An overview of Graph Categories and Graph Primitives

Dino Ienco (**dino.ienco@irstea.fr**)

https://sites.google.com/site/dinoienco/

Topics I'm interested in:

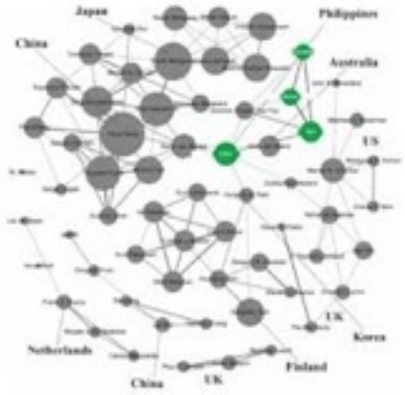
- Graph Database and Graph Data Mining
- Social Network Analysis
- Semi-Supervised and Clustering
- Deep Learning
- Data Stream

Overview

What is a **Graph** and Why Graphs are useful nowadays?

Type of graphs:

- Directed vs Undirected
- Directed Acyclic Graph (DAG)
- Labeled Graph: Node and Edge Labeled
- Hypergraph
- Multigraph



Graph-Based approaches and methodologies:

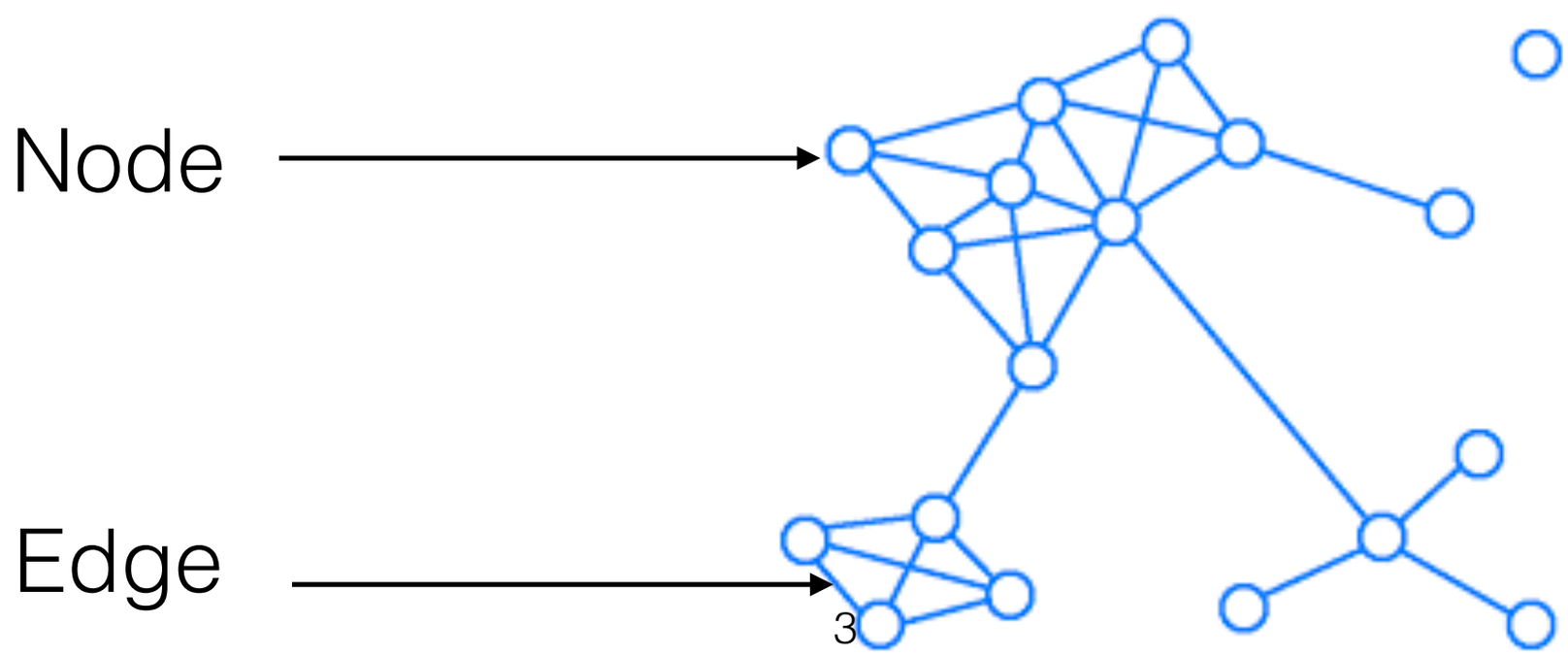
- Graph Data Management
- Graph Mining
- (Social) Network Analysis
- Knowledge Graphs
- Graph Visualization

