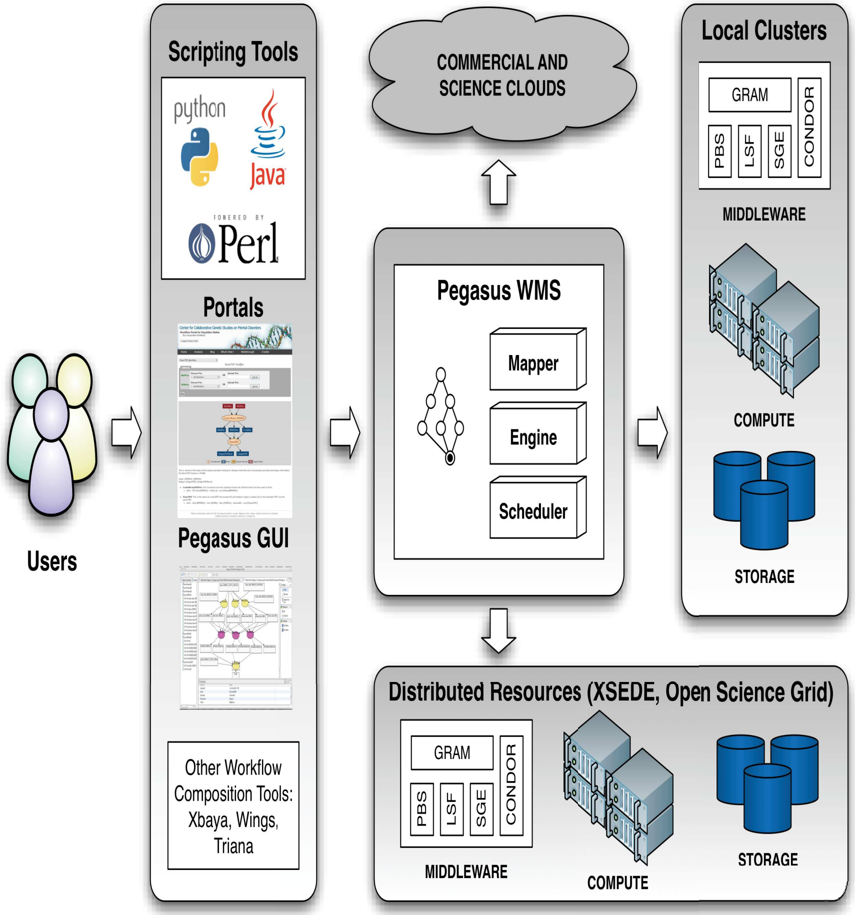
Graph Processing Frameworks

Pegasus

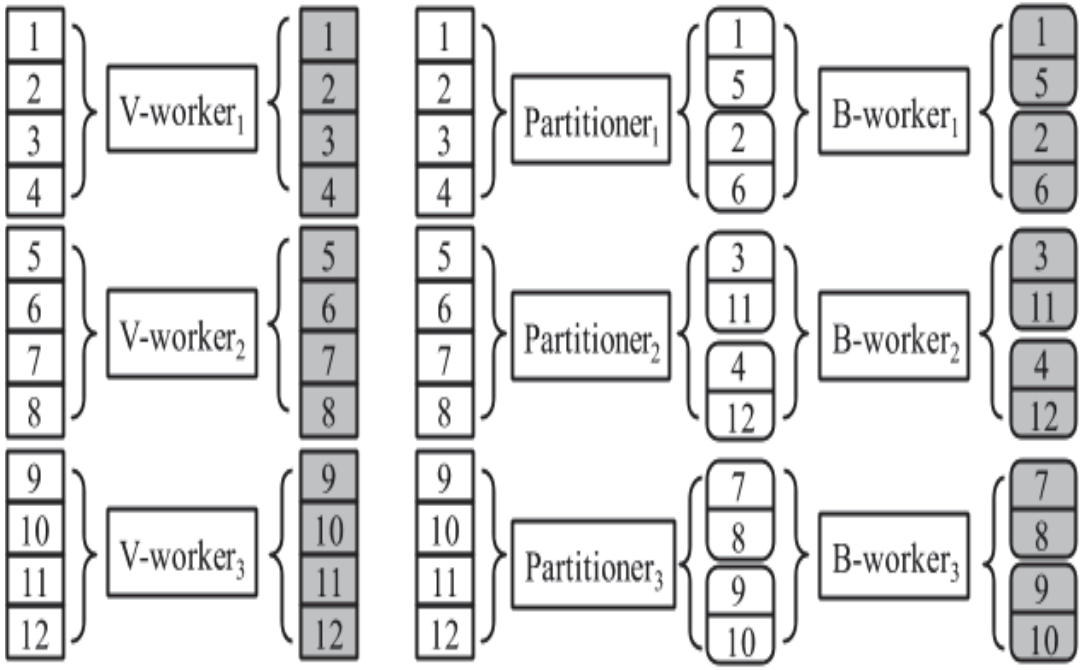
- A system to run, manage and debug complex workflows
- Provides several optimizations
- Provides several API for different languages



Graph Processing Frameworks

Blogel

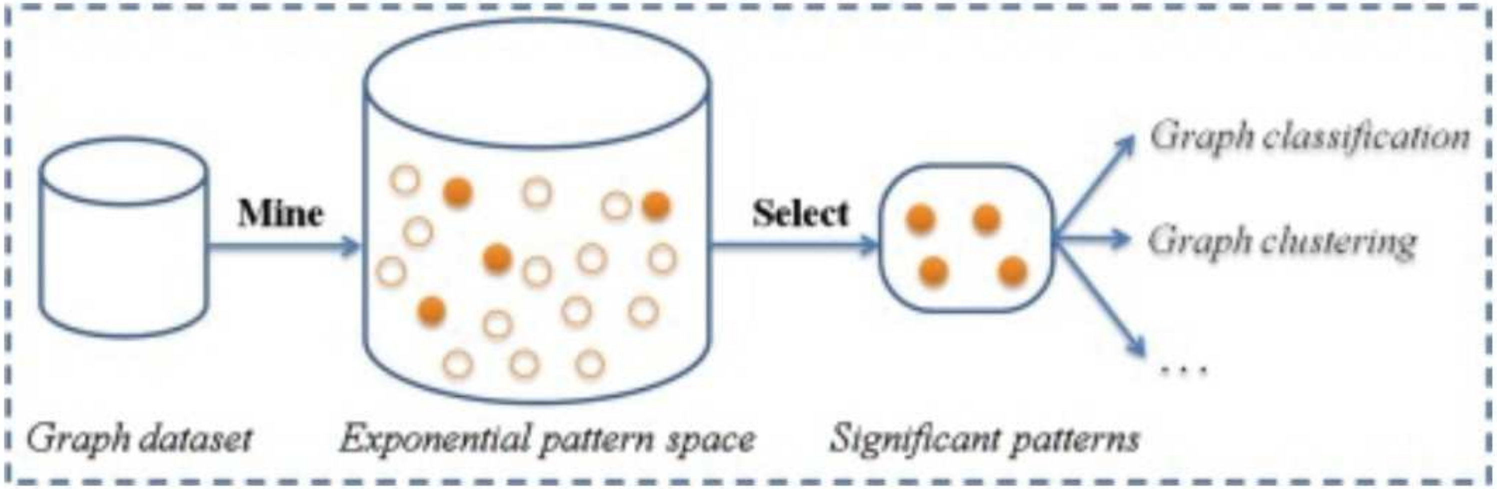
- Graph processing Framework
- Block-centric
- Three computing modes
  - B-mode,
  - V-mode,
  - VB-mode



