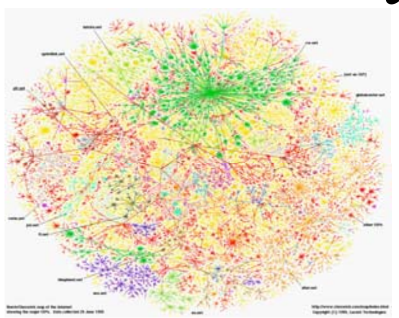
Big Graph Mining: Theory, Engineering, and Discoveries

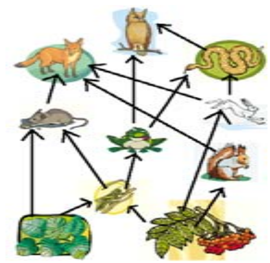
U Kang
Dept. of Computer Science
KAIST

Motivation

- Graphs are everywhere.

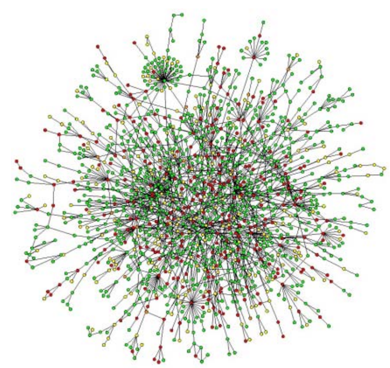


Internet Map
[cheswick.com]



Food Web
[biologycorner.com]

Friendship Network
[fmsag.com]

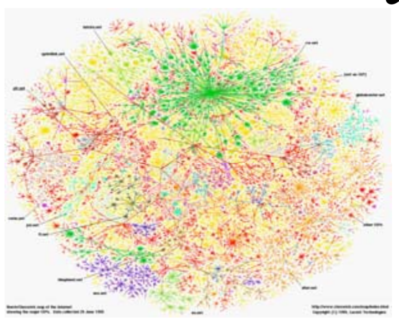


Protein Interactions
[bordalierinstitute.com]

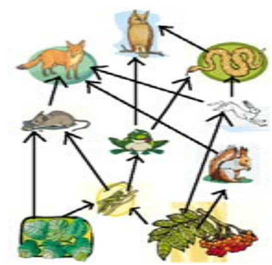
• • •

Motivation

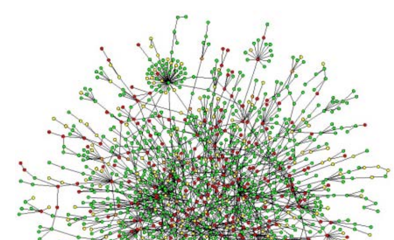
- Graphs are everywhere.



Internet Map
[cheswick.com]



Food Web
[biologycorner.com]



. . .

Goal 1: Find Patterns and Anomalies

Communities, diameter, important nodes, etc.

Motivation

- The sizes of graphs are growing!

facebook

0.5 billion users
60 TBytes/day
15 PBytes/total
[Thusoo+ '10]

bing

ClickStream Data
0.26 PBytes
1 billion query-URL
[Liu+ '09]

YAHOO!

1.4 billion web pages
6.6 billion edges
[Broder+ '04]

Google

20 PBytes/day **[processed]**

[Dean+ '08]

Motivation

- The sizes of graphs are growing!

facebook

0.5 billion users
60 TBytes/day
15 PBytes/total
[Thusoo+ '10]

bing

ClickStream Data
0.26 PBytes
1 billion query-URL
[Liu+ '09]

YAHOO!

Google

Goal 2: Scale-up

For graphs with **billions** of nodes and edges

Goal

- PEGASUS: Peta-Scale Graph Mining System
  - Scalable algorithms for mining very large graphs
    - Pagerank, Random Walk with Restart
    - Connected Component
    - Radius
    - Belief Propagation
    - Eigensolver
    - …

Data

- Real and synthetic graphs

Graph	Nodes	Edges	File Size
YahooWeb	1.4 B	6.6 B	0.12 TB
Twitter	104 M	3.7 B	80 GB
LinkedIn	7.5 M	58 M	1 GB
U.S. Patent	6 M	16 M	264 MB
Wikipedia	3.5 M	42 M	600 MB
Kronecker	177 K	1,977 M	25 GB
Erdos-Renyi	177 K	1,977 M	25 GB

Overview

Task	Discoveries	Algorithm
Structure of Large Graphs	Q1: What do large networks look like?	Q2: How to scale-up structure analysis algorithm?
Eigensolver	Q3: How to spot strange behaviors in networks?	Q4: How to design a billion-scale eigensolver?
Tensor Decomposition	Q5: What are the important concepts and synonyms in a KB tensor?	Q6: How to decompose a billion-scale tensor?

Outline

Task	Discoveries	Algorithm
Structure of Large Graphs	Q1: What do large networks look like?	Q2: How to scale-up structure analysis algorithm?
Eigensolver	Q3: How to spot strange behaviors in networks?	Q4: How to design a billion-scale eigensolver?
Tensor Decomposition	Q5: What are the important concepts and synonyms in a KB tensor?	Q6: How to decompose a billion-scale tensor?

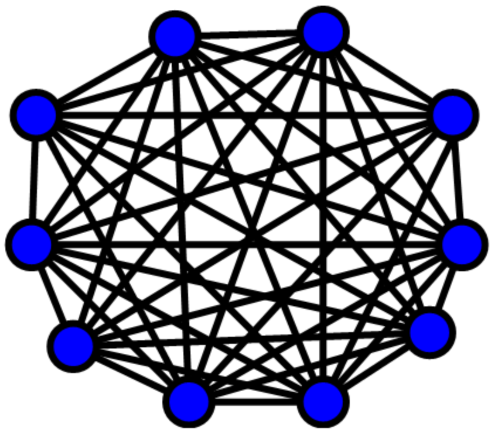
- ☑ **Motivation**
- ☐ Structure of Large Graphs
  - D1. Radius Plots
  - A1. GIM-V
- ☐ Eigensolver
- ☐ Tensor Decomposition
- ☐ Conclusions

Problem Definition

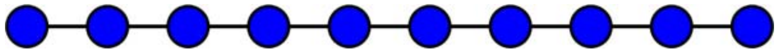
- Q1: What do large networks look like?
  - Q1.1: What is the structure of large networks?
  - Q1.2: Node centrality: which node is the most central?
  - Q1.3: How does the structure of networks change over time?

[Kang et al. SDM'10]

Q1.1: Structure of Large Networks



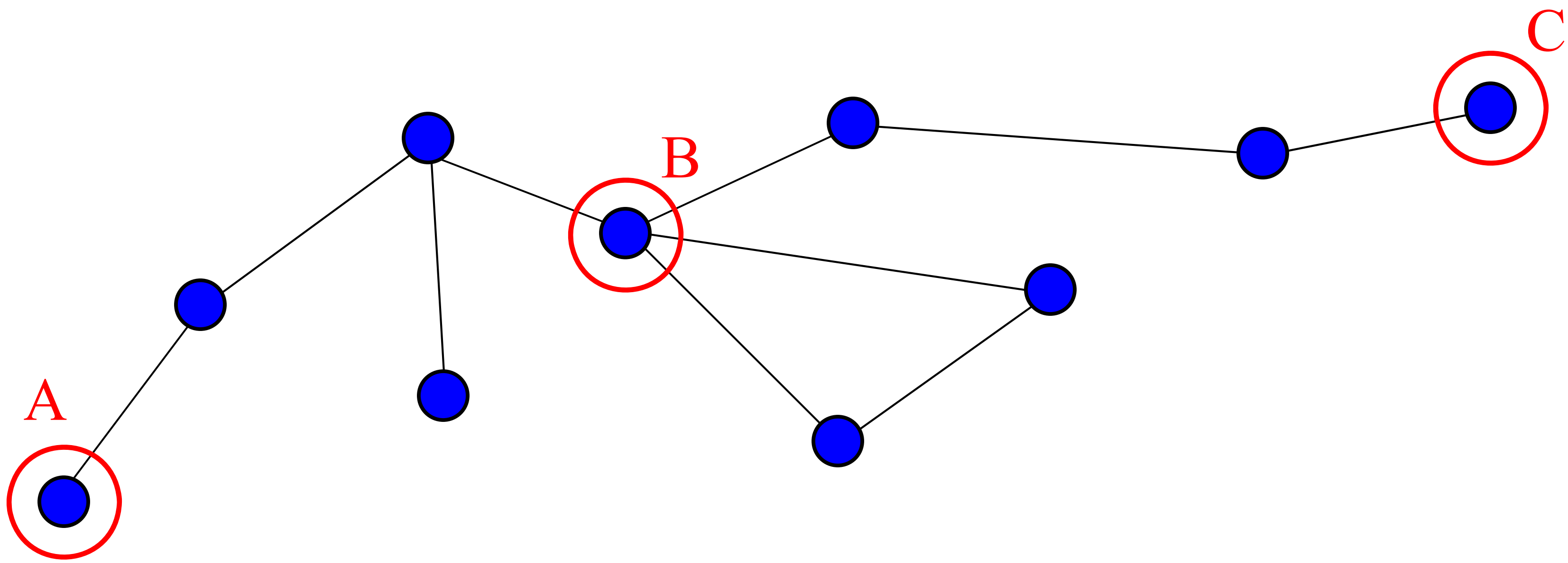
Clique?



Chain?

Q: Can we have a concise summary of the structure of networks?

Q1.2: Node (closeness) centrality



Q: If you have to pick 1 person to advertise, who do you want to choose?

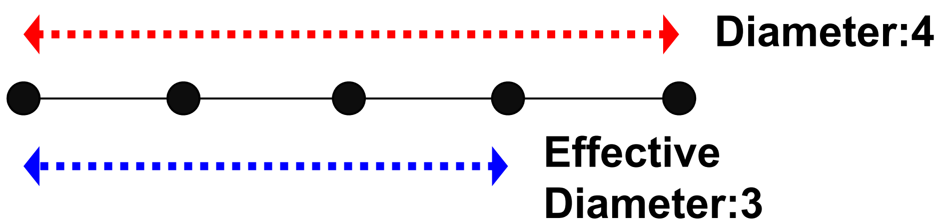
[Kang et al. SDM'10]

Q1.3: Evolution of networks

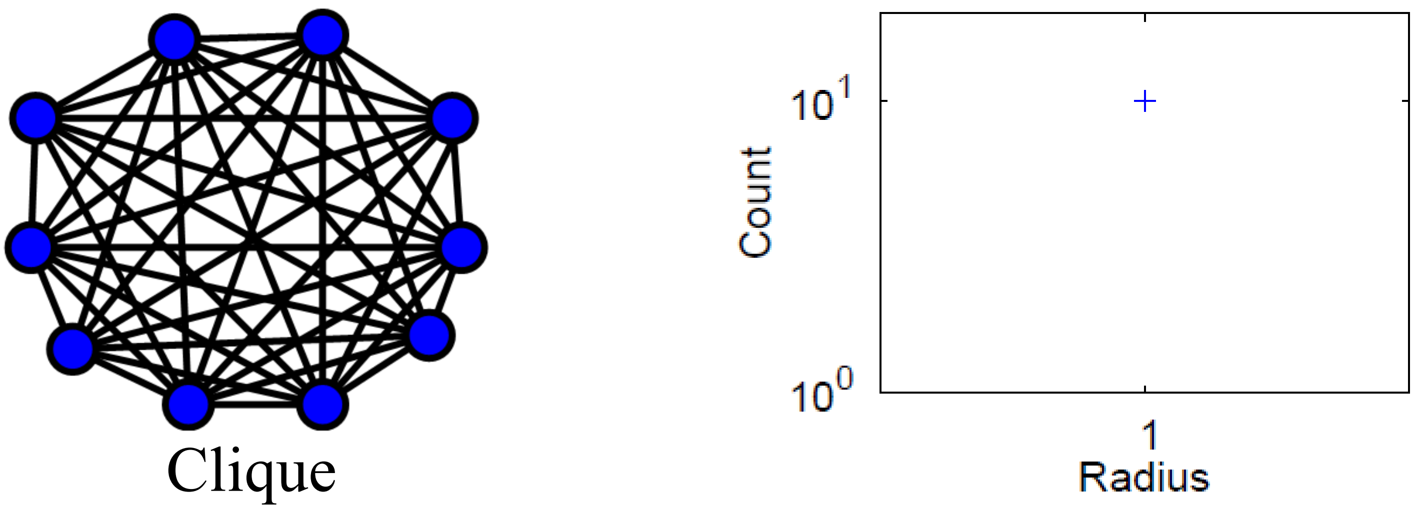
- How does the structure of networks change over time?

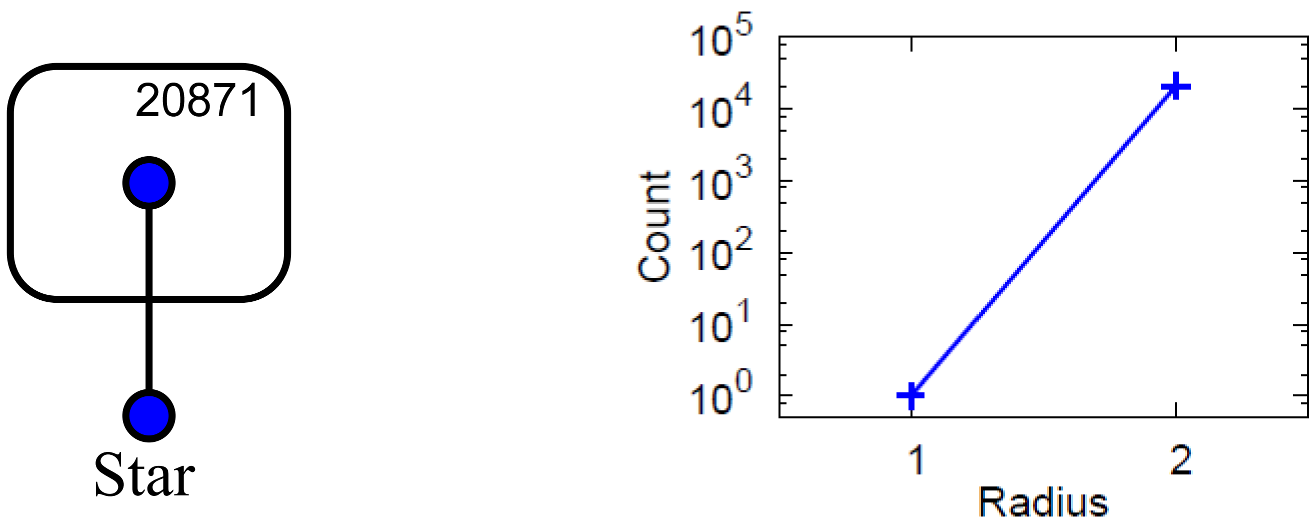
Answer: Radius Plot!