What is a Graph and Why is useful

A graph is a **mathematical structure** (and a computer science abstract data type)

A graph is generally composed by a **set of Vertices** and a **set of Edges:** G = (V,E)

An edge is a pair of nodes that are linked each other



What is a Graph and Why is useful

Graph can be augmented with layer or attribute information

Why graphs are useful nowadays:

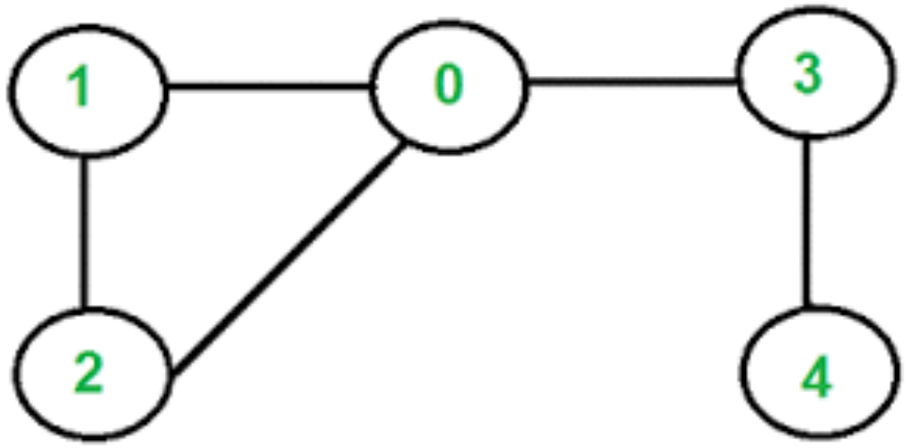
- Many real world data can be represented by this structure
- It allows to represent at the same time **instances (nodes)** and **instances relation (edges)**
- We can associate additional information to each instances

Graphs are expressive structures that allow to model complex information

Type of Graphs

Undirected Graphs:

Graphs in which edges between two nodes do not have direction, edge (1,2) is equal to edge (2,1)



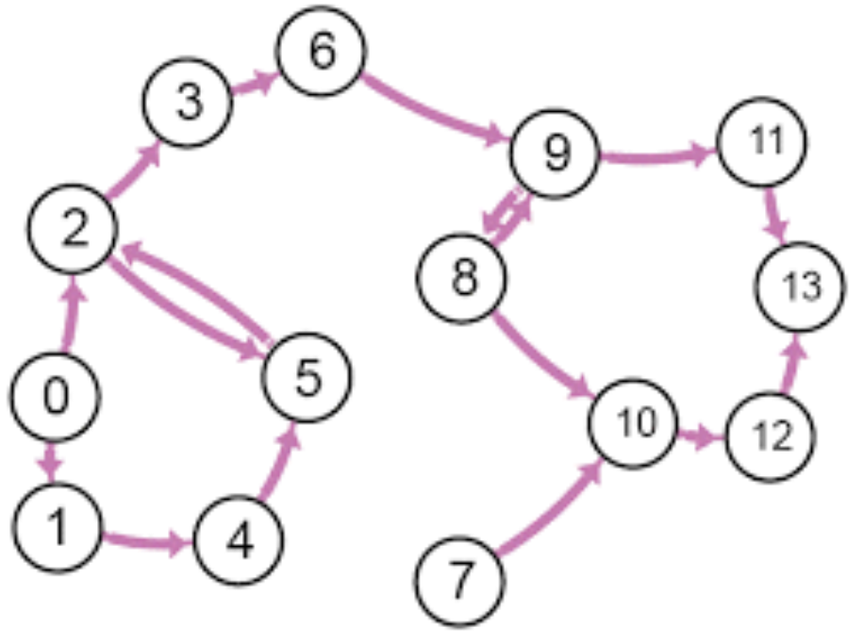
This kind of graphs are usually employed to represent **standard relations** without precedence:

