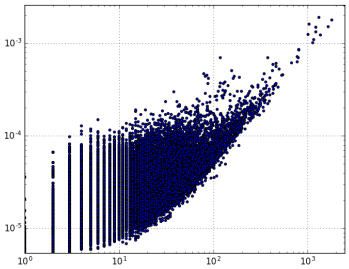
ca-Astro



Oregon1-010519 Observations



p2p-Gnutella24

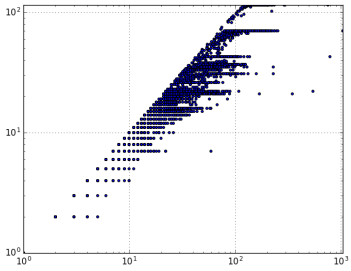


cit-HepTh

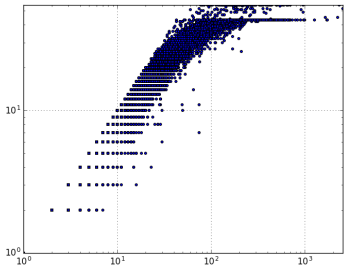
- In log-log scale, degree and pagerank score are positively correlated and approximately growing linearly.
- **P2P networks** and **collaboration networks** are much more sparse and scattered compared to others.
- Some datasets e.g.`cit-HepPh`, we can observe that pagerank score grows super-linearly with degree. This means that growing connections can bring in even bigger impacts.

6.3.8 Degree VS Coreness Distribution

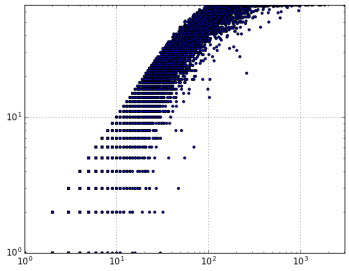
Scatter plot x-axis: degree y-axis: k-core value