- Radius of a node: the longest shortest distance to all other nodes
- Effective radius of a node: 90^{th}-percentile of the radius
- Diameter of a graph: maximum radius
- Effective Diameter of a graph: the number of hops 90% of all pair of nodes can be reached

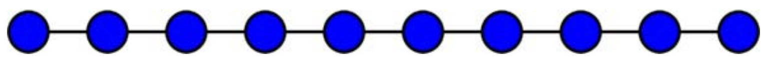


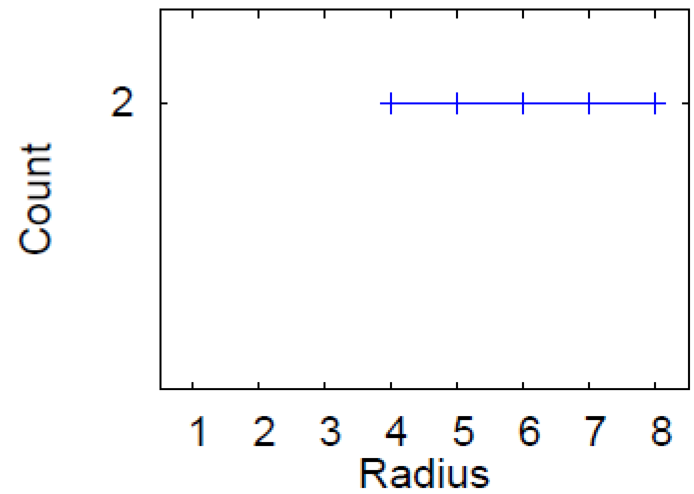
Radius Plot


Clique
Count
Radius
1
10^0
10^1

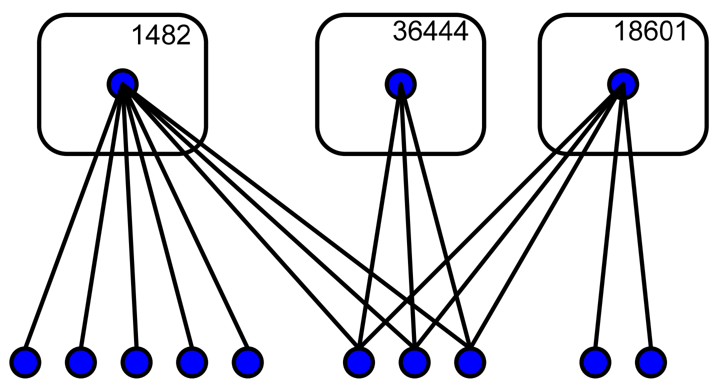

20871
Star
Count
Radius
1
2
10^0
10^1
10^2
10^3
10^4
10^5

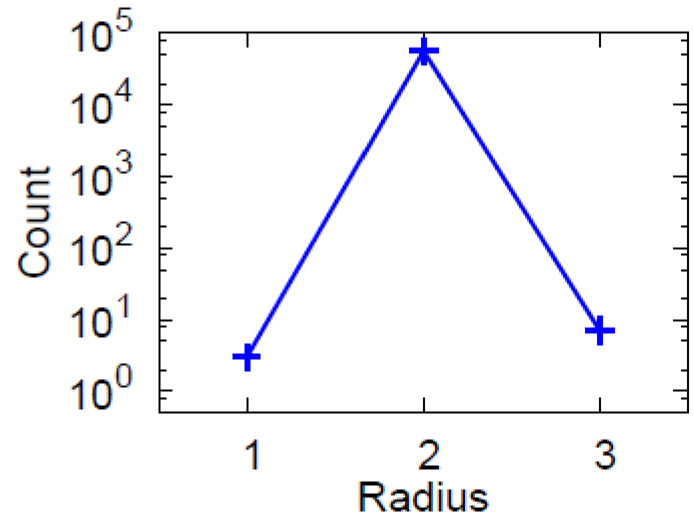
Radius Plot





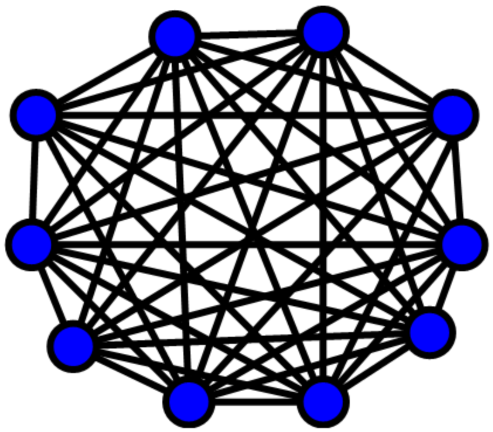
Chain



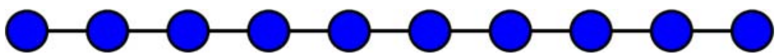


Near-bipartite-core

Q1.1: Structure of Large Networks



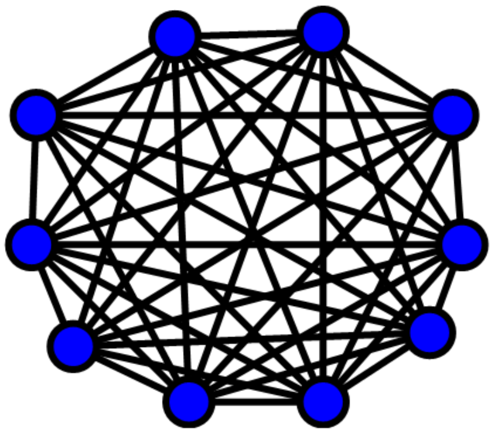
Clique?

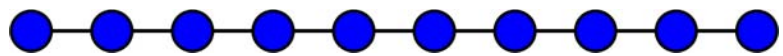


Chain?

Q: Can we have a concise summary of the structure of networks?

Q1.1: Structure of Large Networks

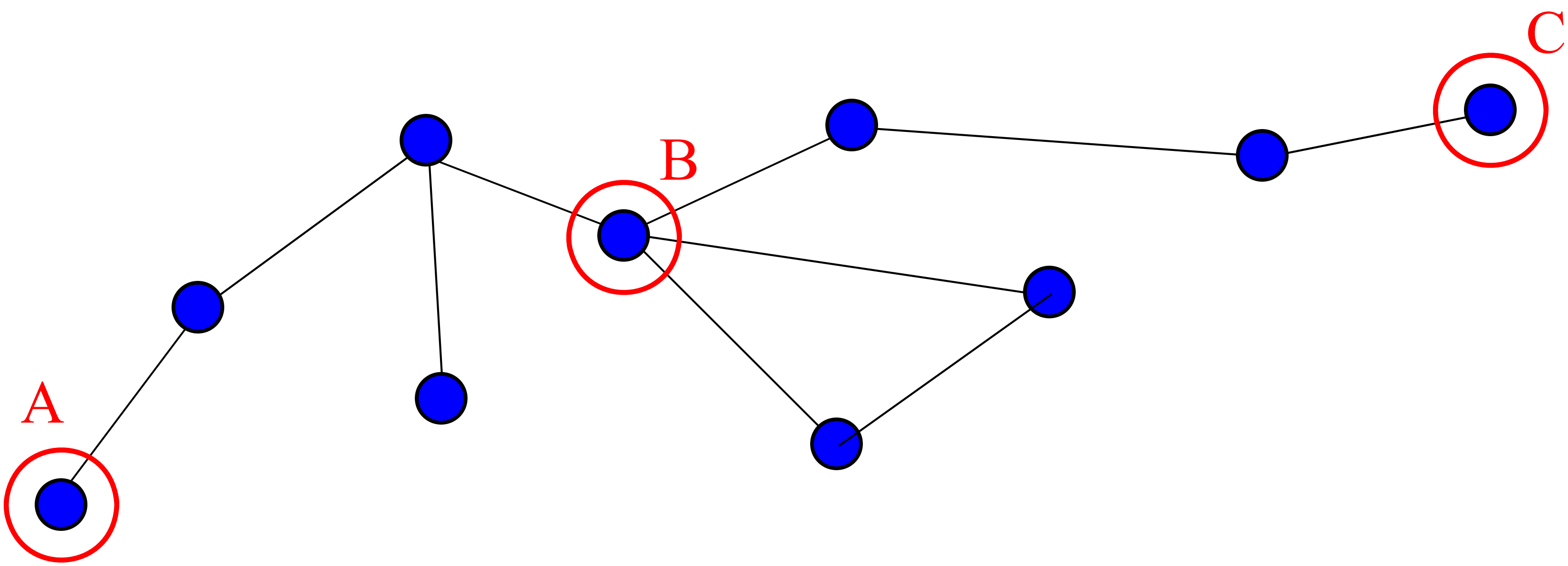




A: Radius plot gives an answer

Q: Can we have a concise summary of the structure of networks?

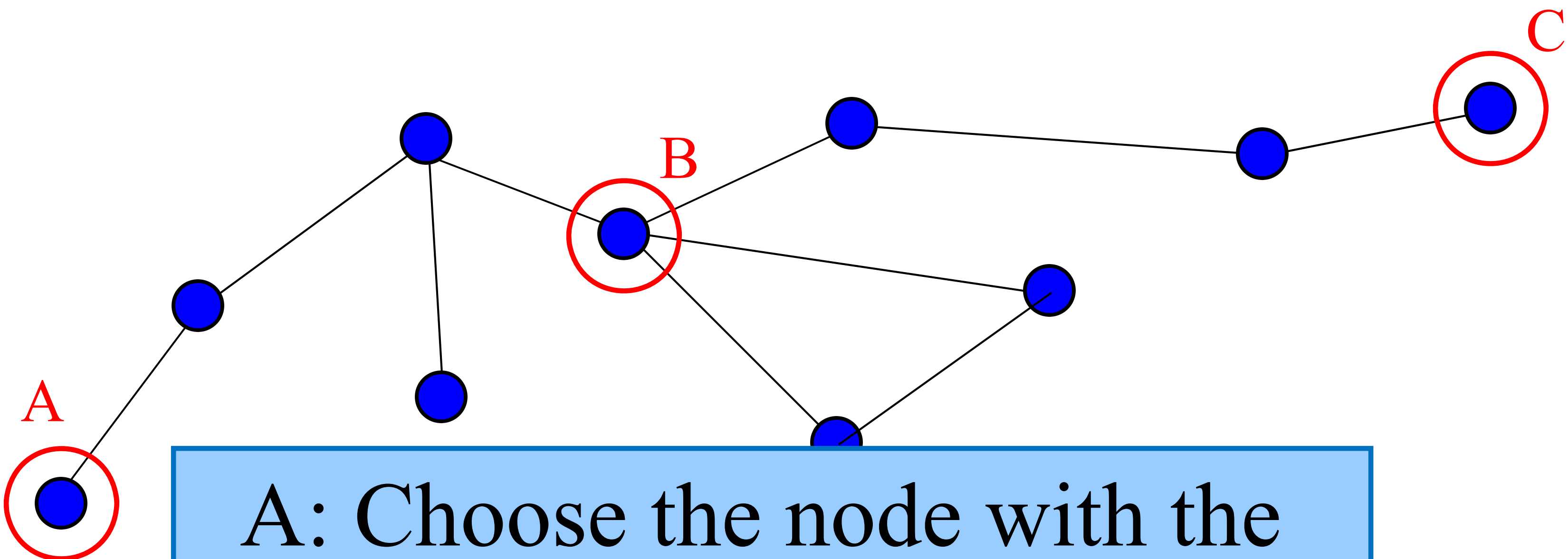
Q1.2: Node (closeness) centrality



Q: If you have to pick 1 person to advertise, who do you want to choose?

[Kang et al. SDM'10]

Q1.2: Node (closeness) centrality



A: Choose the node with the minimum radius!

Q: If you have to pick 1 person to advertise, who do you want to choose?

[Kang et al. SDM'10]

Q1.3: Evolution of networks

- How does the structure of networks change over time?

A: Study Radius plot over time!

[Kang et al. SDM’10]

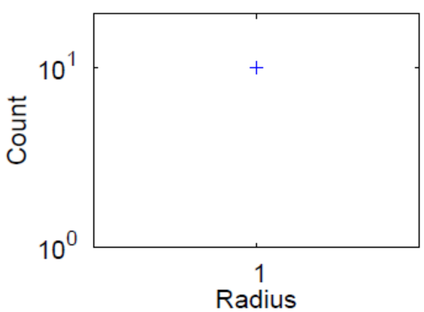
A1.1: Radius plot of real graphs

- LinkedIn: |V|=7.5M, |E|=58M, 1GBytes
- U.S. Patent: |V|=6M, |E|=16M, 264 MBytes

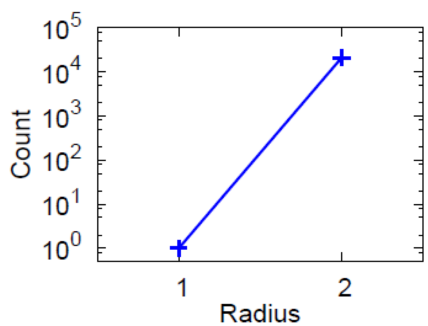
A1.1: Radius plot of real graphs

- LinkedIn: |V|=7.5M, |E|=58M, 1GBytes
- U.S. Patent: |V|=6M, |E|=16M, 264 MBytes

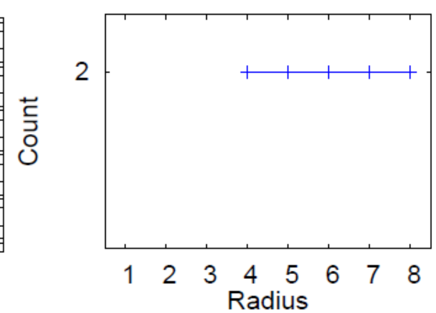
Q: What do the real graphs look like?



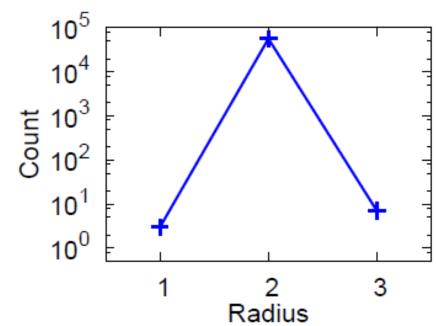
Clique?



Star?



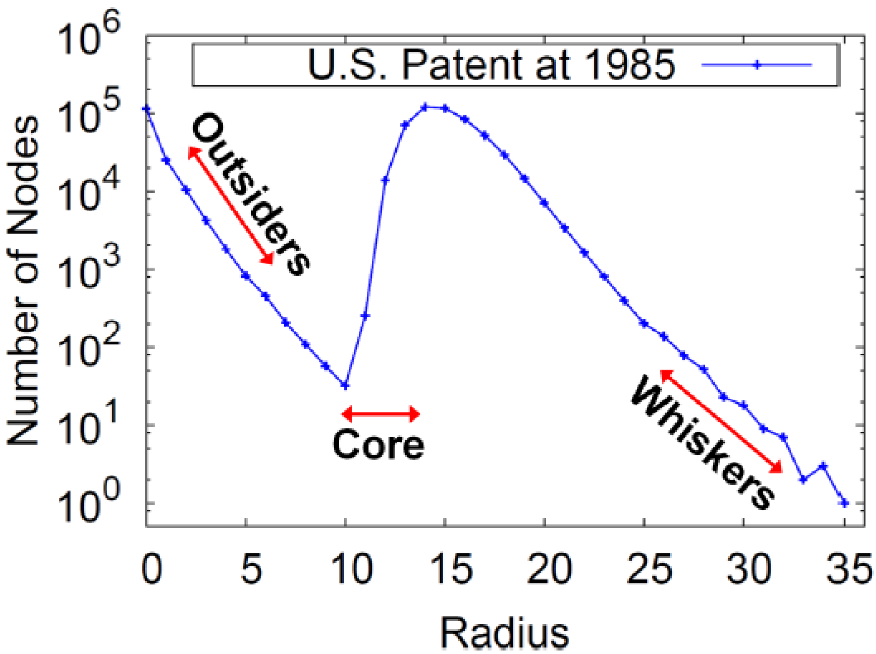
Chain?

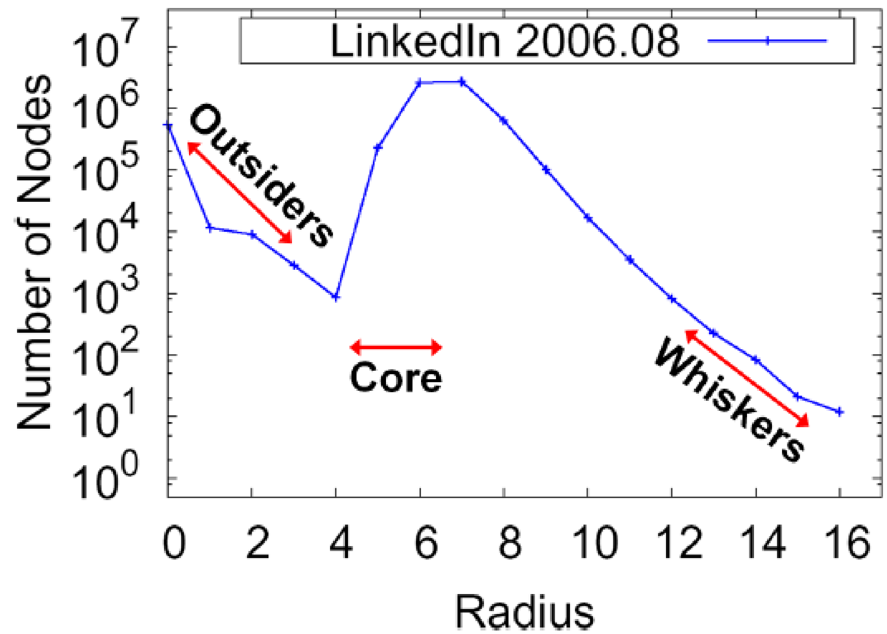


Bipartite Core?

A1.1: Radius plot of real graphs

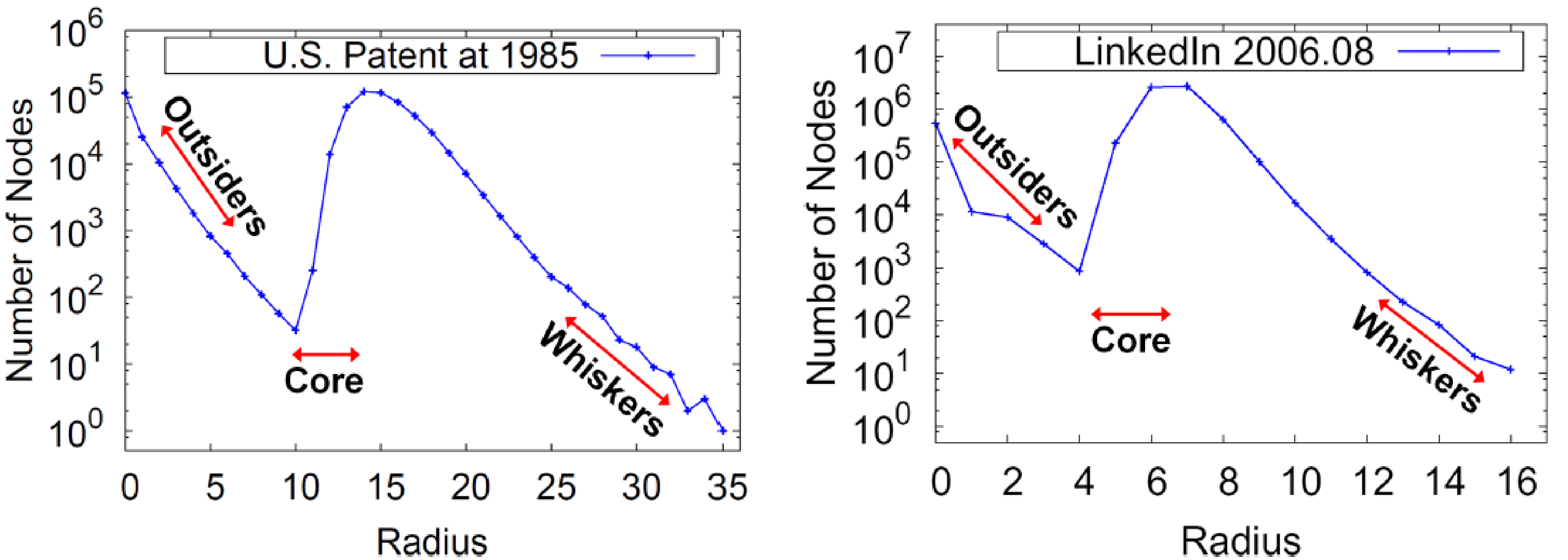
A: Bi-modal!