(a) V-workers

(b) Partitioners and B-workers

Pattern mining in big graphs



Big Graphs Analytics

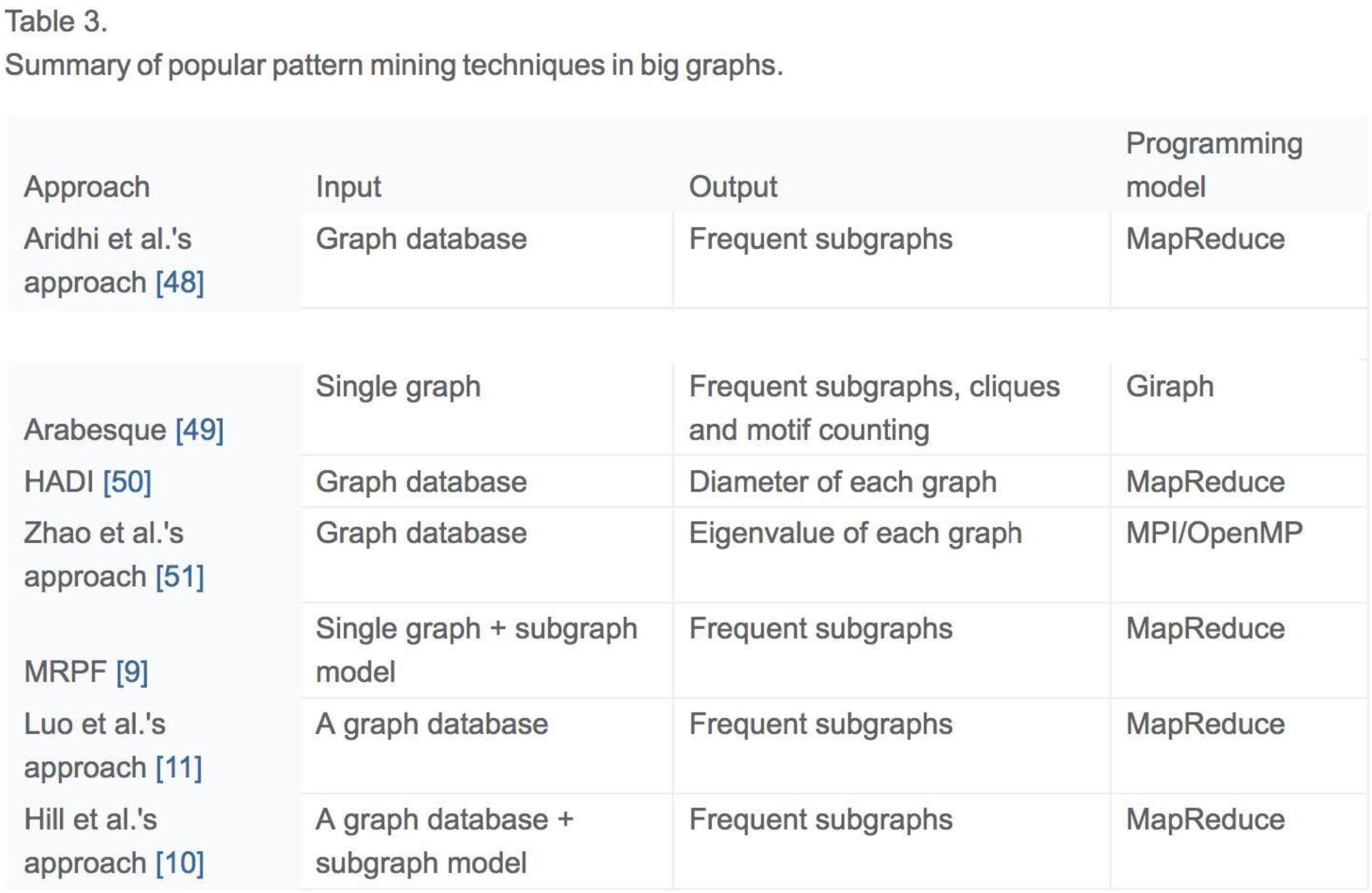


Table 3.
Summary of popular pattern mining techniques in big graphs.

Approach	Input	Output	Programming model
Aridhi et al.'s approach [48]	Graph database	Frequent subgraphs	MapReduce
Arabesque [49]	Single graph	Frequent subgraphs, cliques and motif counting	Giraph
HADI [50]	Graph database	Diameter of each graph	MapReduce
Zhao et al.'s approach [51]	Graph database	Eigenvalue of each graph	MPI/OpenMP
MRPF [9]	Single graph + subgraph model	Frequent subgraphs	MapReduce
Luo et al.'s approach [11]	A graph database	Frequent subgraphs	MapReduce
Hill et al.'s approach [10]	A graph database + subgraph model	Frequent subgraphs	MapReduce

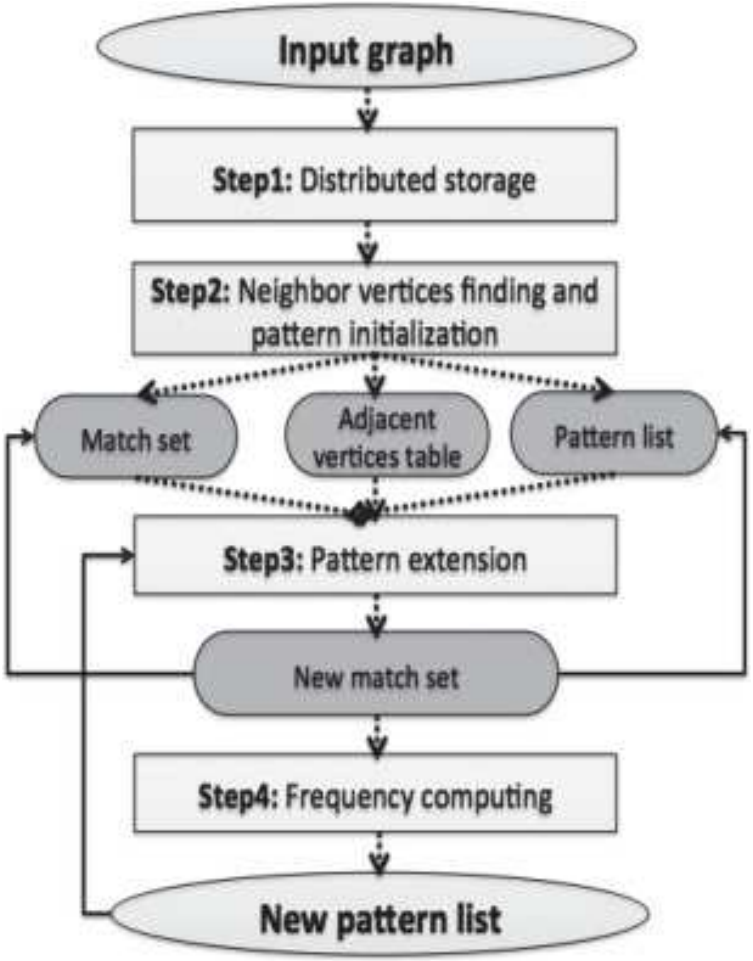
Big Graphs Analytics

MRPF (Liu *etal*., 2009)

Finding patterns from a complex and large network.
Four steps:

(1) distributed storage of the graph,
(2) neighbor vertices finding and pattern initialization,
(3) pattern extension, and
(4) frequency computing.