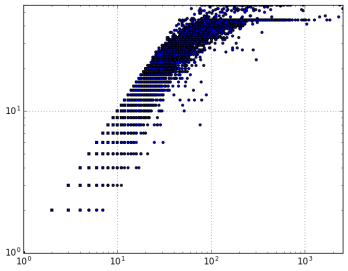
Facebook



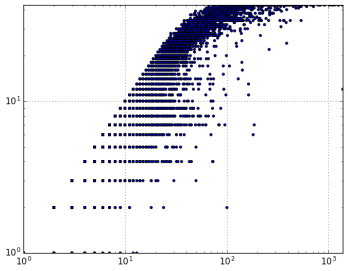
soc-Slashdot0811



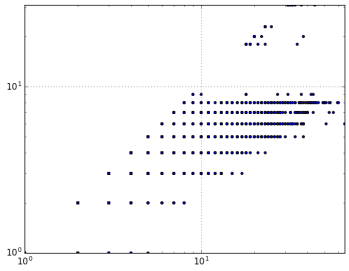
soc-Epinions1



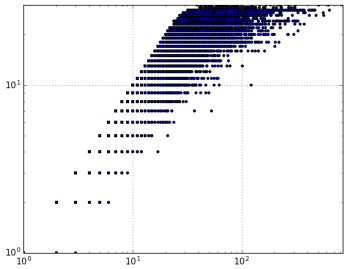
soc-Slashdot0922



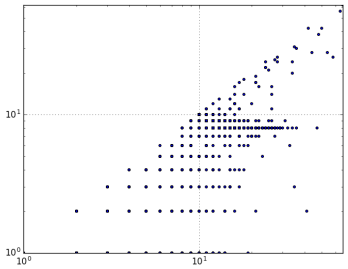
email-Enron



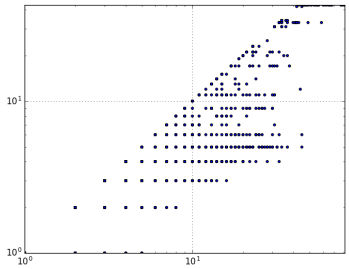
ca-HepTh



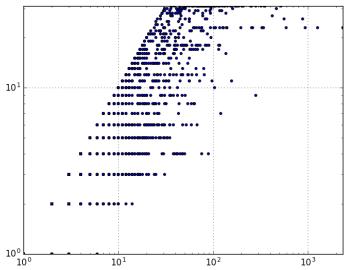
cit-HepPh



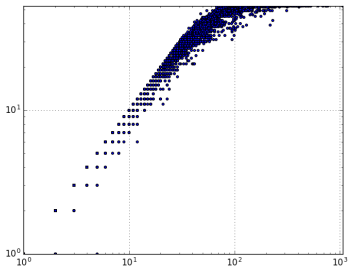
p2p-Gnutella25



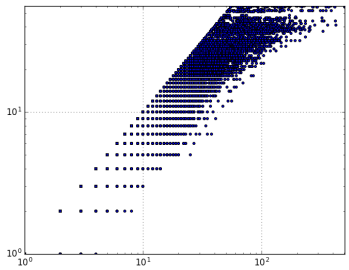
ca-GrQc



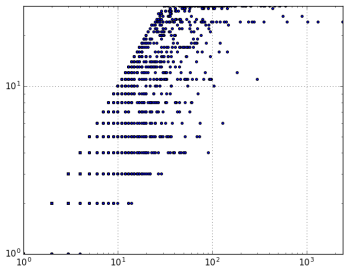
Oregon1-010331



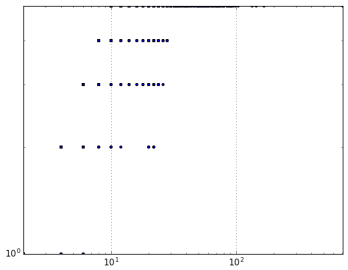
wiki-Vote



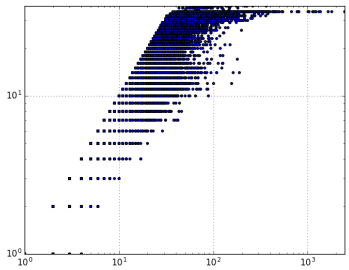
ca-Astro



Oregon1-010519



p2p-Gnutella24



cit-HepTh

Observations

- Like previous sections, degree VS coreness follows the same trend. Node's degree and kcore value are positively correlated and growing linearly.
- **P2P networks** and **collaboration networks** are not as dense as others. For such networks, kcore values and degree values are much more limited in a small range.

6.4 Conclusions

We have provided over 15 observations in the above sections. In general, we see find most graphs follow the power law either in scatter / PDF / CCDF form, i.e. a majority of nodes have low degrees / pagerank scores / k-core values. We also spotted some anomolies from plots.

7 Labor Division

- Yuwei Zhang: Degree distribution, Eigen Value distribution, Radius Distribution, Degree VS PageRank Distribution
- Silun Wang: Pagerank Distribution, Connected-component-size Distribution, Coreness Value Distribution, Degree VS Coreness Distribution

8 Acknowledgement

We would like to thank Nijith Jacob and Sharif Doghmi for their previous work on Graph Miner toolset. We also acknowledge SNAP for the dataset.

References

[1] https://msdn.microsoft.com/en-us/library/ms190457.aspx.

[2] Ignacio Alvarez-Hamelin, Luca DallAsta, Alain Barrat, and Alessandro Vespignani. k-core decomposition: a tool for the visualization of large scale networks. In *arXiv.org*, 2005.

[3] Cody Dunne and Ben Shneiderman. Motif simplification: Improving network visualization readability with fan, connector, and clique glyphs. In *CHI*, 2013.

[4] Wolfgang Gatterbauer, Stephan Gunnemann, Danai Koutra, and Christos Faloutsos. Linearized and single-pass belief propagation. In *VLDB*, pages 581 – 592, 2015.

[5] Christos Giatsidis, Klaus Berberich, Dimitrios M Thilikos, and Michalis Vazirgiannis. Visual exploration of collaboration networks based on graph degeneracy. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1512–1515. ACM, 2012.

[6] Christos Giatsidis, Dimitrios M Thilikos, and Michalis Vazirgiannis. Evaluating cooperation in communities with the k-core structure. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 87–93. IEEE, 2011.

[7] Alekh Jindal, Praynaa Rawlani, Eugene Wu, Samuel Madden, Amol Deshpande, and Mike Stonebraker. Vertexica: your relational friend for graph analytics! *Proceedings of the VLDB Endowment*, 7(13):1669–1672, 2014.