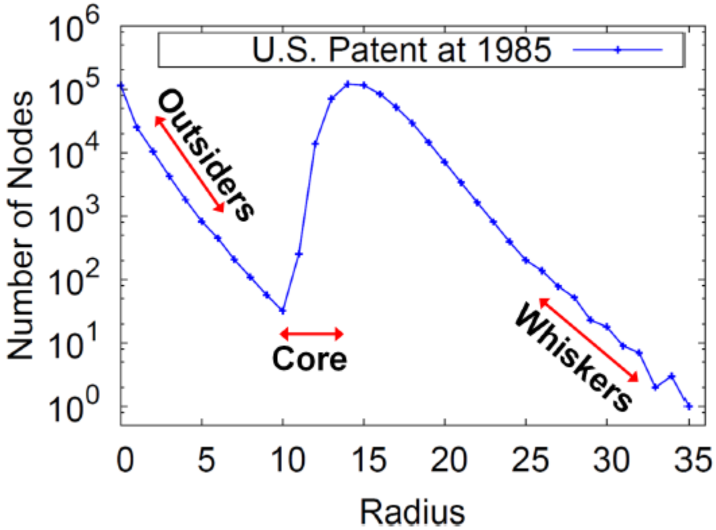
A1.1: Radius plot of real graphs

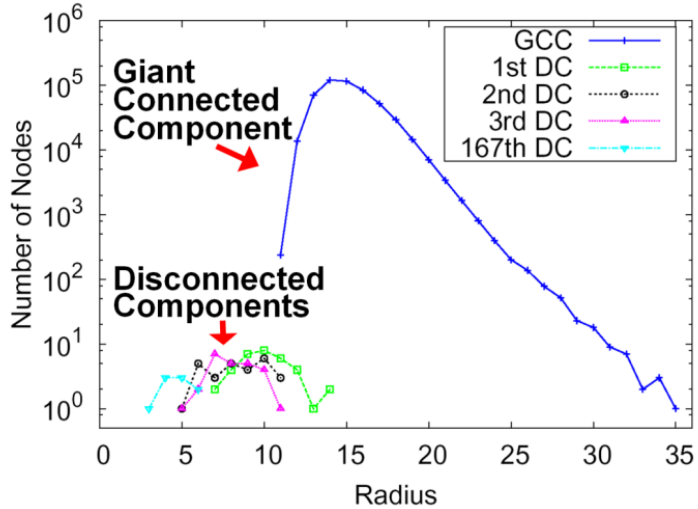
A: Bi-modal!

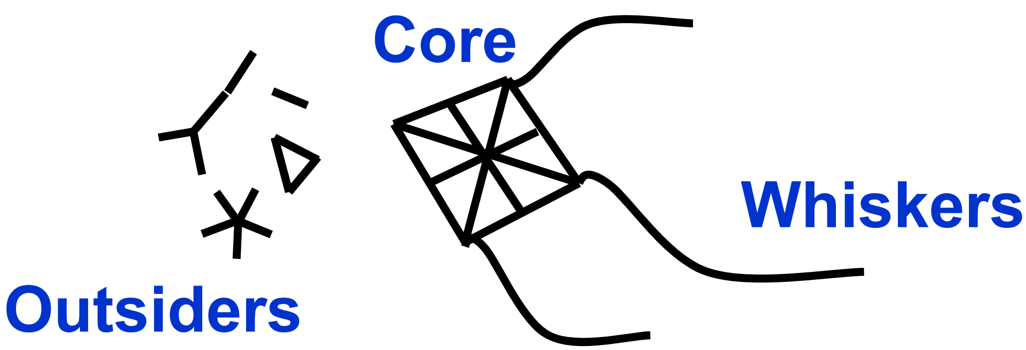


Q: What is the reason for this bi-modality?

A1.1: Radius plot of real graphs







A1.1: Radius plot of YahooWeb

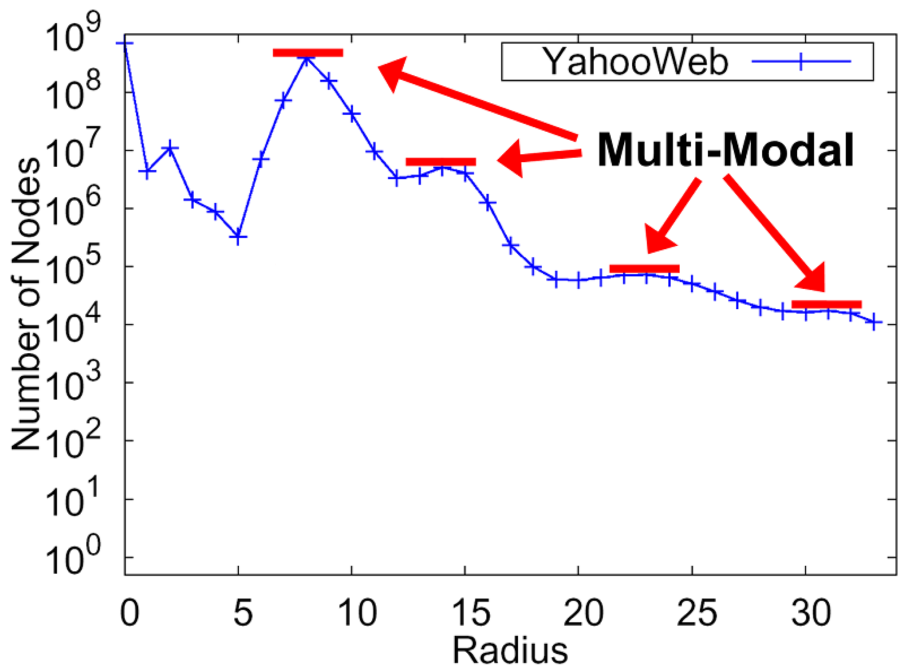
- YahooWeb: |V|=1.4B, |E|=6.6B, 120GBytes

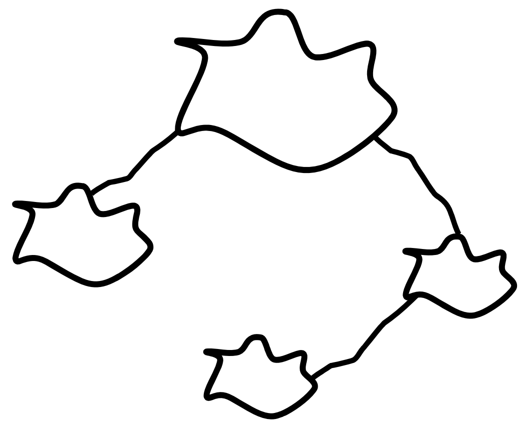
Q: How about the radius plot of a much larger graph? Also bi-modal?

[Kang et al. SDM'10]

A1.1: Radius plot of YahooWeb

A: Multi-modality!





- Multi-modality possibly from mixture of cores

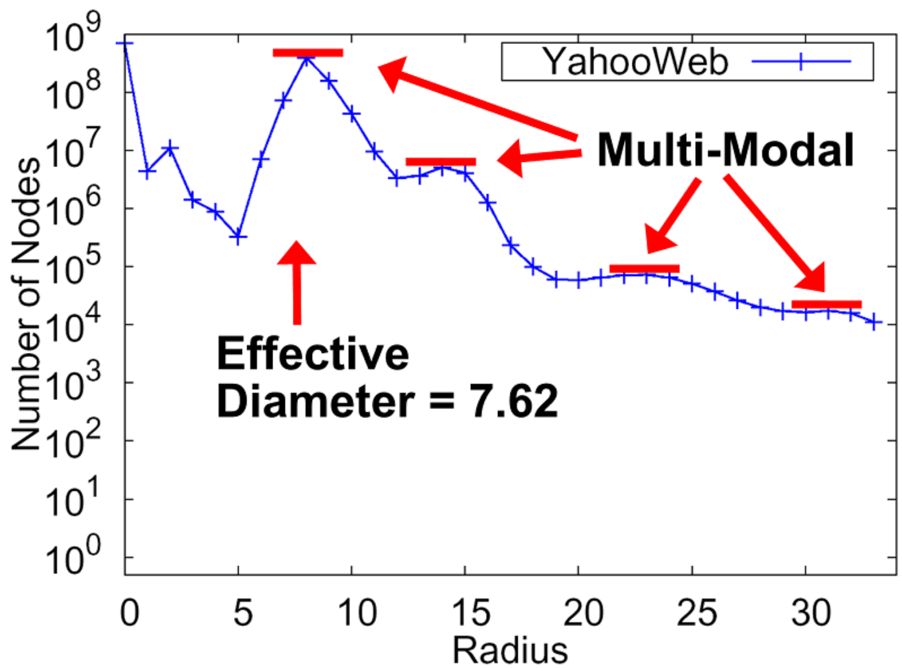
A1.1: Radius plot of YahooWeb

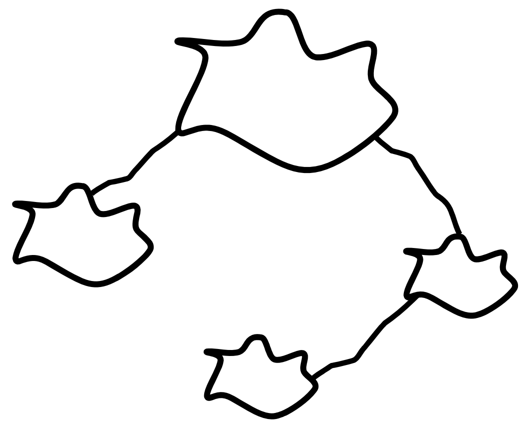
- YahooWeb: |V|=1.4B, |E|=6.6B, 120GBytes

Q: What is the diameter of the Web?

A1.1: Radius plot of YahooWeb

A: 7 degrees of separation!





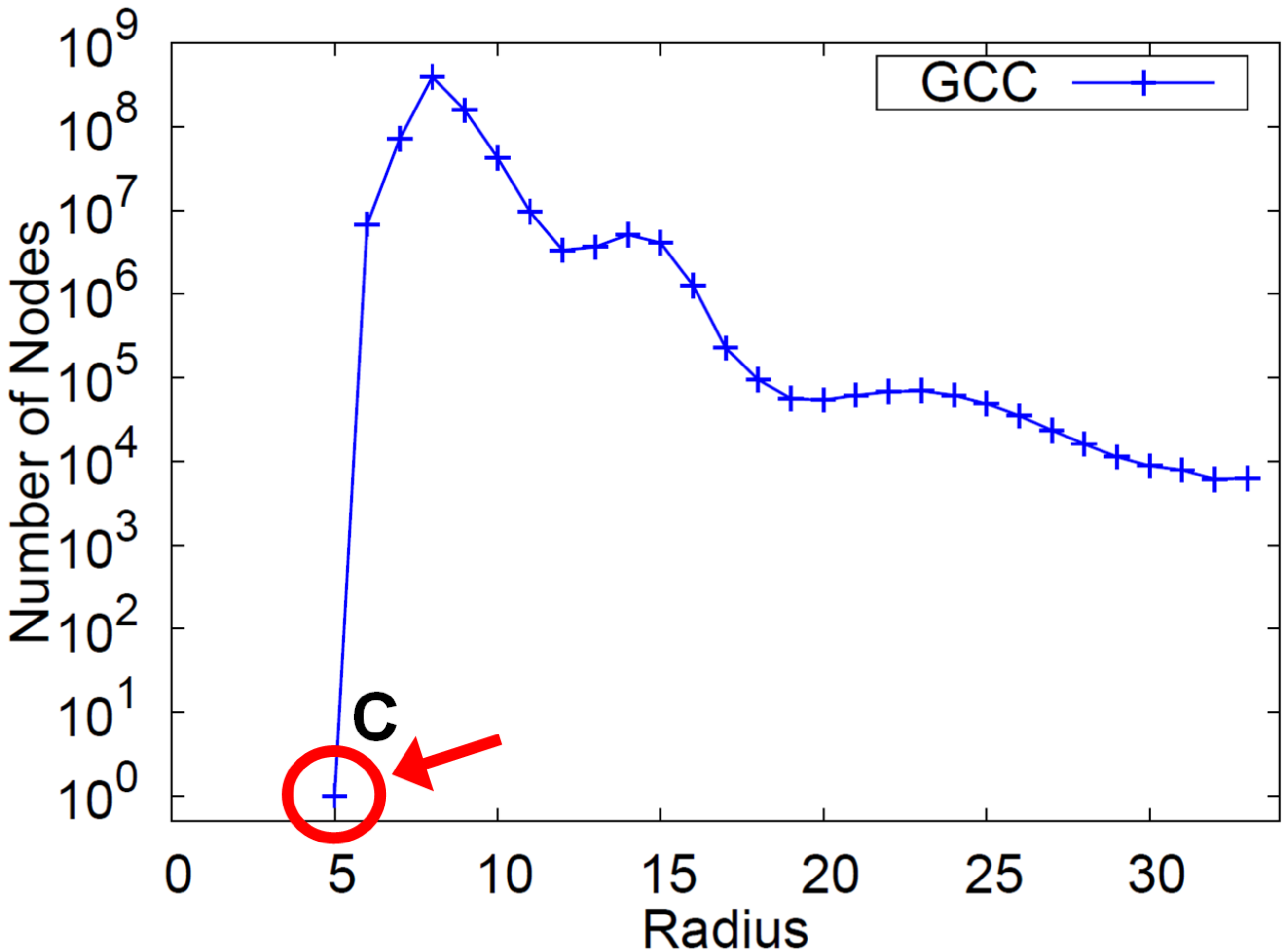
- Multi-modality possibly from mixture of cores
- Effective diameter: surprisingly small

A1.2: Node (closeness) centrality

- YahooWeb: |V|=1.4B, |E|=6.6B, 120GBytes

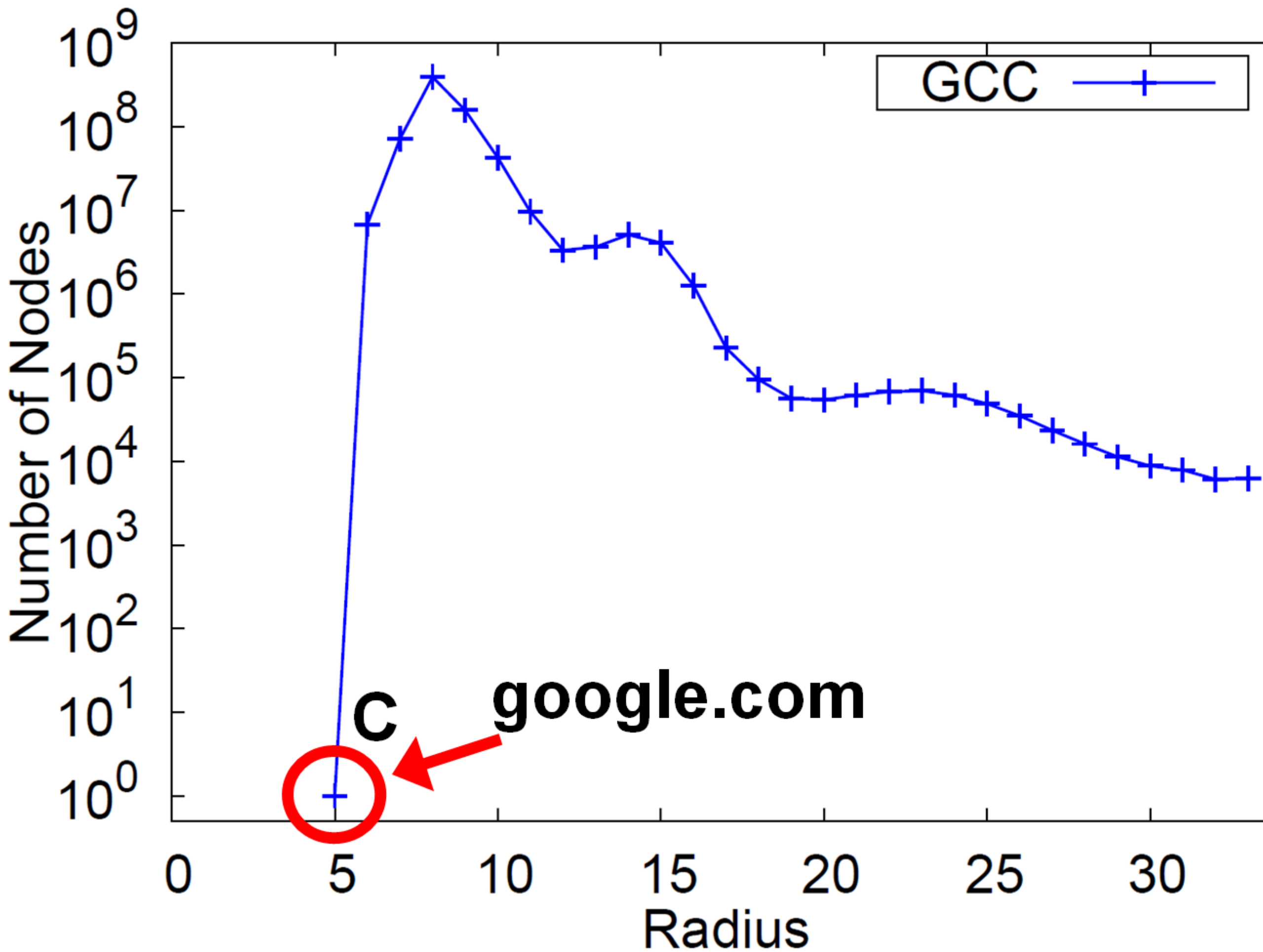
Q: What is the most central node in the Web?