Each step is implemented by a MapReduce pass


Input graph
Step1: Distributed storage
Step2: Neighbor vertices finding and pattern initialization
Match set
Adjacent vertices table
Pattern list
Step3: Pattern extension
New match set
Step4: Frequency computing
New pattern list

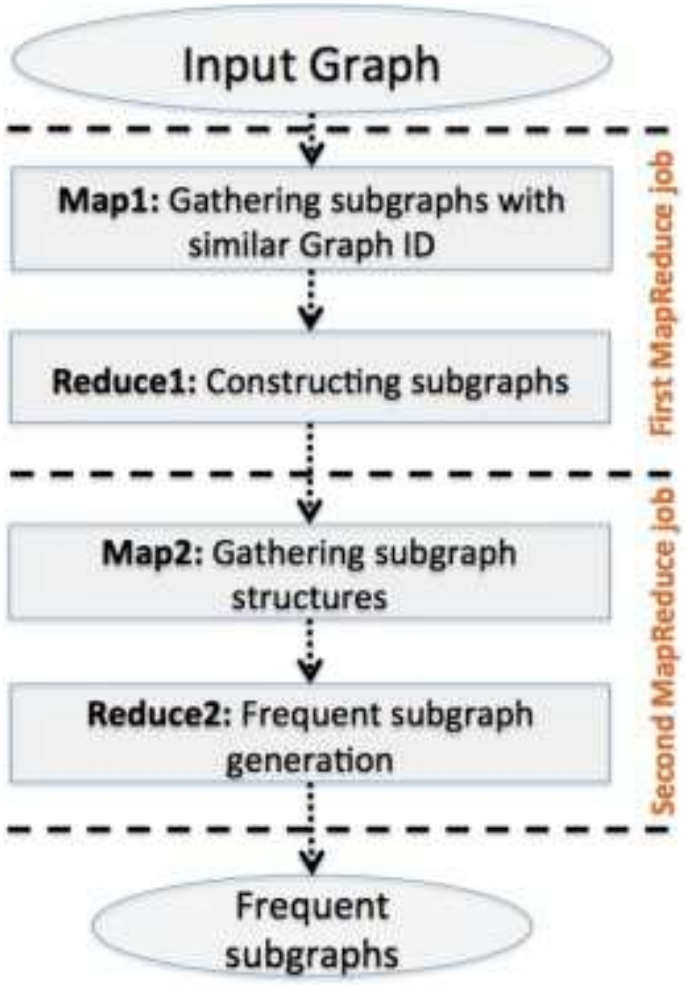
Big Graphs Analytics

Hill *etal.*'s approach (2012)

An iterative MapReduce-based approach for frequent subgraph mining

Generates the set of frequent subgraphs by performing two heterogeneous MapReduce jobs per iteration:

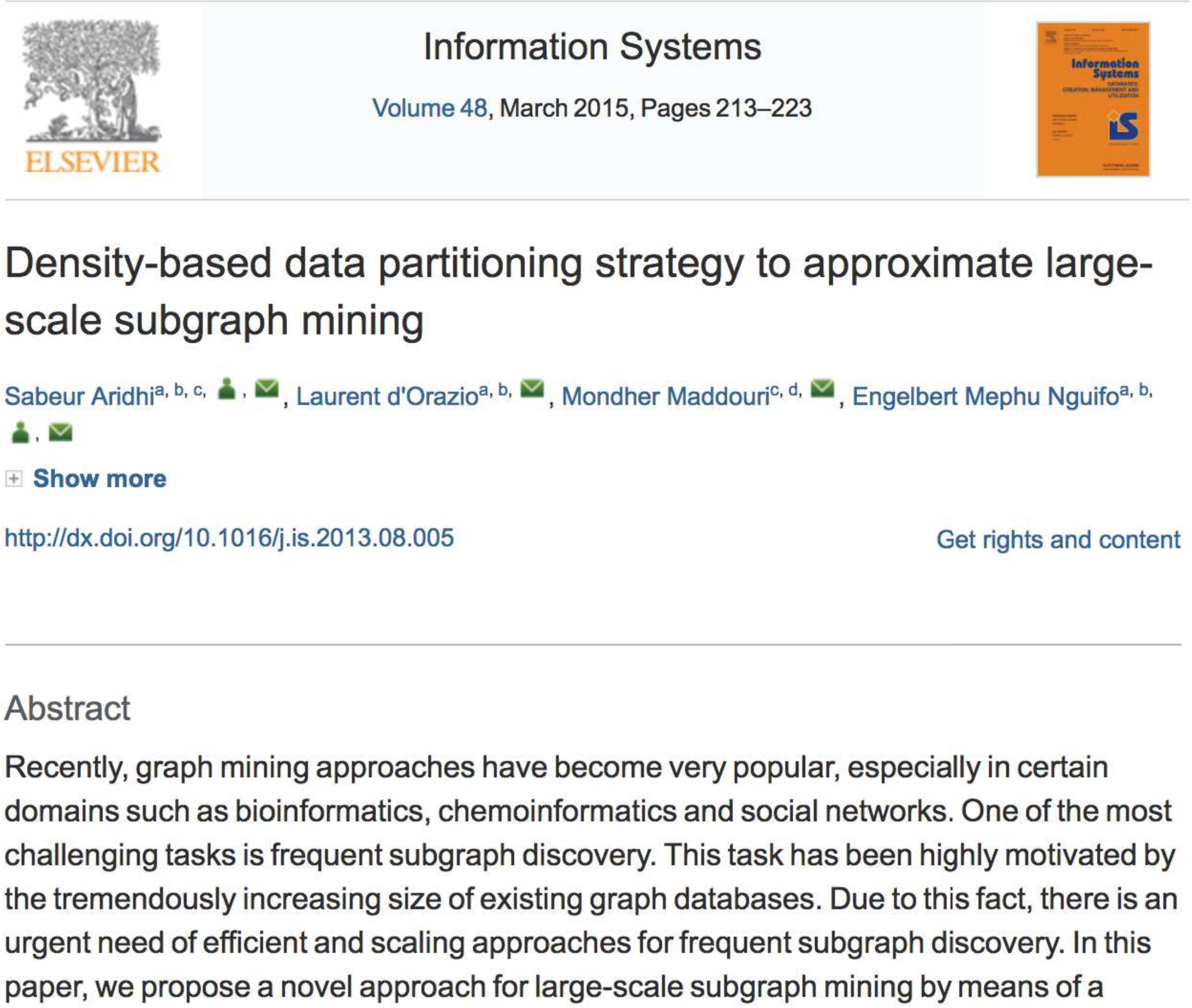
(1) gathering subgraphs for the construction of the next generation of subgraphs, and
(2) counting these structures to remove irrelevant data.


Input Graph
Map1: Gathering subgraphs with similar Graph ID
Reduce1: Constructing subgraphs
First MapReduce job
Map2: Gathering subgraph structures
Reduce2: Frequent subgraph generation
Second MapReduce job
Frequent subgraphs

Outline

- Graphs and graph mining
- Big data frameworks/analytics
- Big Graph frameworks/Analytics
- Two contributions
- Conclusion

Contribution 1



ELSEVIER

Information Systems

Volume 48, March 2015, Pages 213–223

Density-based data partitioning strategy to approximate large-scale subgraph mining

Sabeur Aridhi[a, b, c], 👤, ✉, Laurent d'Orazio[a, b], ✉, Mondher Maddouri[c, d], ✉, Engelbert Mephu Nguifo[a, b], 👤, ✉