A1.2: Node Centrality

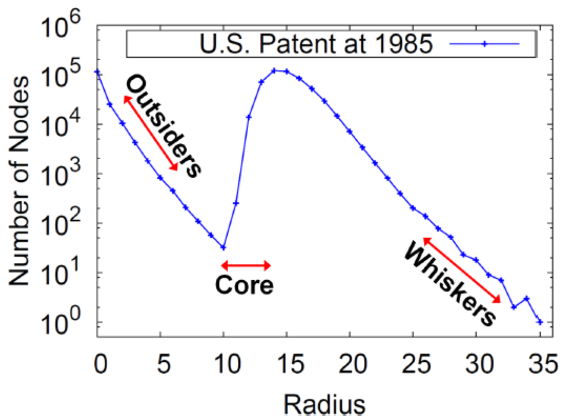


[Kang et al. SDM'10]

A1.2: Node Centrality



A1.3: Radius plots over time



?

?

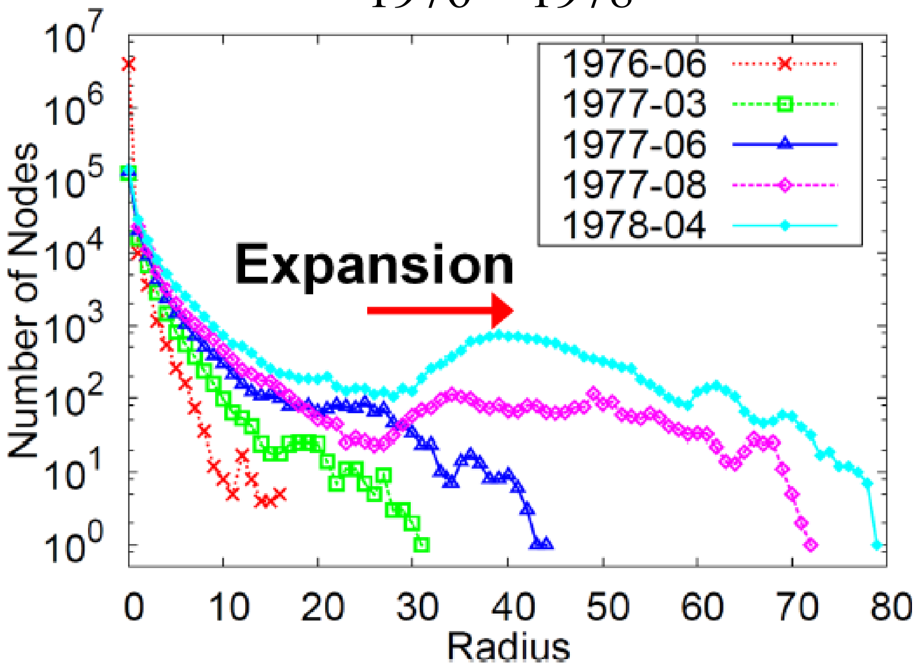
At time T **T+1** **T+2**

Q: How the radius plots change over time?

[Kang et al. SDM'10]

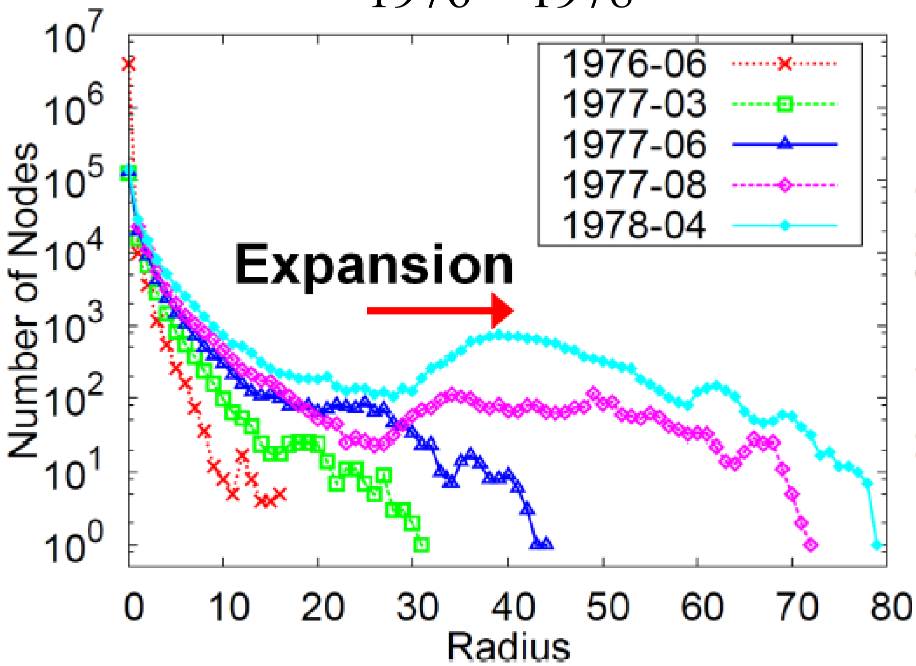
A1.3: Radius plots over time

1976 ~ 1978

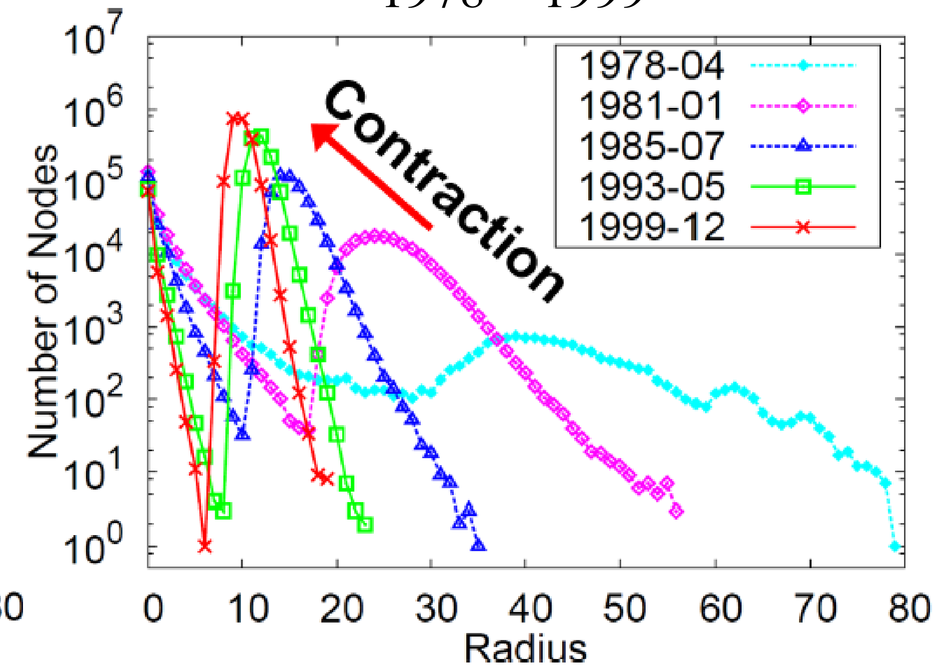


A1.3: Radius plots over time

1976 ~ 1978



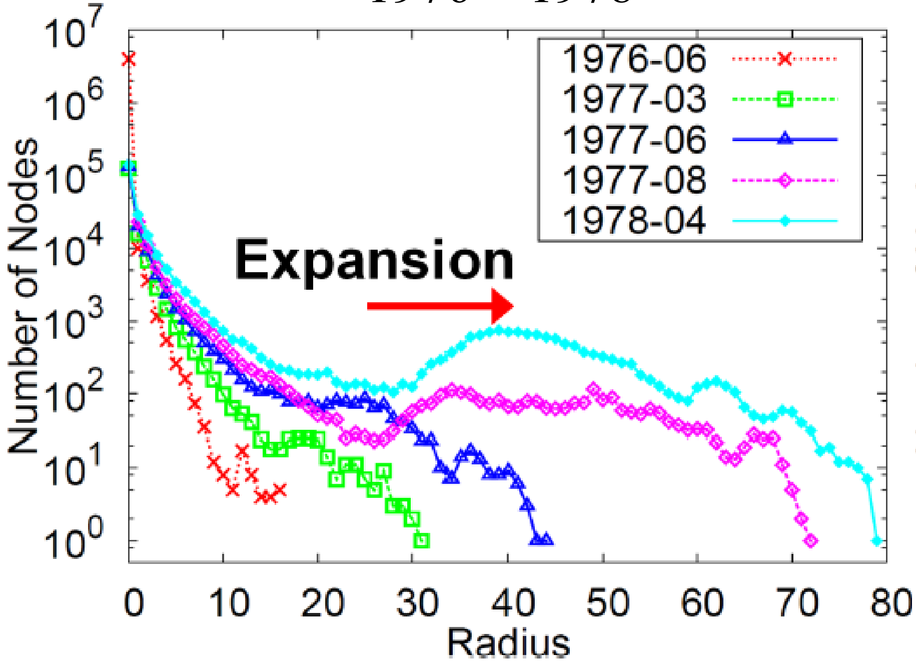
1978 ~ 1999



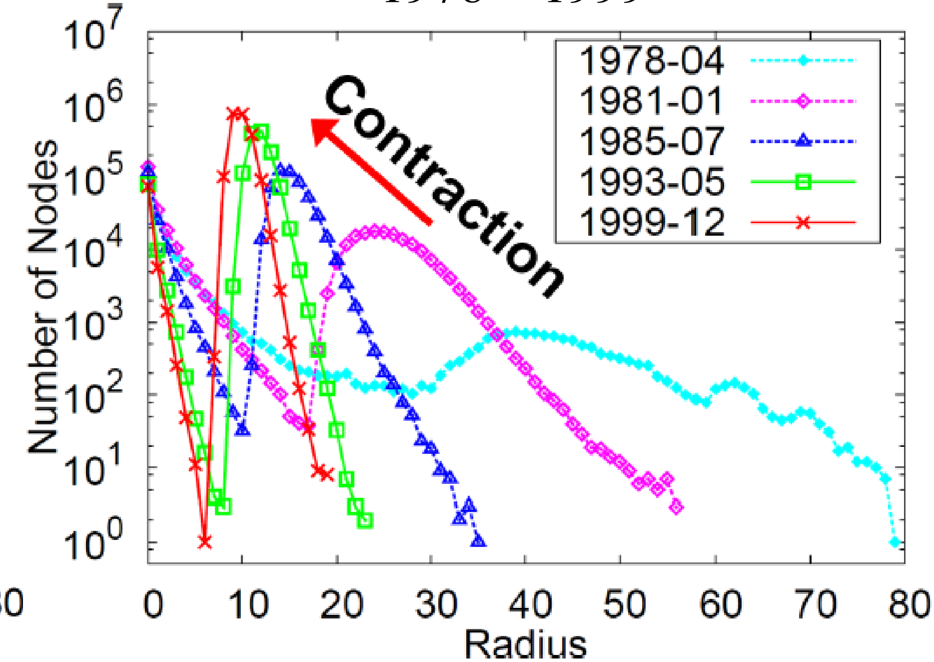
A: Expansion-Contraction!

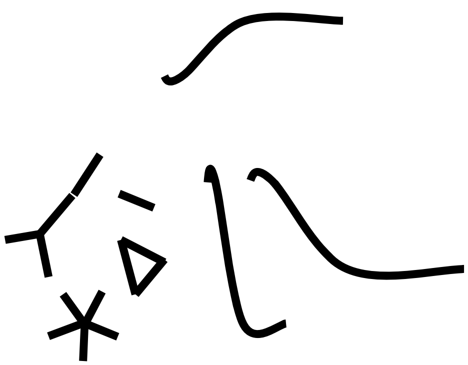
A1.3: Radius plots over time

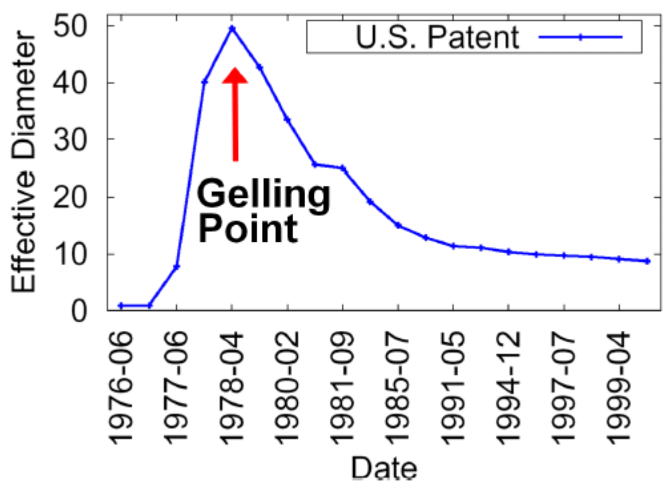
1976 ~ 1978

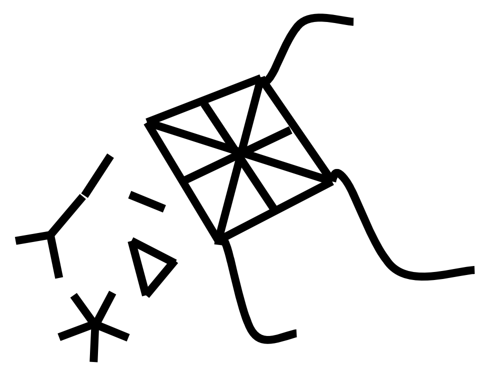


1978 ~ 1999









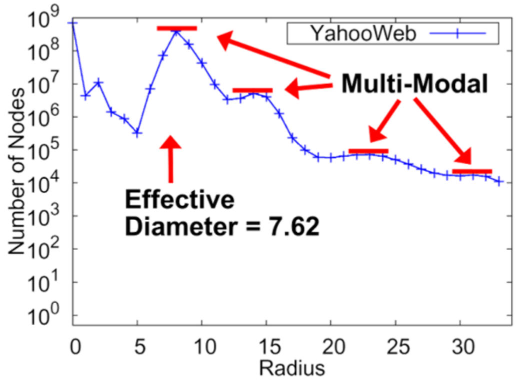
Task	Discoveries	Algorithm
Structure of Large Graphs	Q1: What do large networks look like?	Q2: How to scale-up structure analysis algorithm?
Eigensolver	Q3: How to spot strange behaviors in networks?	Q4: How to design a billion-scale eigensolver?
Tensor Decomposition	Q5: What are the important concepts and synonyms in a KB tensor?	Q6: How to decompose a billion-scale tensor?

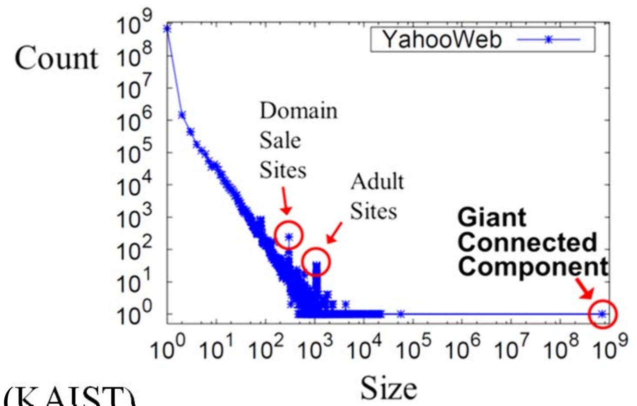
Outline

- ☑ **Motivation**
- ☐ Structure of Large Graphs
  - D1. Radius Plots
  - A1. GIM-V
- ☐ Eigensolver
- ☐ Tensor Decomposition
- ☐ Conclusions

Problem Definition

- Q2: How to scale-up structure analysis algorithm?
  - Q2.1: How to **unify** many structure analysis algorithms (connected components, PageRank, diameter/radius)?
  - Q2.2: How to design a **scalable** algorithm for the structure analysis?





[Kang et al. ICDM'09]

Q2.1: Unifying Algorithms

- Given a graph, can we compute
  - connected components,
  - PageRank,
  - Random Walk with Restart,
  - diameter/radius

  with ***one algorithm***?

[Kang et al. ICDM'09]

Q2.1: Unifying Algorithms

- Given a graph, can we compute
  - connected components,
  - PageRank,
  - Random Walk with Restart,
  - diameter/radius

  with ***one algorithm***?

Yes!

How ?

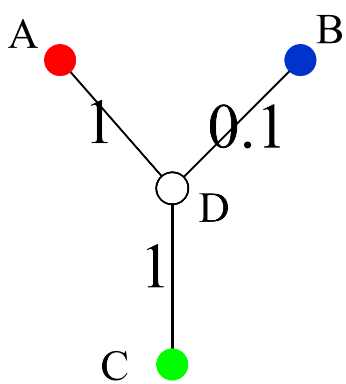
Main Idea

- GIM-V
  - Generalized Iterative Matrix-Vector Multiplication
  - Extension of plain matrix-vector multiplication
  - includes
    - Connected Components
    - PageRank
    - RWR (Random Walk With Restart)
    - Diameter Estimation

[Kang et al. ICDM'09]

Main Idea: Intuition

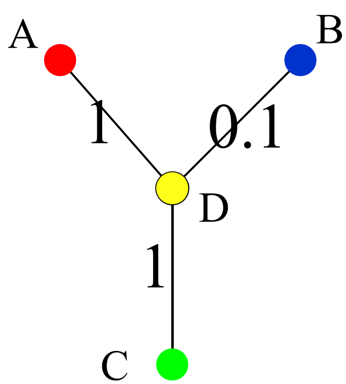
- Plain M-V multiplication



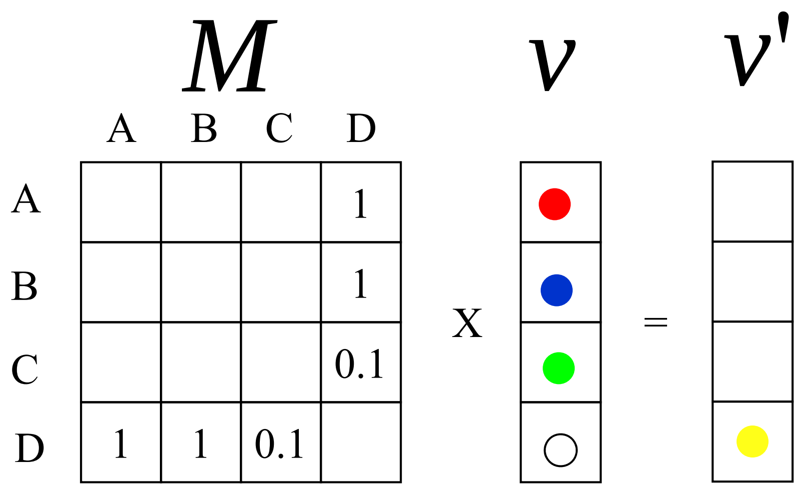
- Weighted Combination of Colors
- ~ Message Passing

Main Idea: Intuition

- Plain M-V multiplication



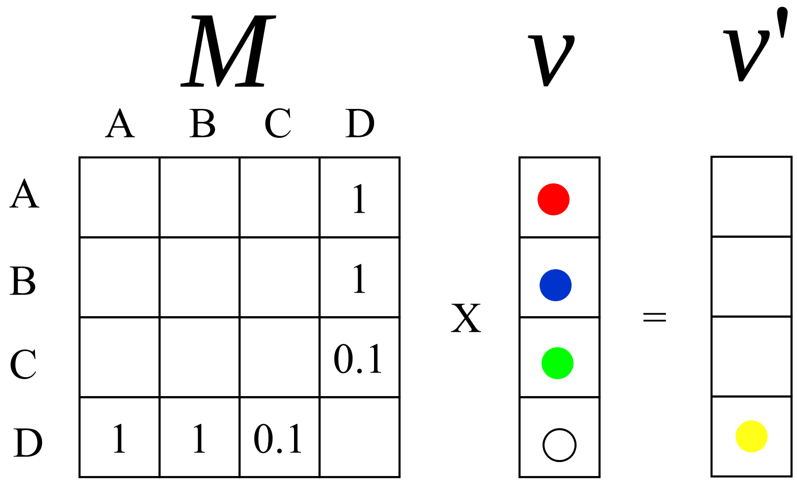
- Weighted Combination of Colors
- ~ Message Passing

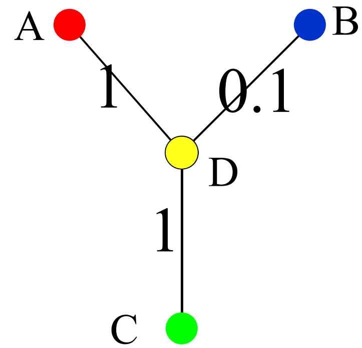


$$v_4' = \sum_{i=1}^{4} m_{4i} v_i$$

Main Idea: Intuition

- Plain M-V multiplication

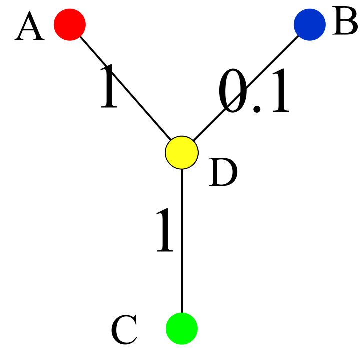




$$M \times v = v'$$

$$v_j' = \sum_{i=1}^{4} m_{ji} v_i$$

Main Idea: Intuition



- Plain M-V multiplication

M v v'

	A	B	C	D
A				1
B				1
C				0.1
D	1	1	0.1	

X v = v'

$$M \times v = v'$$

$$v_j' = \sum_{i=1}^{4} m_{ji} v_i$$

Three Implicit Operations here:

multiply m_{ji} and v_i	combine2	Message sending
sum n multiplication results	combineAll	Message combination
update v_j'	assign	

[Kang et al. ICDM’09]

Main Idea

- GIM-V

Operations	Standard MV	Con. Cmpt.	PageRank	RWR	(approx.) Diameter
combine2	Multiply	Multiply	Multiply with c	Multiply with c	Multiply bit-vector
combineAll	Sum	MIN	Sum with rj prob.	Sum with restart prob	BIT-OR()
assign	Assign	MIN	Assign	Assign	BIT-OR()

[Kang et al. ICDM'09]

Q2.2: Scalable Algorithm

- The sizes of graphs are growing!

facebook

0.5 billion users
60 TBytes/day
15 PBytes/total
[Thusoo+ '10]

bing

ClickStream Data
0.26 PBytes
1 billion query-URL
[Liu+ '09]

YAHOO!

1.4 billion web pages
6.6 billion edges
[Broder+ '04]

Google

20 PBytes/day

[Dean+ '08]

[Kang et al. ICDM'09]

Q2.2: Scalable Algorithm

- The sizes of graphs are growing!

Q: How can we handle large graphs which don't fit into the memory, or disks of a single machine?

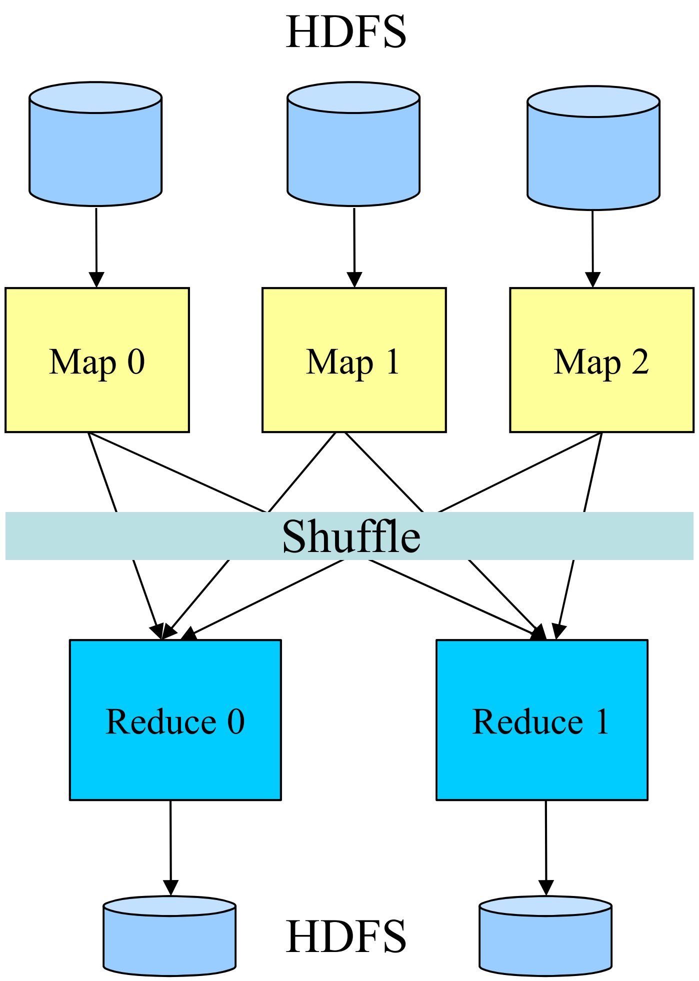
A: Parallelism, with MapReduce!

[Broder+ '04]

[Dean+ '08]

Background: MapReduce

- MapReduce/Hadoop Framework



HDFS: fault tolerant, scalable, distributed storage system

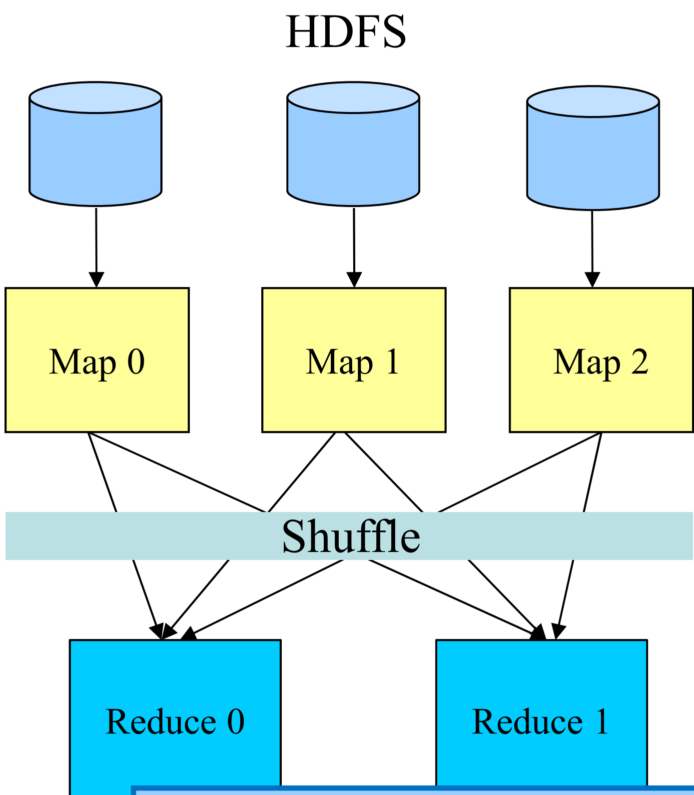
Mapper: read data from HDFS, output (k,v) pair

Output sorted by the key

Reducer: read output from mappers, output a new (k,v) pair to HDFS

Background: MapReduce

- MapReduce/Hadoop Framework



HDFS: fault tolerant, scalable, distributed storage system

Mapper: read data from HDFS, output (k,v) pair

Output sorted by the key

Reducer: read output from mappers, output a new (k,v) pair

Programmers need to provide only map() and reduce() functions

Two Restrictions on HDFS

- [R1] HDFS is location transparent
  - Users don't know which file is located in which machine
- [R2] A line is never split
  - A large file is split into pieces of a size(e.g. 256 MB)
  - Users don't know the point of the split

Fast Algorithms for GIM-V

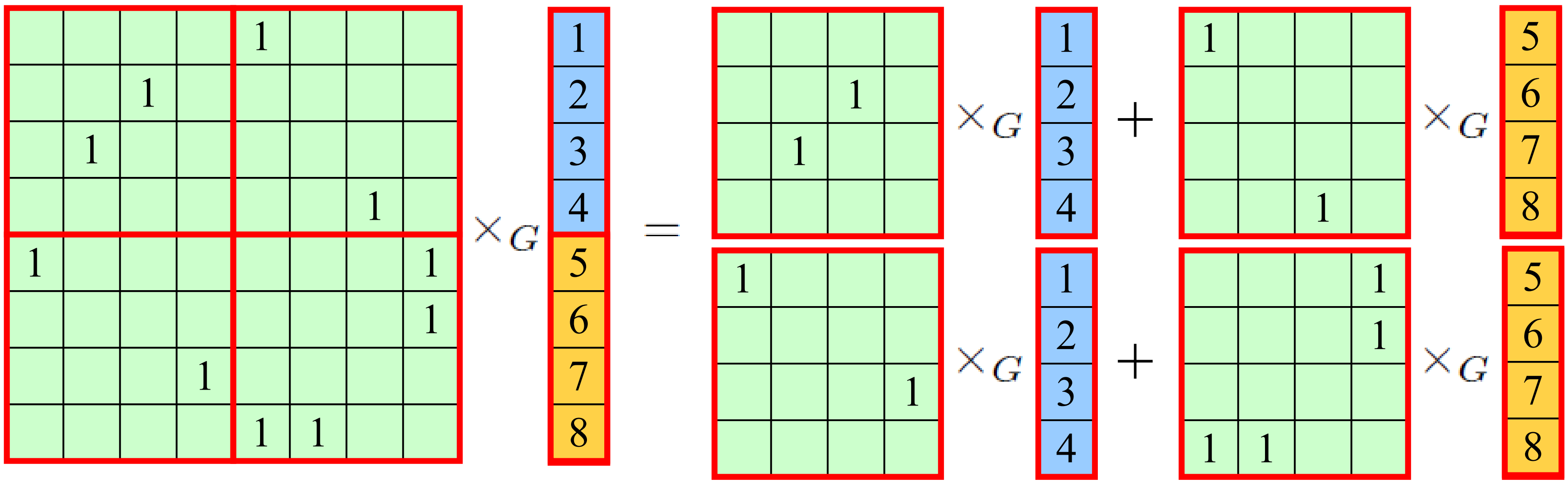
- Given the two restrictions R1 and R2, how can we make faster algorithms for GIM-V in Hadoop?
  - Three main ideas:
    - I1) Block Multiplication
    - I2) Clustering
    - I3) Compression

KAIST

[Kang et al. ICDM'09]