⊞ Show more

http://dx.doi.org/10.1016/j.is.2013.08.005

Get rights and content

Abstract

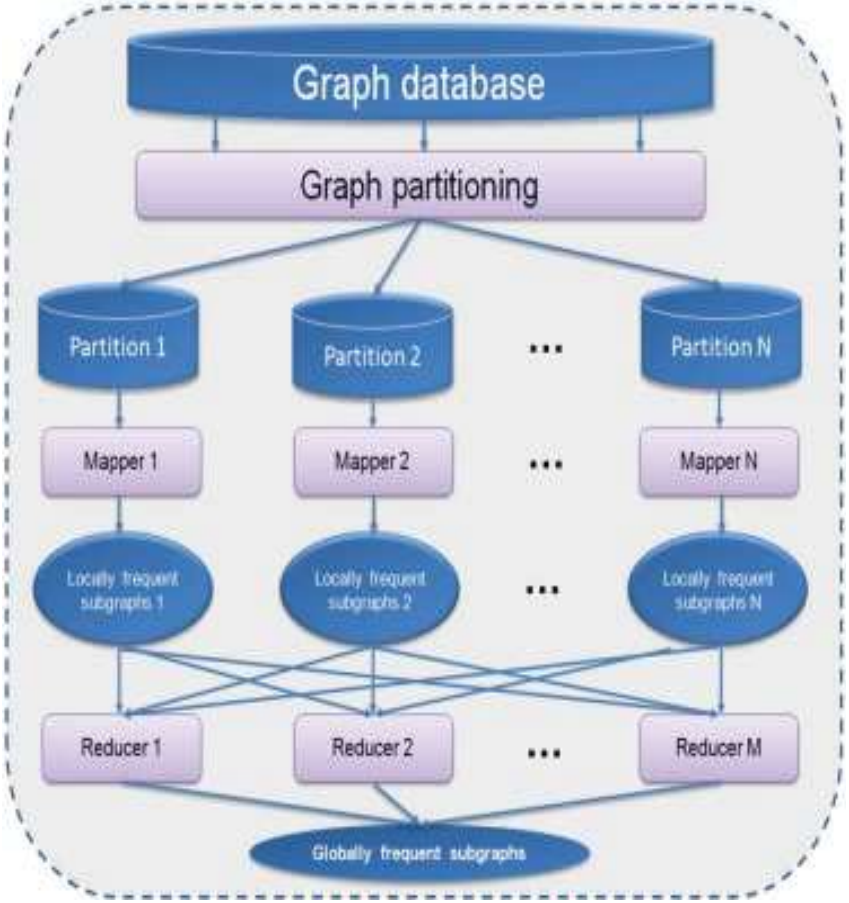
Recently, graph mining approaches have become very popular, especially in certain domains such as bioinformatics, chemoinformatics and social networks. One of the most challenging tasks is frequent subgraph discovery. This task has been highly motivated by the tremendously increasing size of existing graph databases. Due to this fact, there is an urgent need of efficient and scaling approaches for frequent subgraph discovery. In this paper, we propose a novel approach for large-scale subgraph mining by means of a

Big Graph Analytics

Aridhi et al.'s approach

Three steps:

1. Input graph database is partitioned into N partitions.
2. Mapper i reads the assigned data partition and generates the corresponding locally frequent subgraphs
3. The reducer computes for each subgraph its support in the whole graph database. Then, it outputs the set of globally frequent subgraphs

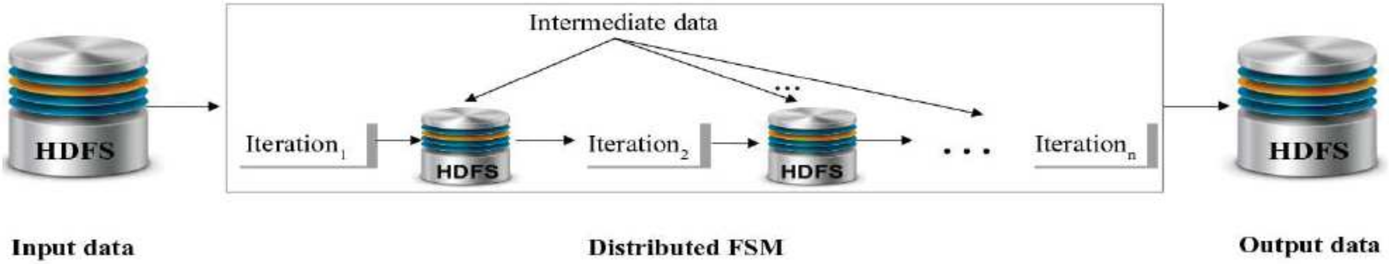


Big Graph Analytics

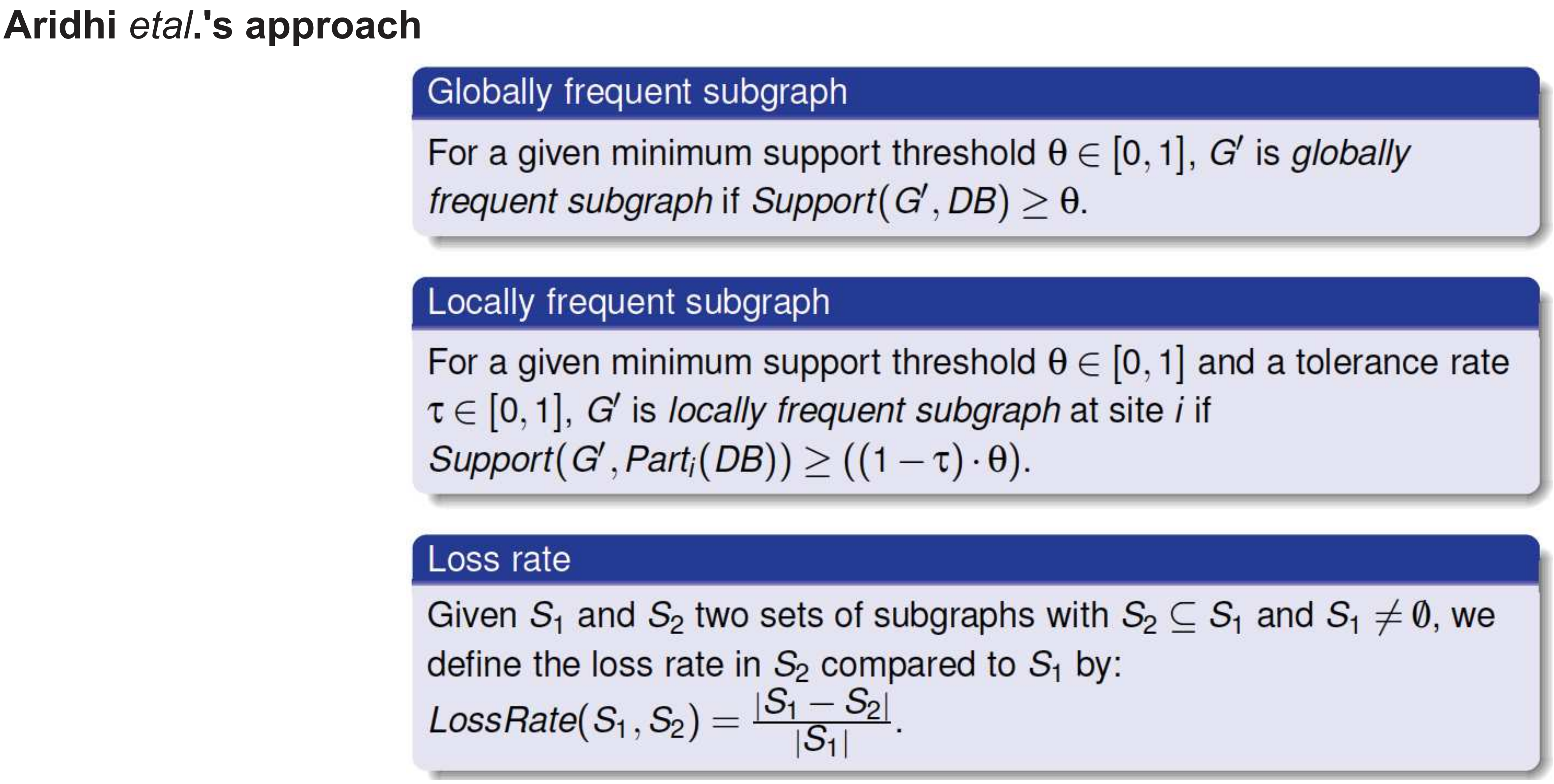
Aridhi *etal*.'s approach

In our work

- We focus on distributed FSM techniques from large graph databases.
- Two crucial problems with existing approaches:
  1. No data partitioning according to data characteristics.
  2. Construct the final set of frequent subgraphs iteratively.