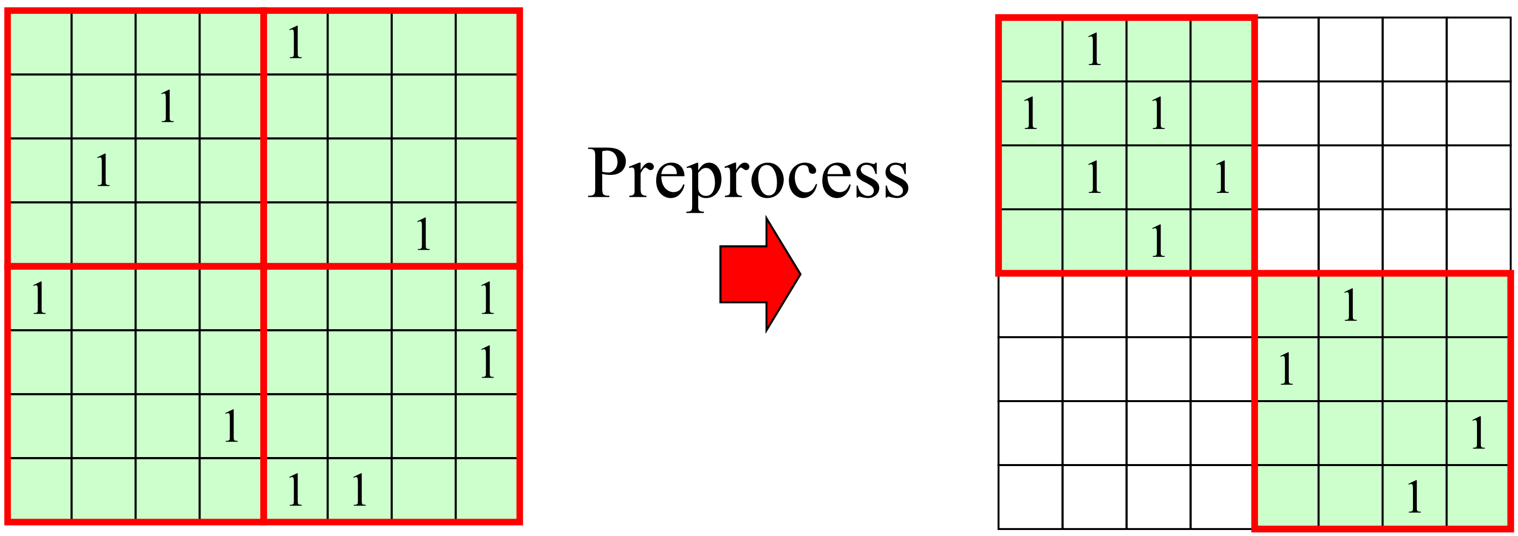
Fast Algorithms for GIM-V

- I1) Block-Method



Fast Algorithms for GIM-V

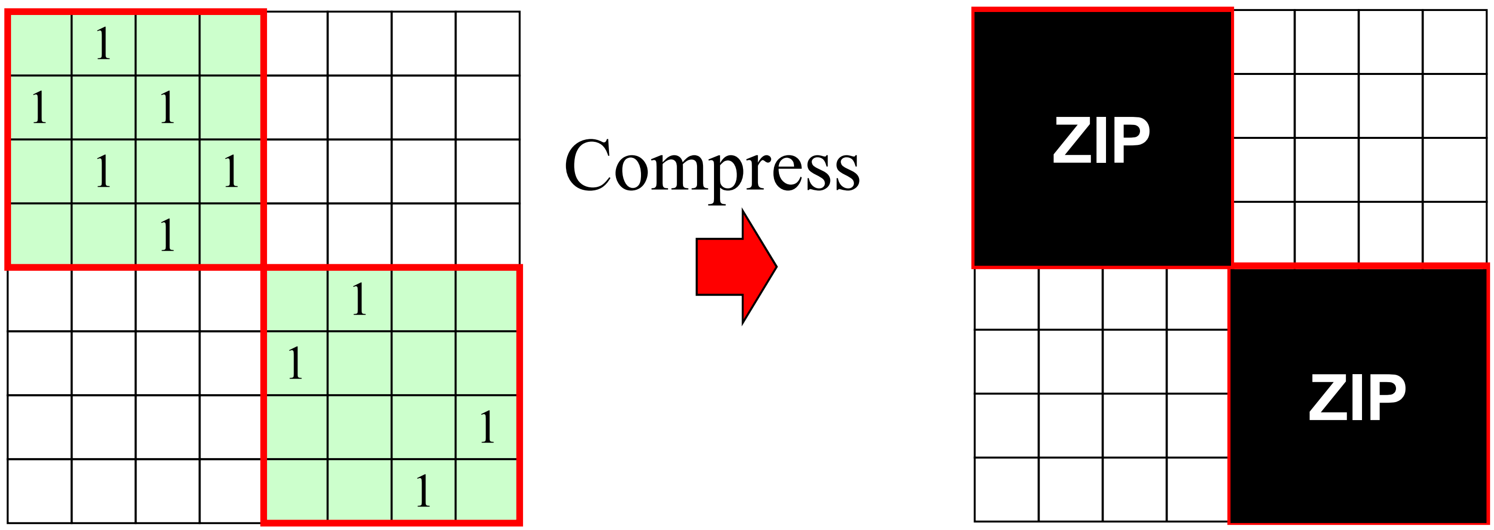
- I2) Clustering



A: preprocessing for clustering
(only green blocks are stored in HDFS)

Fast Algorithms for GIM-V

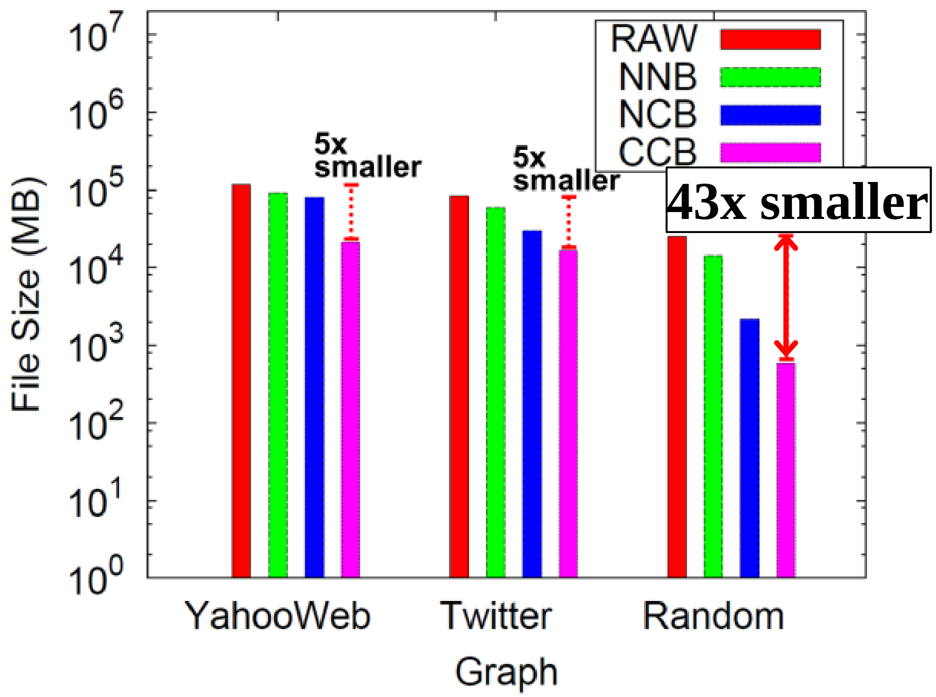
- I3) Compression

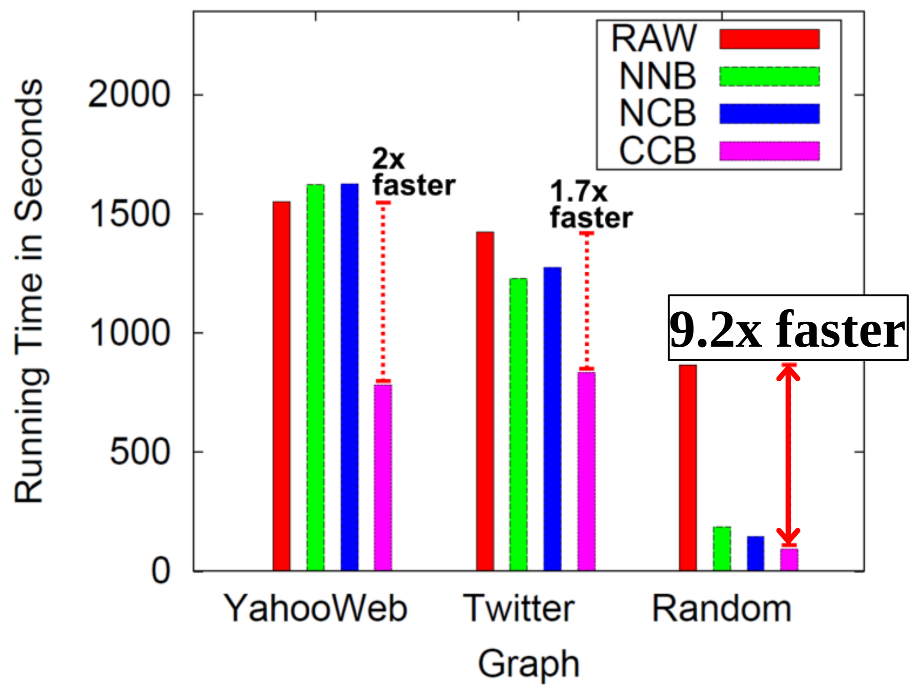


A: compress clustered blocks

[Kang et al. VLDBJ'12]

Fast Algorithms for GIM-V

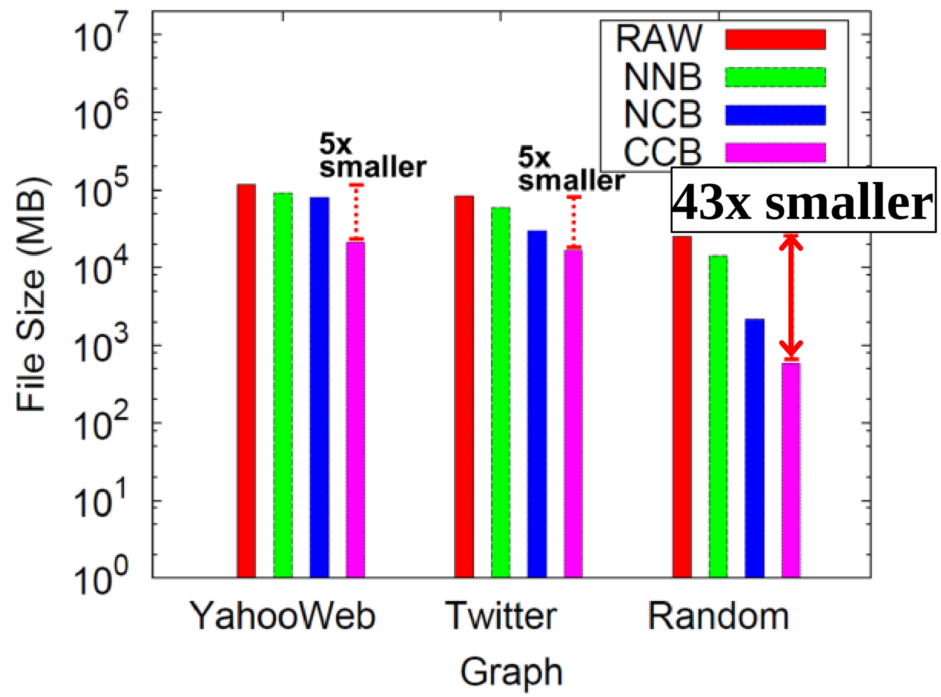


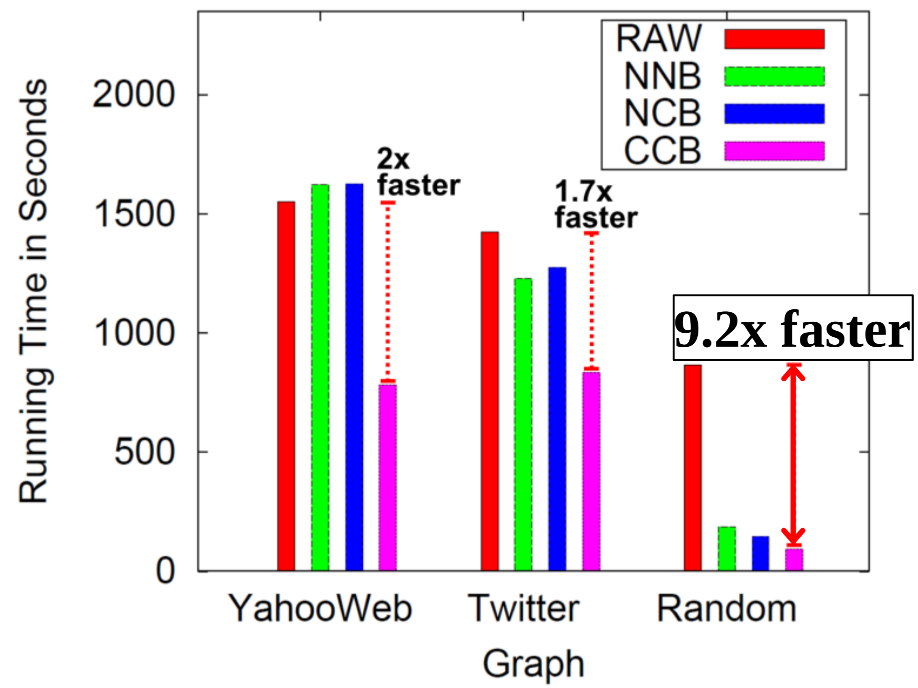


	Block Encoding?	Compression?	Clustering?
RAW	No	No	No
NNB	Yes	No	No
NCB	Yes	Yes	No
CCB	Yes	Yes	Yes

[Kang et al. VLDBJ’12]

Fast Algorithms for GIM-V





A: Proposed Method(CCB) provides 43x smaller storage, 9.2x faster running time

Outline

Task	Discoveries	Algorithm
Structure of Large Graphs	Q1: What do large networks look like?	Q2: How to scale-up structure analysis algorithm?
Eigensolver	Q3: How to spot strange behaviors in networks?	Q4: How to design a billion-scale eigensolver?
Tensor Decomposition	Q5: What are the important concepts and synonyms in a KB tensor?	Q6: How to decompose a billion-scale tensor?

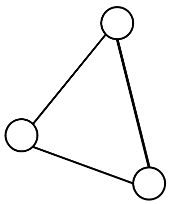
- ☑ **Motivation**
- ☑ Structure of Large Graphs
- ☐ Eigensolver
  - D2. Triangle Counting
  - A2. HEigen
- ☐ Tensor Decomposition
- ☐ Conclusions

[Kang et al. PAKDD'11]

Triangle Counting

- Q3: How to spot strange behaviors in networks?
  - E.g.) Twitter who-follows-whom graph?

Triangle Counting

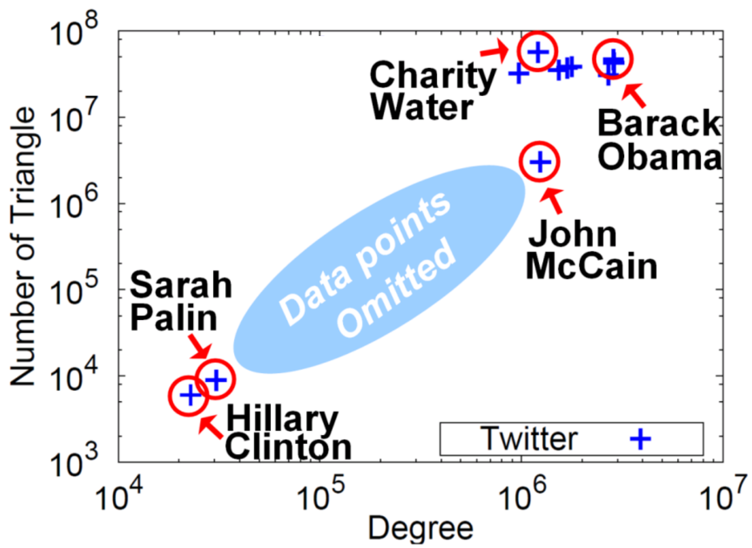


- Triangle Counting
  - Real social networks have a lot of triangles
    - Friends of friends are friends
- But, triangles are expensive to compute
  - (3-way join; several approx. algos)
- Q: Can we do that quickly?
- A: Yes!
  - #triangles = $\frac{1}{6}\sum_i {\lambda_i}^3$
  - (and, because of skewness in eigenvalues, we only need the top few eigenvalues!)

[Kang et al. PAKDD'11]

Triangle Counting

- Triangle counting in Twitter social network

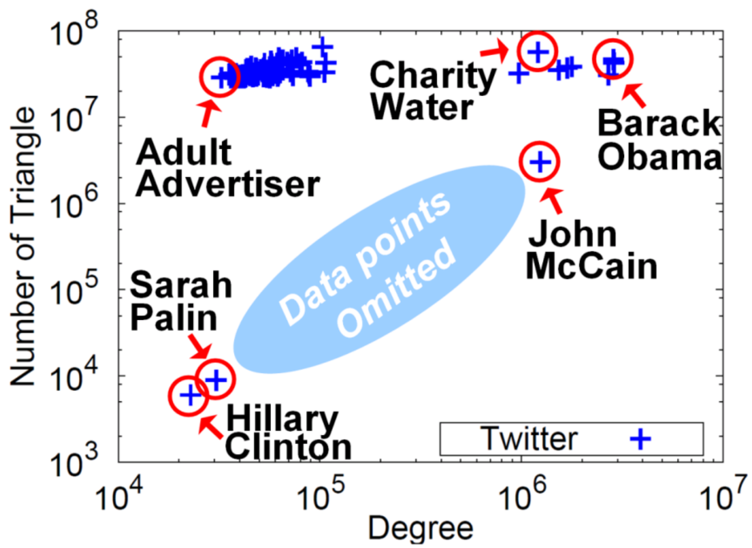


[Twitter 2009;
~ 60 million nodes
~ 3 billion edges]

- U.S. politicians: moderate number of triangles vs. degree

Triangle Counting

- Triangle counting in Twitter social network



[Twitter 2009;
~ 60 million nodes
~ 3 billion edges]

- U.S. politicians: moderate number of triangles vs. degree
- Adult sites: very large number of triangles vs. degree

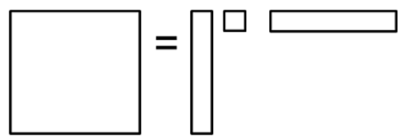
Outline

Task	Discoveries	Algorithm
Structure of Large Graphs	Q1: What do large networks look like?	Q2: How to scale-up structure analysis algorithm?
Eigensolver	Q3: How to spot strange behaviors in networks?	Q4: How to design a billion-scale eigensolver?
Tensor Decomposition	Q5: What are the important concepts and synonyms in a KB tensor?	Q6: How to decompose a billion-scale tensor?