Big Graph Analytics



Aridhi *etal.*'s approach

Globally frequent subgraph

For a given minimum support threshold $\theta \in [0,1]$, G' is *globally frequent subgraph* if $Support(G', DB) \geq \theta$.

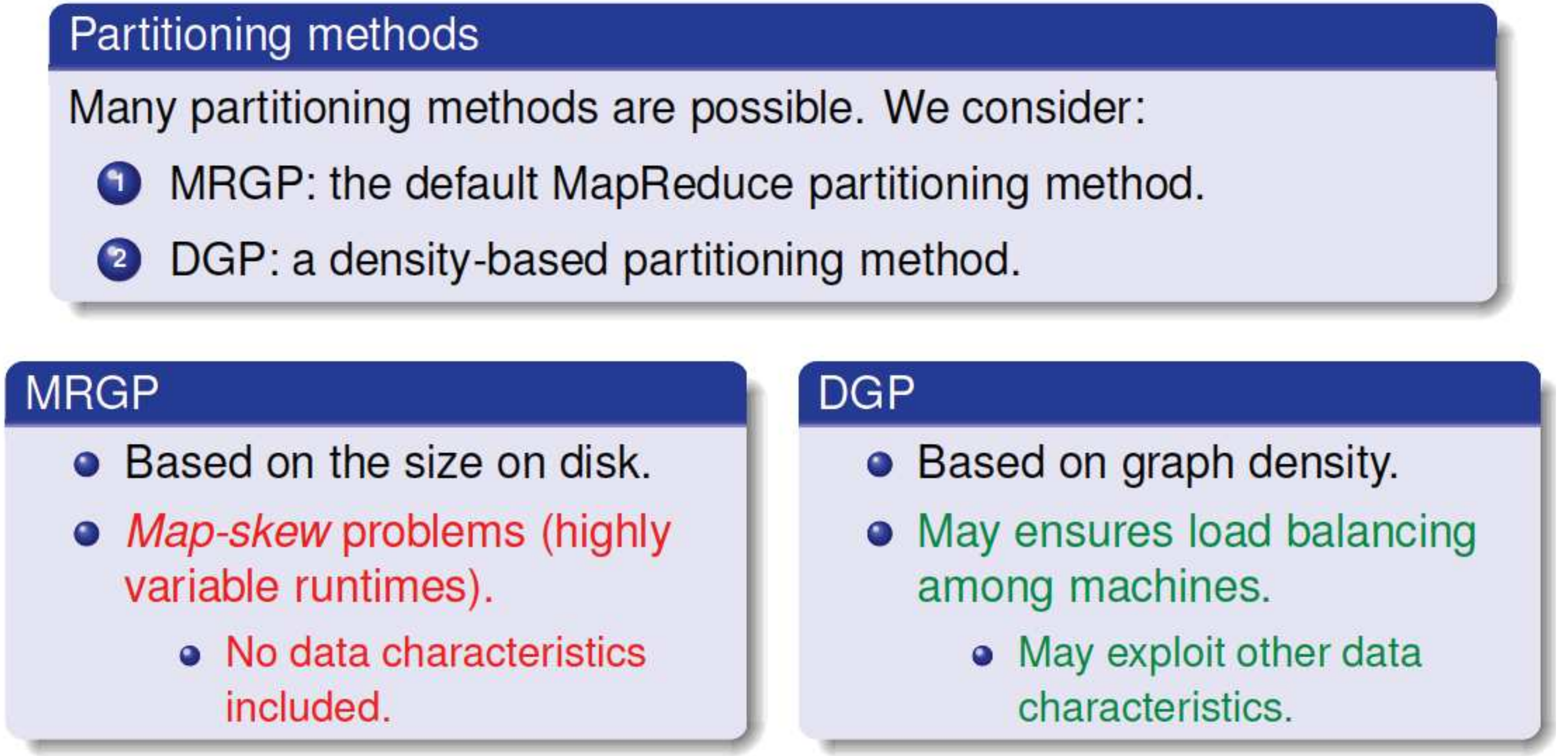
Locally frequent subgraph

For a given minimum support threshold $\theta \in [0,1]$ and a tolerance rate $\tau \in [0,1]$, G' is *locally frequent subgraph* at site i if $Support(G', Part_i(DB)) \geq ((1-\tau) \cdot \theta)$.

Loss rate

Given S_1 and S_2 two sets of subgraphs with $S_2 \subseteq S_1$ and $S_1 \neq \emptyset$, we define the loss rate in S_2 compared to S_1 by: $LossRate(S_1, S_2) = \frac{|S_1 - S_2|}{|S_1|}$.

Aridhi *etal*.'s approach



Partitioning methods

Many partitioning methods are possible. We consider:

1. MRGP: the default MapReduce partitioning method.
2. DGP: a density-based partitioning method.

MRGP

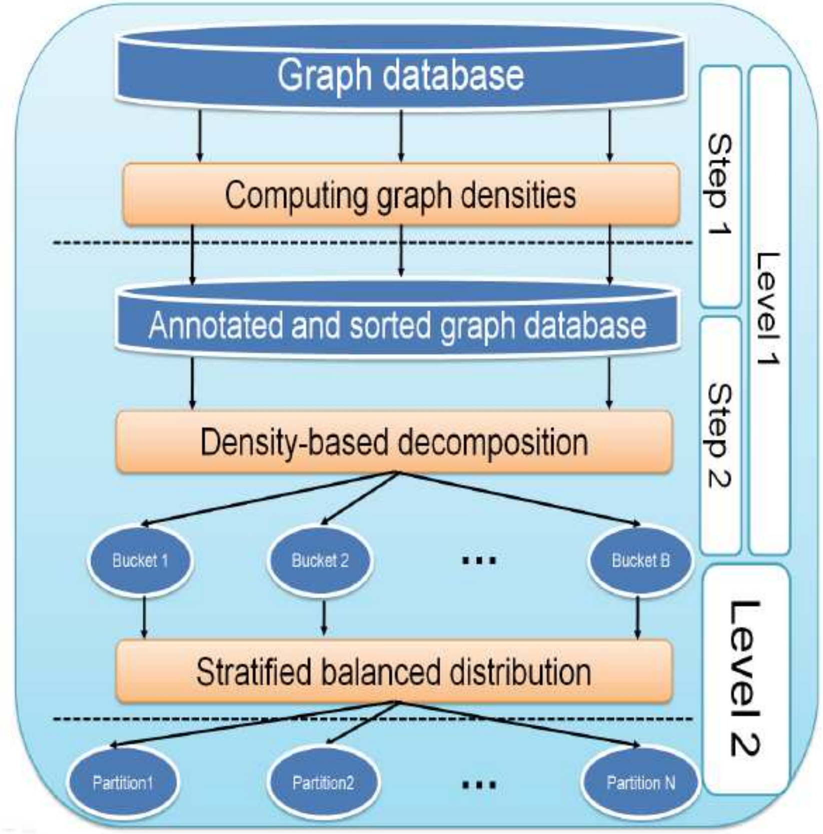
- Based on the size on disk.
- *Map-skew* problems (highly variable runtimes).
  - No data characteristics included.

DGP

- Based on graph density.
- May ensures load balancing among machines.
  - May exploit other data characteristics.