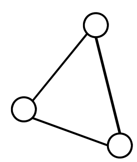
- ☑ **Motivation**
- ☑ Structure of Large Graphs
- ☐ Eigensolver
  - D2. Triangle Counting
  - A2. HEigen
- ☐ Tensor Decomposition
- ☐ Conclusions

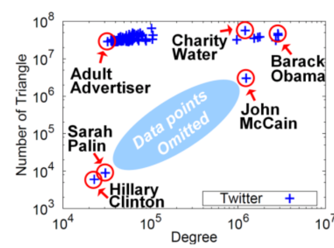
Background: Eigensolver

- Eigensolver
  - Given: (adjacency) matrix A,
  - Compute: top k eigenvalues and eigenvectors of A
  - Application:
    - SVD

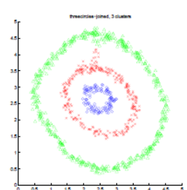


    - triangle counting





    - spectral clustering



[Kang et al. PAKDD'11]

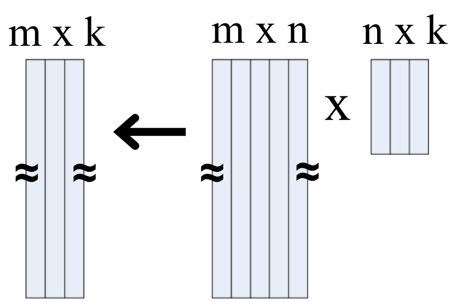
Problem Definition

- Q4: How to design a billion-scale eigensolver?
  - Existing eigensolver: can handle millions of nodes and edges

[Kang et al. PAKDD'11]

Proposed Method

- HEigen algorithm (Hadoop Eigen-solver)
  - Selectively parallelize 'Lanczos-SO' algorithm
  - Block encoding
  - Exploiting skewness in matrix-matrix mult.
    - (m >> n > k)

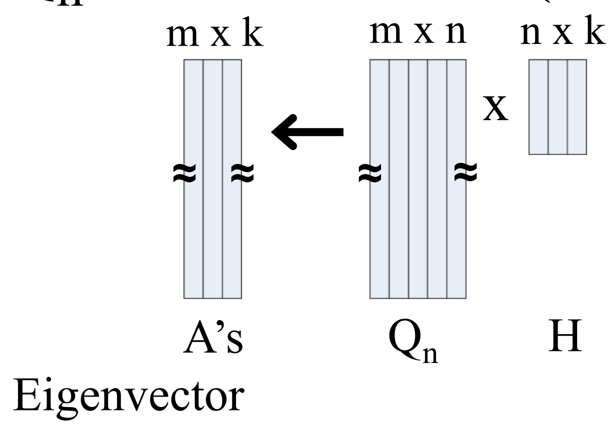


[Kang et al. PAKDD'11]

Details

Skewed Matrix-Matrix Mult.

- Multiply $Q_n^{m x n}$ and $H^{n x k}$ $(m >> n > k)$



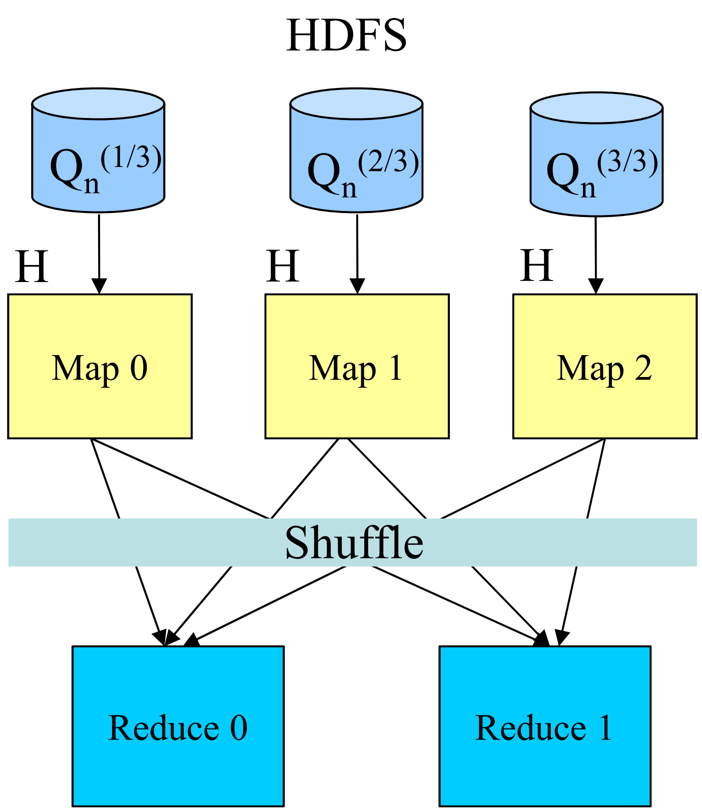
Q_n: O(100 Gbytes)
H: O(Kbytes)

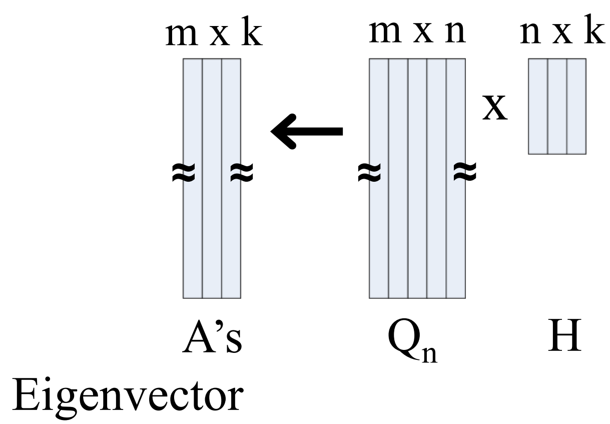
- Naïve multiplication: too expensive
- Proposed:
  - `cache'-based multiplication: broadcast the small matrix H to all the machines that contains Q_n

Details

Skewed Matrix-Matrix Mult.

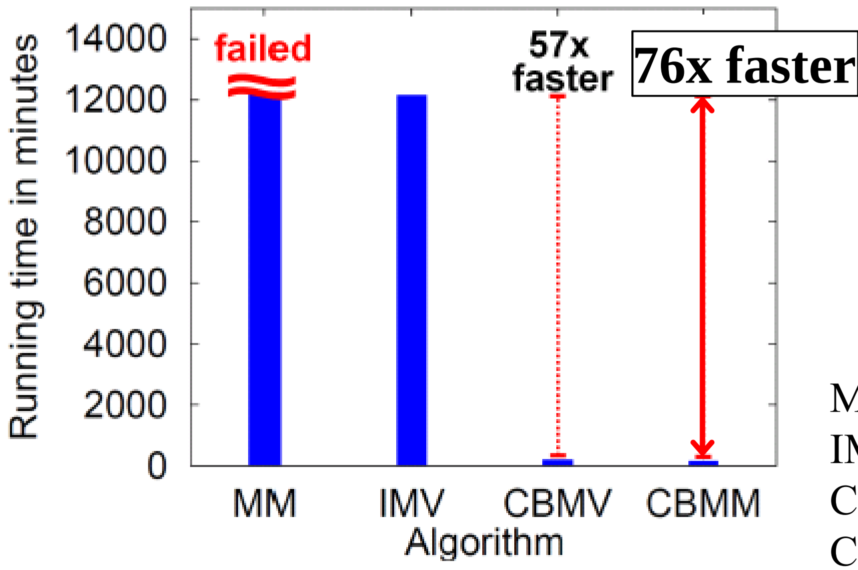
- `cache'-based multiplication: broadcast the small matrix H to all the machines that contains Q_n



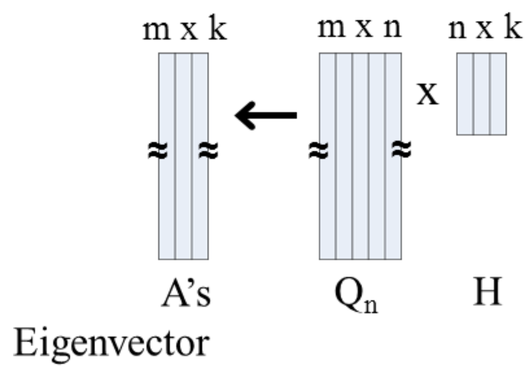


Skewed Matrix-Matrix Mult.

Which Matrix-Matrix multiplication algorithm runs the fastest?



Time vs. algorithms



MM: naïve mat-mat mult.
IMV: naïve iterative mat-vec mult.
CBMV: cache-based iterative mat-vec mult.
CBMM: cache-based mat-mat mult.

(100 machines used)

Cache-based MM runs 76x faster

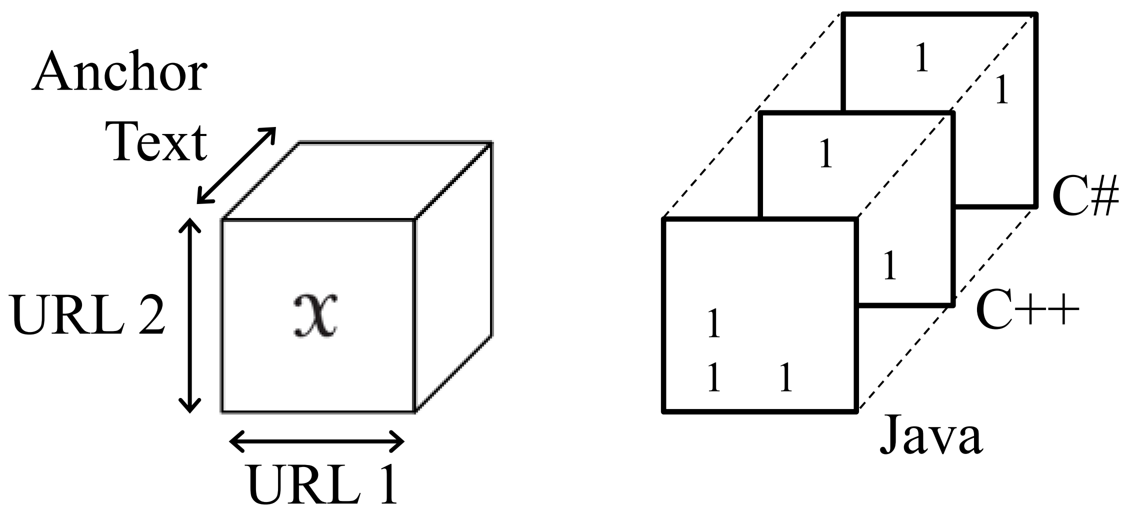
Task	Discoveries	Algorithm
Structure of Large Graphs	Q1: What do large networks look like?	Q2: How to scale-up structure analysis algorithm?
Eigensolver	Q3: How to spot strange behaviors in networks?	Q4: How to design a billion-scale eigensolver?
Tensor Decomposition	Q5: What are the important concepts and synonyms in a KB tensor?	Q6: How to decompose a billion-scale tensor?

Outline

- ☑ **Motivation**
- ☑ Structure of Large Graphs
- ☑ Eigensolver
- ☐ Tensor Decomposition
  - D3. Knowledge Base Tensor
  - A3. GigaTensor
- ☐ Conclusions

Background: Tensor

- Tensors (=multi-dimensional arrays) are everywhere
  - Hyperlinks and anchor texts in Web graphs

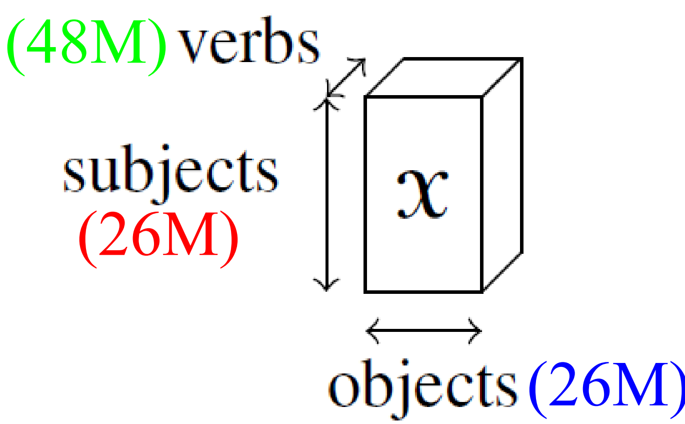


Background: Tensor

- Tensors (=multi-dimensional arrays) are everywhere
  - Sensor stream (time, location, type)
  - Predicates (subject, verb, object) in knowledge base

"Eric Clapton plays guitar"

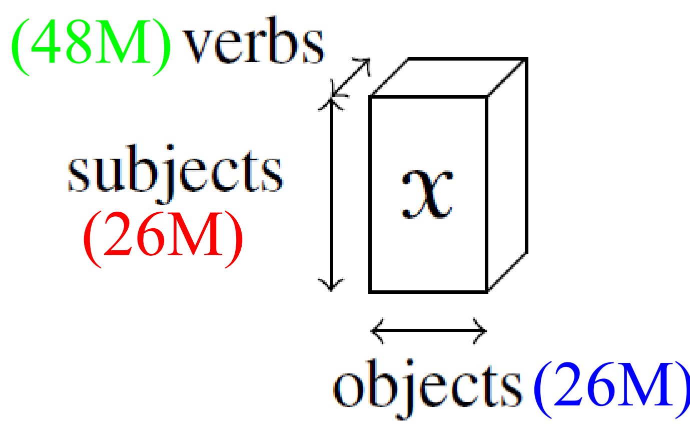
"Barrack Obama is the president of U.S."



NELL (Never Ending Language Learner) data
Nonzeros =144M

Problem Definition

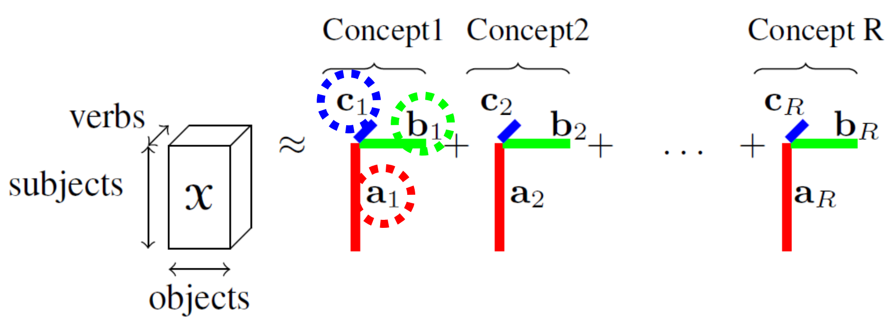
- Q5: What are the important concepts and synonyms in a KB tensor?
  - Q5.1: What are the dominant concepts in the knowledge base tensor?
  - Q5.2: What are the synonyms to a given noun phrase?



NELL (Never Ending Language Learner) data
Nonzeros =144M

A5.1: Concept Discovery

- Concept Discovery in Knowledge Base



Noun Phrase 1	Noun Phrase 2	Context
Concept 1: "Web Protocol"		
internet	protocol	'np1' 'stream' 'np2'
file	software	'np1' 'marketing' 'np2'
data	suite	'np1' 'dating' 'np2'
Concept 2: "Credit Cards"		
credit	information	'np1' 'card' 'np2'
Credit	debt	'np1' 'report' 'np2'
library	number	'np1' 'cards' 'np2'
Concept 3: "Health System"		
health	provider	'np1' 'care' 'np2'
child	providers	'np' 'insurance' 'np2'
home	system	'np1' 'service' 'np2'
Concept 4: "Family Life"		
life	rest	'np2' 'of' 'my' 'np1'
family	part	'np2' 'of' 'his' 'np1"
body	years	'np2' 'of' 'her' 'np1'

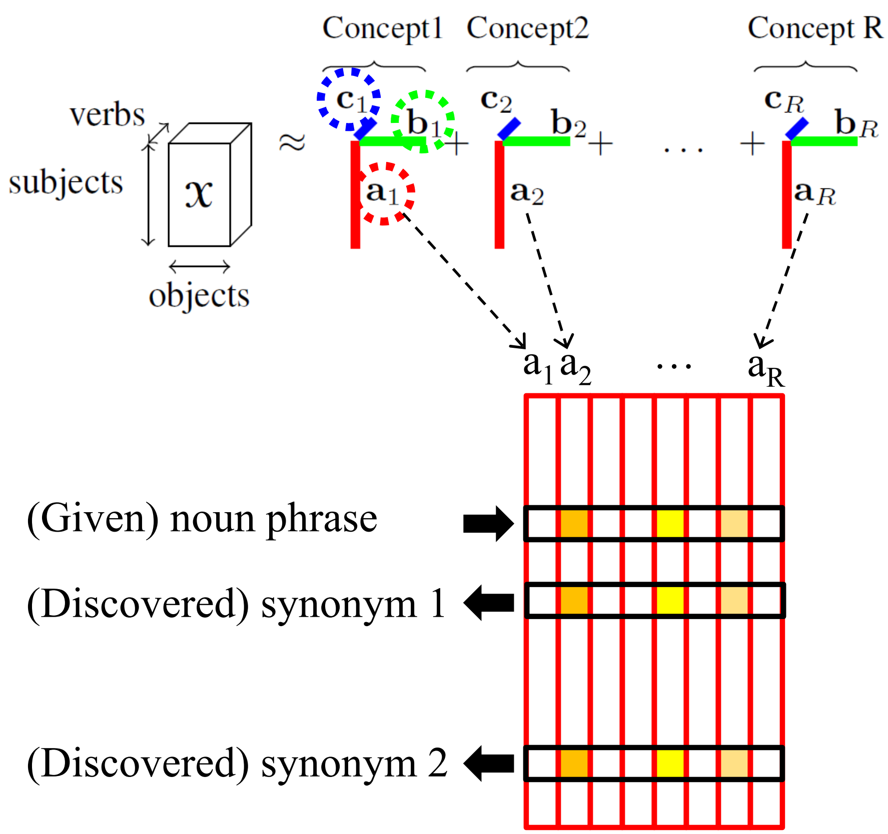
[Kang et al. KDD'12]

A5.1: Concept Discovery

Noun Phrase 1	Noun Phrase 2	Context
Concept 1: "Web Protocol"		
internet	protocol	'np1' 'stream' 'np2'
file	software	'np1' 'marketing' 'np2'
data	suite	'np1' 'dating' 'np2'
Concept 2: "Credit Cards"		
credit	information	'np1' 'card' 'np2'
Credit	debt	'np1' 'report' 'np2'
library	number	'np1' 'cards' 'np2'
Concept 3: "Health System"		
health	provider	'np1' 'care' 'np2'
child	providers	'np' 'insurance' 'np2'
home	system	'np1' 'service' 'np2'
Concept 4: "Family Life"		
life	rest	'np2' 'of' 'my' 'np1'
family	part	'np2' 'of' 'his' 'np1"
body	years	'np2' 'of' 'her' 'np1'

A5.2: Synonym Discovery

- Synonym Discovery in Knowledge Base



(Given) Noun Phrase	(Discovered) Potential Synonyms
pollutants	dioxin, sulfur dioxide, greenhouse gases, particulates, nitrogen oxide, air pollutants, cholesterol
disabilities	infections, dizziness, injuries, diseases, drowsiness, stiffness, injuries
vodafone	verizon, comcast
Christian history	European history, American history, Islamic history, history
disbelief	dismay, disgust, astonishment
cyberpunk	online-gaming
soul	body

[Kang et al. KDD'12]

A5.2: Synonym Discovery

(Given) Noun Phrase	(Discovered) Potential Synonyms
pollutants	dioxin, sulfur dioxide, greenhouse gases, particulates, nitrogen oxide, air pollutants, cholesterol
disabilities	infections, dizziness, injuries, diseases, drowsiness, stiffness, injuries
vodafone	verizon, comcast
Christian history	European history, American history, Islamic history, history
disbelief	dismay, disgust, astonishment
cyberpunk	online-gaming
soul	body

Outline

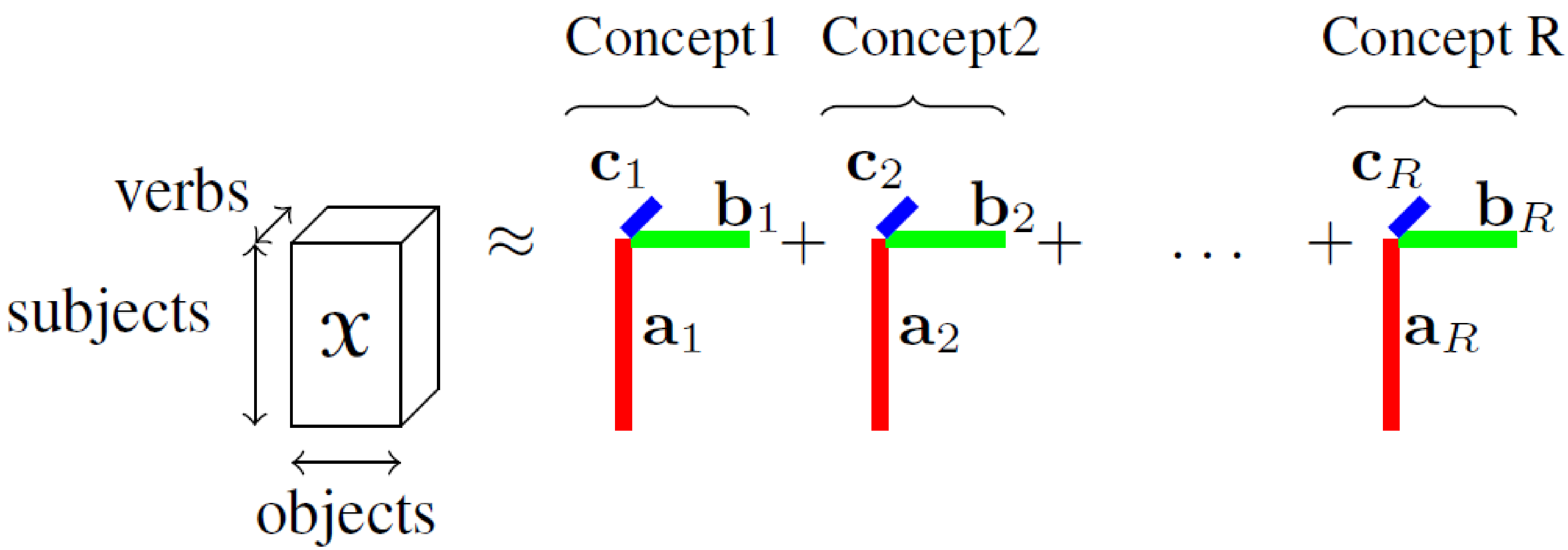
Task	Discoveries	Algorithm
Structure of Large Graphs	Q1: What do large networks look like?	Q2: How to scale-up structure analysis algorithm?
Eigensolver	Q3: How to spot strange behaviors in networks?	Q4: How to design a billion-scale eigensolver?
Tensor Decomposition	Q5: What are the important concepts and synonyms in a KB tensor?	Q6: How to decompose a billion-scale tensor?

- ☑ **Motivation**
- ☑ Structure of Large Graphs
- ☑ Eigensolver
- ☐ Tensor Decomposition
  - D3. Knowledge Base Tensor
  - A3. GigaTensor
- ☐ Conclusions

[Kang et al. KDD'12]

Problem Definition

- Q6: How to decompose a billion-scale tensor?
  - Corresponds to SVD in 2D case

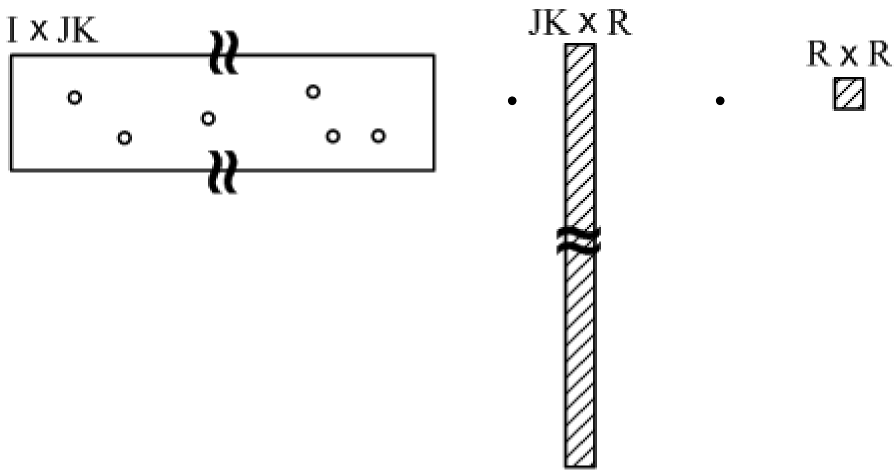


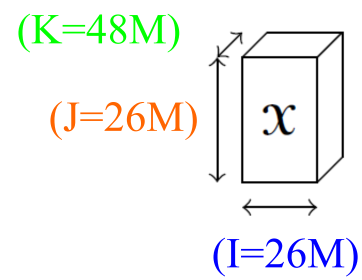
Challenge

Details

- Alternating Least Square (ALS) Algorithm

$$\hat{\mathbf{A}} \leftarrow \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})(\mathbf{C}^T\mathbf{C} * \mathbf{B}^T\mathbf{B})^{\dagger}$$





$\odot$: Khatri-Rao
$*$: Hadamard
$\dagger$: pseudo-inverse

How to design fast MapReduce algorithm for the ALS?

[Kang et al. KDD'12]

Details

Main Idea

- 1. Ordering of Computation

Our choice

$$\boxed{[\mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})](\mathbf{C}^T\mathbf{C} * \mathbf{B}^T\mathbf{B})^{\dagger}}$$

$\mathbf{8 \cdot 10^9}$ FLOPS (NELL data)

$$\mathbf{X}_{(1)}[(\mathbf{C} \odot \mathbf{B})(\mathbf{C}^T\mathbf{C} * \mathbf{B}^T\mathbf{B})^{\dagger}]$$

$\mathbf{2.5 \cdot 10^{17}}$ FLOPS (NELL data)

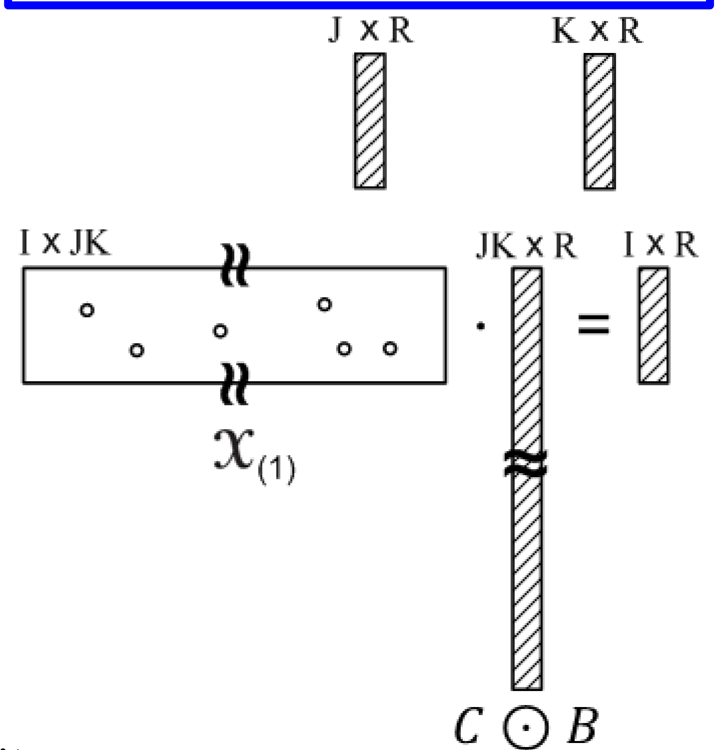
[Kang et al. KDD'12]

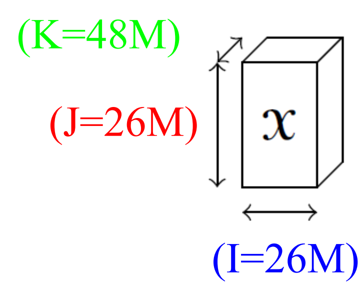
Details

Main Idea

- 2. Avoiding Intermediate Data Explosion

$$\boxed{[\mathbf{X}_{(1)}(\mathbf{C}\odot\mathbf{B})]}(\mathbf{C}^T\mathbf{C}*\mathbf{B}^T\mathbf{B})^\dagger$$





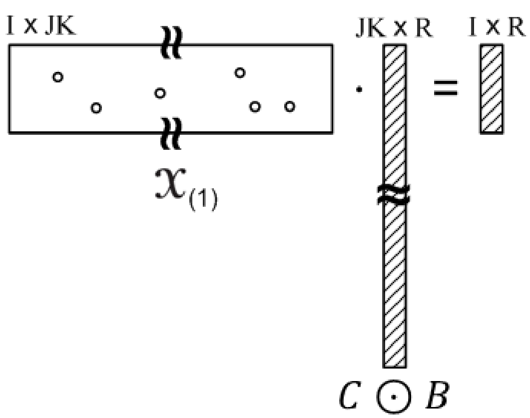
Size of Intermediate Data (NELL)
- Naïve: 100 PB

Main Idea

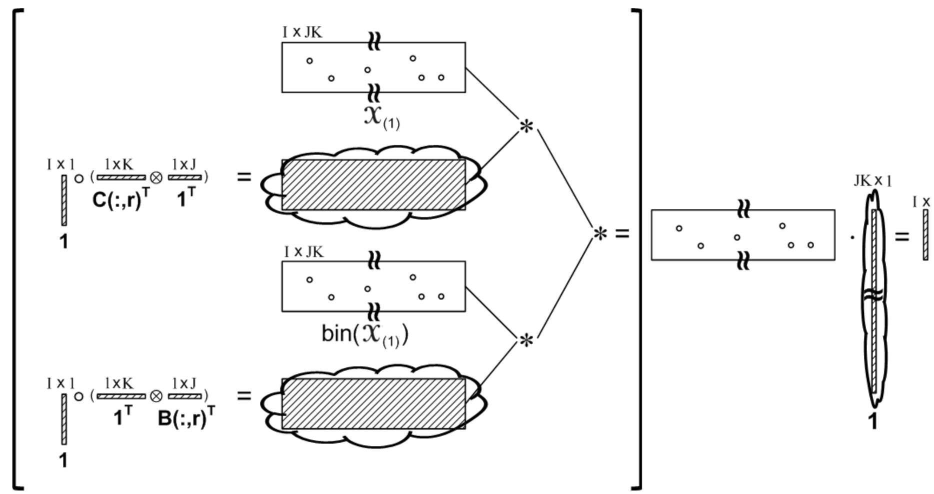
Details

- 2. Avoiding Intermediate Data Explosion

(Before)



(After)



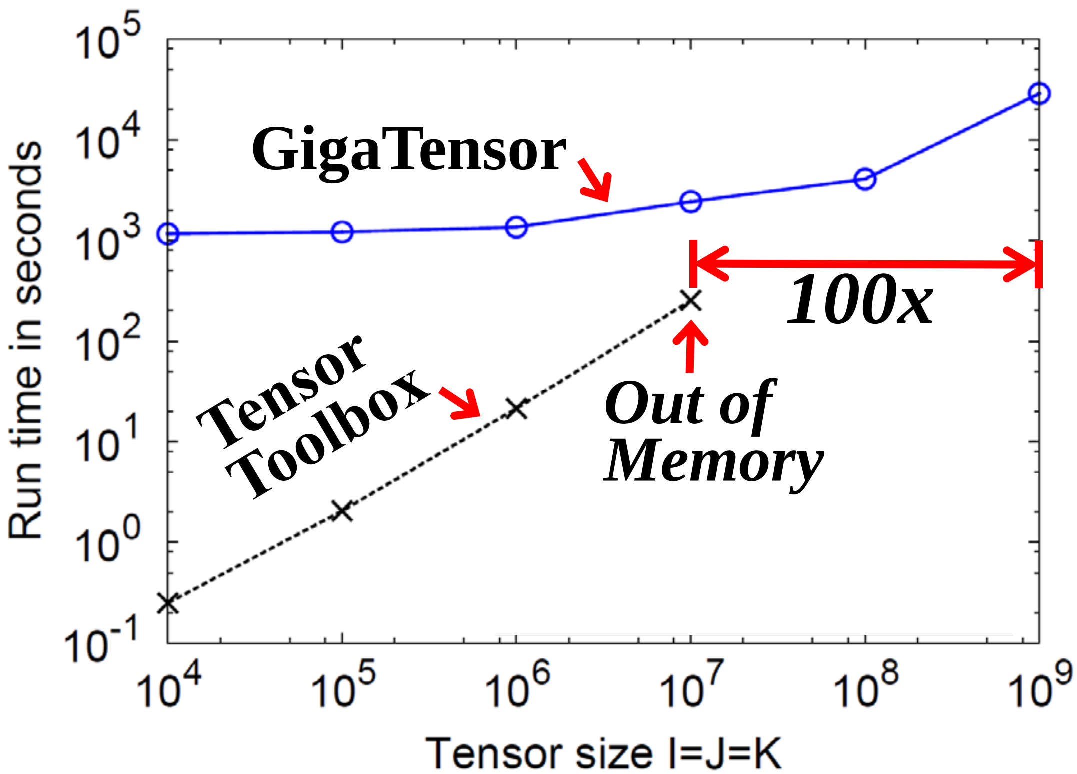
Size of Intermediate Data (NELL)
- Naïve: 100 PB

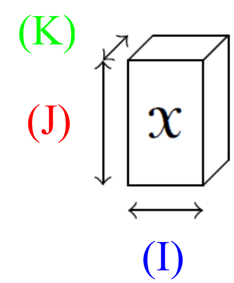
Size of Intermediate Data (NELL)
- Proposed: 1.5 GB

[Kang et al. KDD'12]

Scalability

- GigaTensor solves ***100x*** larger problem





Number of nonzero = I / 50

Outline

- ☑ **Motivation**
- ☑ Structure of Large Graphs
- ☑ Eigensolver
- ☑ Tensor Decomposition
- ☐ Conclusions

Conclusions

- Big graphs open big opportunities for
  - Anomaly detection
  - Scalable algorithms
  - Real-world applications

Conclusions

- PEGASUS: Peta-Scale Graph Mining System
  - 12.8 K lines of JAVA code (Hadoop on M45 cluster)
  - Open source (Apache license)
  - Outreach
    - Downloaded ≥ 800 times from 83 countries
    - 2 U.S. patents, 2 best paper awards
    - Microsoft : part of Hadoop distribution for Windows Azure

Thank you !

web.kaist.ac.kr/~ukang