Big Graph Analytics

Aridhi *etal*.'s approach


Graph database
Computing graph densities
Step 1
Level 1
Annotated and sorted graph database
Step 2
Density-based decomposition
Bucket 1
Bucket 2
...
Bucket B
Stratified balanced distribution
Level 2
Partition1
Partition2
...
Partition N

DGP overview

Two-levels approach:

1. Dividing the graph database into *B* buckets,
2. Constructing the final list of partitions.

Aridhi *etal*.'s approach

Distributed FSM step

- A single MapReduce job.
  - **Input:** a set of partitions.
  - **Output:** the set of globally frequent subgraphs.

In the Mapper machine

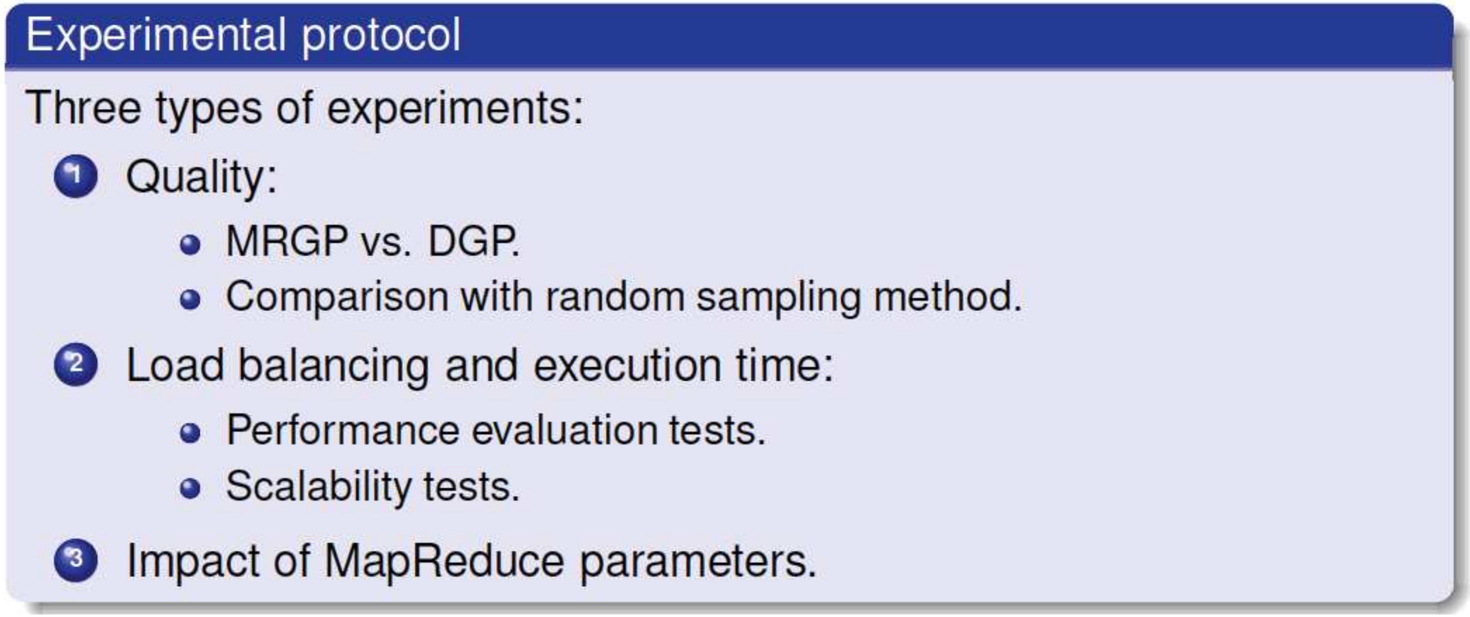
- We run a subgraph mining technique on each partition in parallel.
- Mapper *i* produces a set of locally frequent subgraphs.
  - Pairs of $\langle s, Support(s, Part_i(DB))\rangle$.

In the Reducer machine

- We compute the set of globally frequent subgraphs
  - Pairs of $\langle s, Support(s, DB)\rangle$.
  - No false positives generated.

Big Graph Analytics

Aridhi et al.'s approach

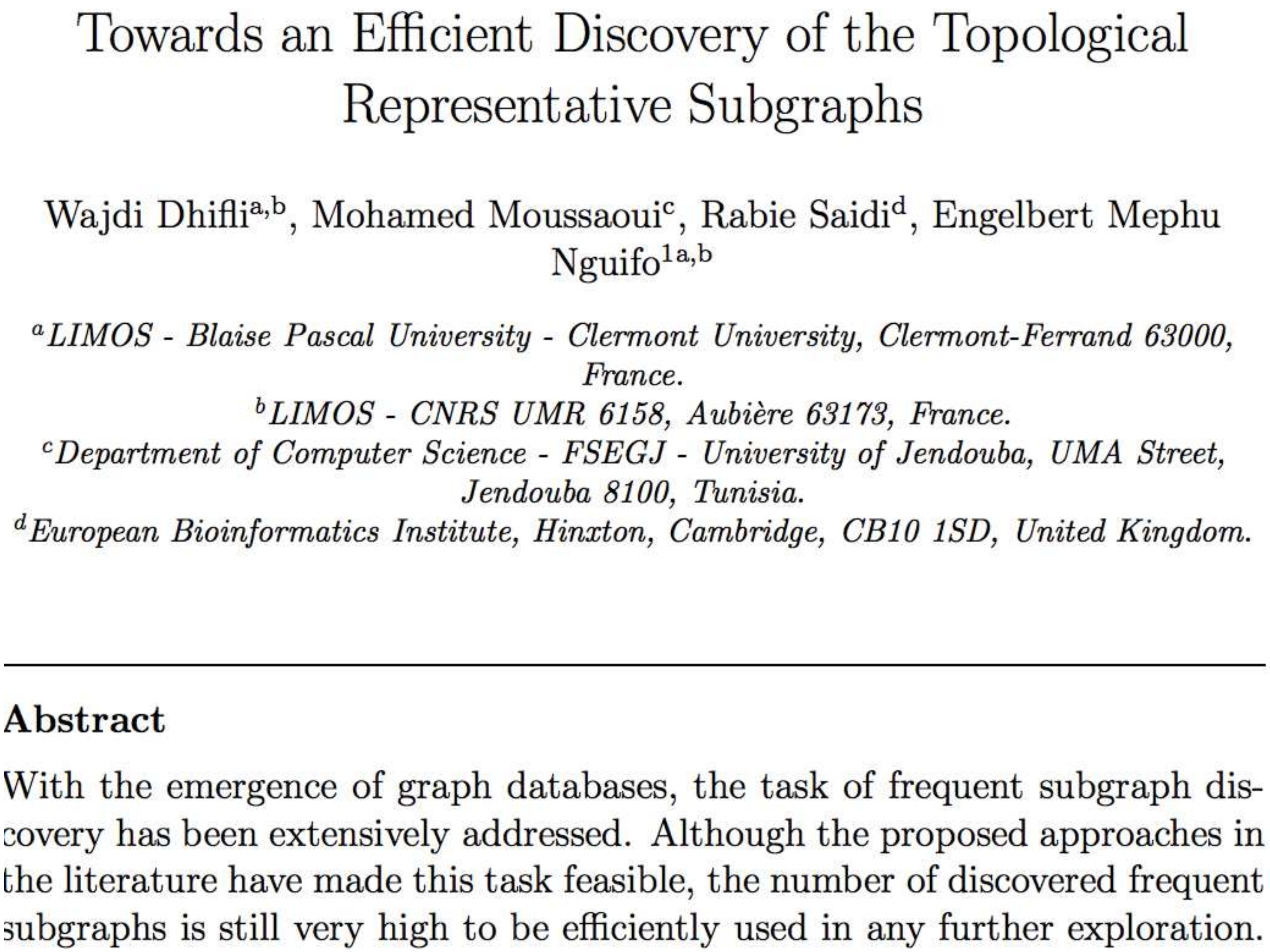


Experimental protocol

Three types of experiments:

1. Quality:
   - MRGP vs. DGP.
   - Comparison with random sampling method.
2. Load balancing and execution time:
   - Performance evaluation tests.
   - Scalability tests.
3. Impact of MapReduce parameters.

Contribution 2



Towards an Efficient Discovery of the Topological Representative Subgraphs

Wajdi Dhifli[a,b], Mohamed Moussaoui[c], Rabie Saidi[d], Engelbert Mephu Nguifo[1a,b]

[a] *LIMOS - Blaise Pascal University - Clermont University, Clermont-Ferrand 63000, France.*
[b] *LIMOS - CNRS UMR 6158, Aubière 63173, France.*
[c] *Department of Computer Science - FSEGJ - University of Jendouba, UMA Street, Jendouba 8100, Tunisia.*
[d] *European Bioinformatics Institute, Hinxton, Cambridge, CB10 1SD, United Kingdom.*

Abstract

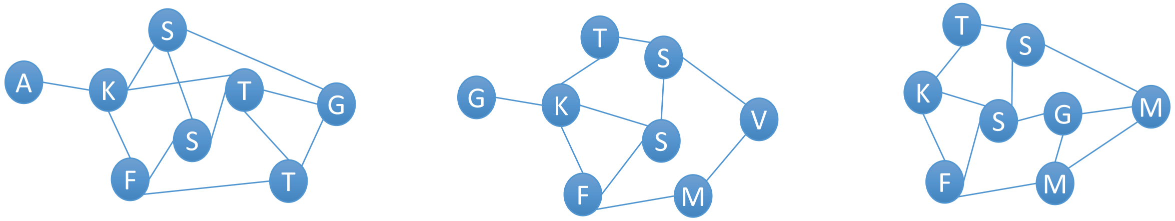
With the emergence of graph databases, the task of frequent subgraph discovery has been extensively addressed. Although the proposed approaches in the literature have made this task feasible, the number of discovered frequent subgraphs is still very high to be efficiently used in any further exploration.

Frequent subgraph discovery

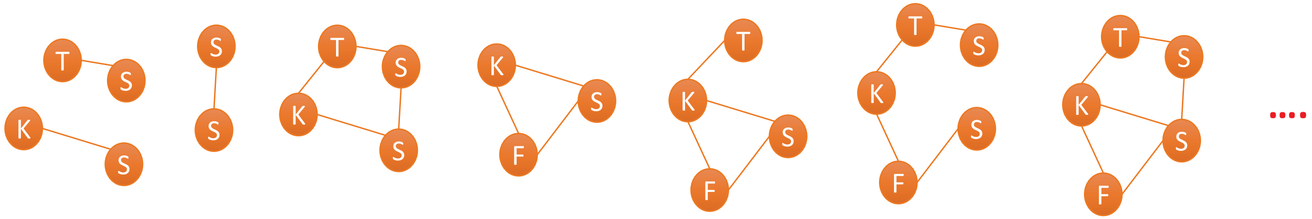
Goal:

- Finding subgraphs that occur in graph data, giving a minimum support

Example:



Graph database



Frequent Subgraphs
(minimum support = 3)

Frequent subgraph discovery

Existing approaches:

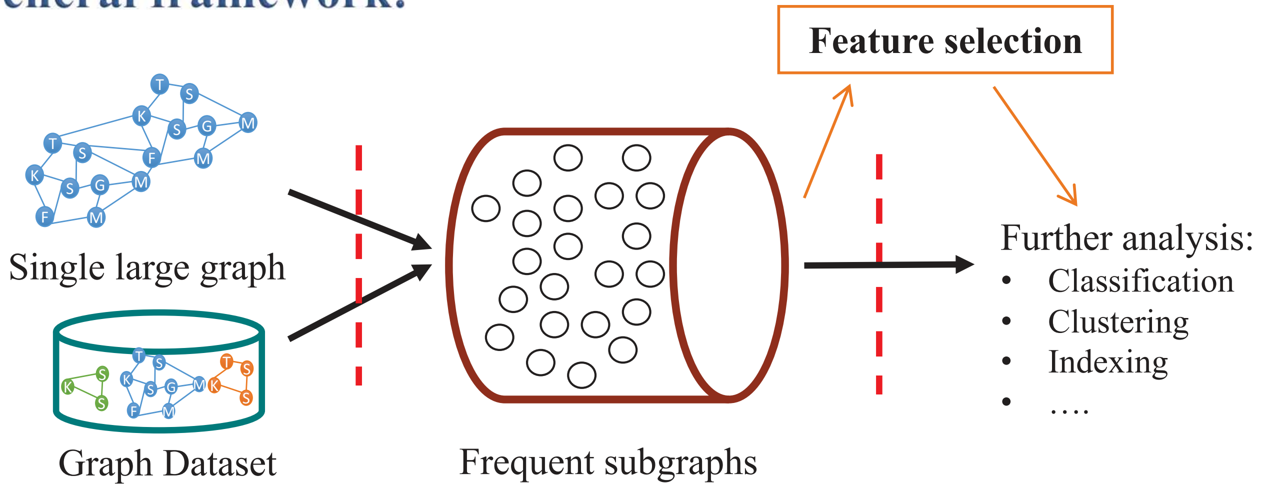
- **Greedy search**
  - Subdue (KDD 1994, OSDM 2005), GBI (AI 1994, PAKDD 2005), …
- **Inductive logic programming**
  - WARMR (KDD 1998), FARMER (IJCAI 2001), …
- **Apriori based**
  - AGM/AcGM (PKDD 2000), FFSM (ICDM 2003), …
- **Pattern growth based**
  - gSpan (ICDM 2002), Gaston (KDD 2004), …

Isomorphism remains a big challenge (NP)

But in practice: feasible in reasonable time

Frequent subgraph discovery

General framework:



Feature selection

Single large graph

Graph Dataset

Frequent subgraphs

Further analysis:

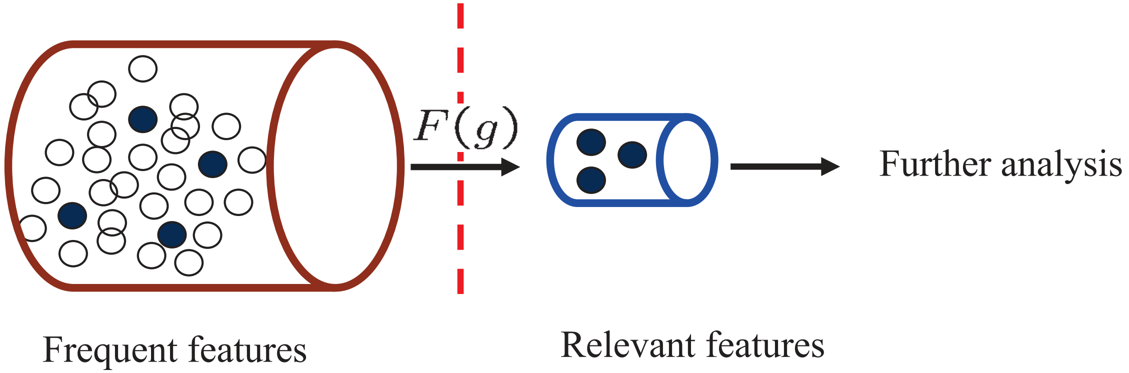
- Classification
- Clustering
- Indexing
- ….

Main issues:

- **Information overload**: Exponential number of subgraphs
  - ➡ Curse of **dimensionality**
- Relevance of discovered subgraphs: **redundancy** due to **structural** / **semantic** similarities

Feature selection

General framework:



Aims:

- Decrease the exponential number of features by removing redundant and irrelevant ones
- Enhancing (or at least maintaining) the quality of the feature set

Feature selection

Existing feature selection approaches for subgraphs:

- **Top-k and Clustering-based**
    - Redundancy-aware top-k patterns (KDD 2006), RING (ICDM 2009), TGP (ADMA 2010), ...
- **Sampling-based**
    - ORIGAMI (ICDM 2007), MCSs (ML 2011), …
- **Approximation-based**
    - Smoothing-clustering (CIKM 2008), Approximate mining with label cost (KDD 2013), …
- **Discriminative**
    - Leap (SIGMOD 2008), gPLS (KDD 2008), COM (CIKM 2009), LPGBCMP (KDD 2010), …
- **Other constraints-based**
    - SkyGraph (DMKD 2008), MIPs (ECML-PKDD 2009), Ant-motif (JOBIM 2012), …

Feature selection

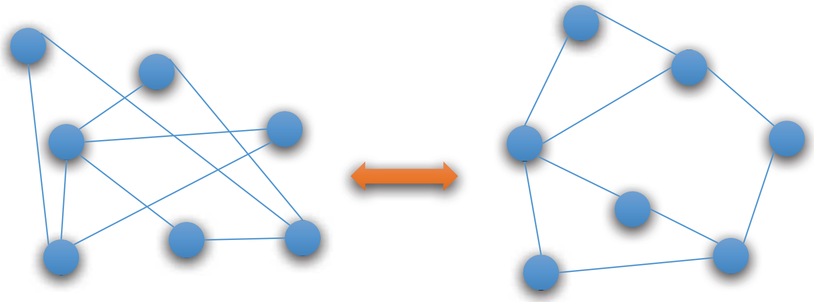
Existing feature selection techniques for subgraphs:

- Perform isomorphism test => **computational cost** !!!
- Slight structural differences **do not** matter in many applications!
- Do not allow targeting **a particular** structural property?
- Do not consider **hidden** similarities: diameter, density, clustering coefficient, …

Topological Representative Subgraphs

Idea:

- Structurally similar subgraphs have similar topological properties



Topological Representative Subgraphs

Graph Classification via Topological and Label Attributes

Geng Li, Murat Semerci[†], Bülent Yener, and Mohammed J. Zaki
Rensselaer Polytechnic Institute, Troy, NY
[†]Bogazici University, Istanbul, Turkey
{lig2,yener,zaki}@cs.rpi.edu, semercim@gmail.com

1. Number of nodes
2. Number of edges
3. Average degree
4. Density
5. Average clustering coefficient
6. …

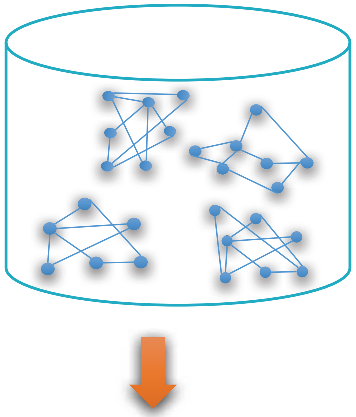
Complexity :
O(1) ou O(n + m) …….. O(n^2)

Subgraphs : Small size, Sparse, …

Topological Representative Subgraphs

Graph properties

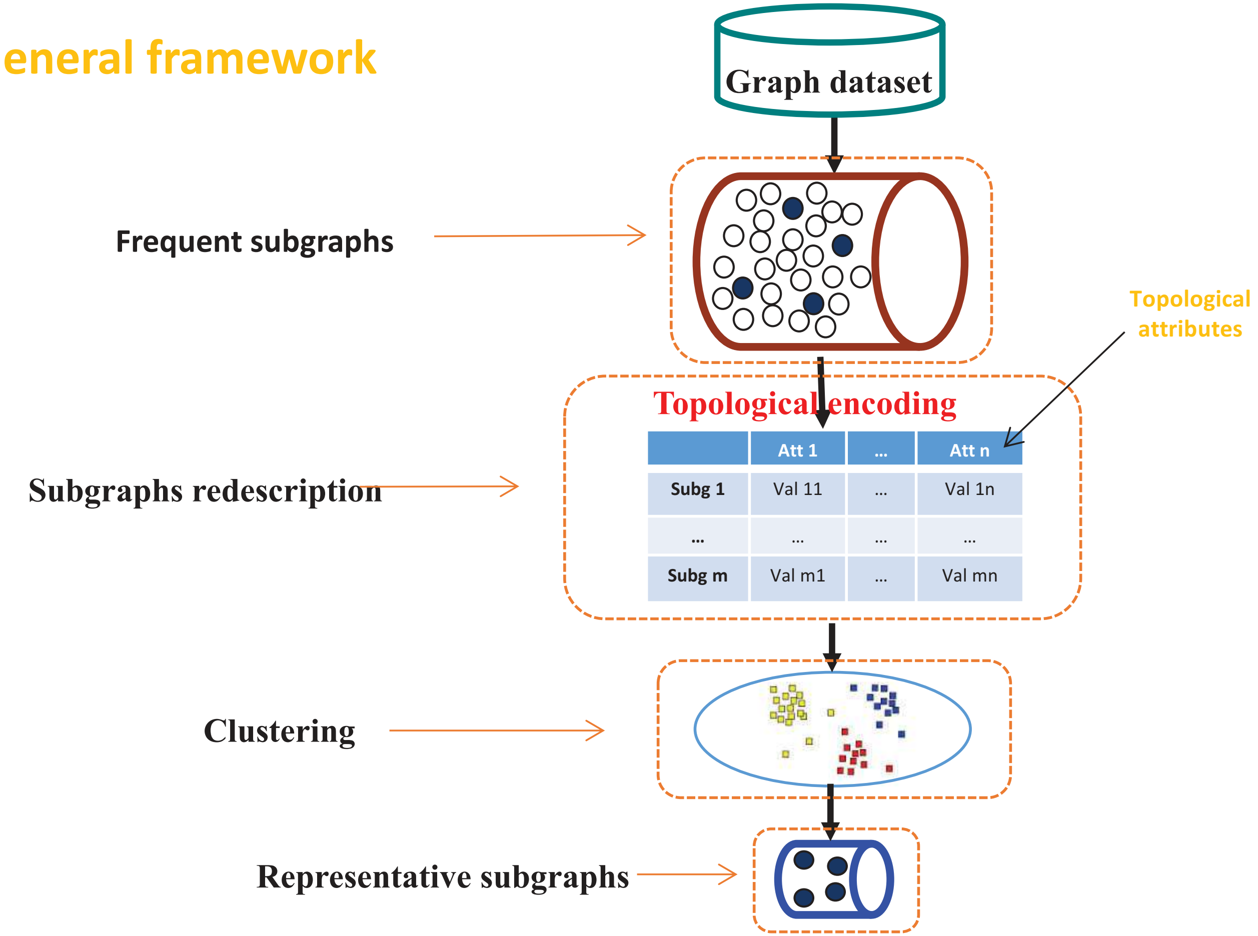
1. Number of nodes
2. Number of edges
3. Average degree
4. Density
5. Average clustering coefficient
6. …



Subg \ Attr	Attribute 1	…	Attribute n
Subg 1	Val 11	…	Val 1n
…	…	…	…
Subg m	Val m1	…	Val mn

Topological Representative Subgraphs

General framework



	Att 1	...	Att n
Subg 1	Val 11	...	Val 1n
...	...	...	...
Subg m	Val m1	...	Val mn

Outline

- Graphs and graph mining
- Big data frameworks/analytics
- Big graph frameworks/analytics
- Two contributions
- Conclusion

Conclusion

We need Big Graph Analytics

- Survey of *main* frameworks and techniques
- Not exhaustive
- not a deep/experimental comparison between tools

- Many tools are still in progress ….