- Biological Networks
- Social Networks
- Interaction Networks
- Co-Authorship Networks
- Image Segmentation Networks

Type of Graphs

Directed Graphs:

Graphs in which edges have direction, edge (1,2) is different from edge (2,1)



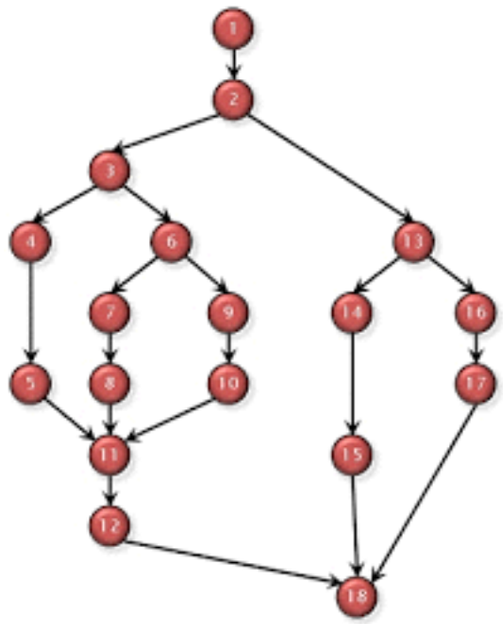
This kind of graphs are usually employed to represent **causality or information propagation**:

- Biological Networks
- Social Networks
- Communication Networks
- Co-Authorship Networks
- Citation Networks

Type of Graphs

Directed Acyclic Graphs:

A particular class of Directed Graph in which there is no cycle (we cannot start and end a graph traversal from and to the same node)



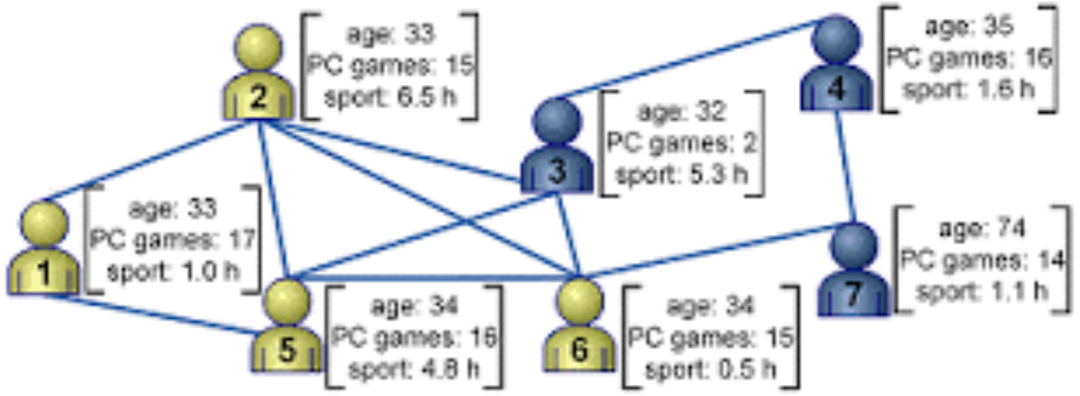
This kind of graphs are usually employed to represent **workflows or describe temporal state evolutions**:

- Any kind of workflows
- Particular case of human (or animal) interactions (spread phenomena)

Type of Graphs

Attributed Graphs:

Directed (or Undirected) graphs that have a vector of information (or a set of items) associated to each node



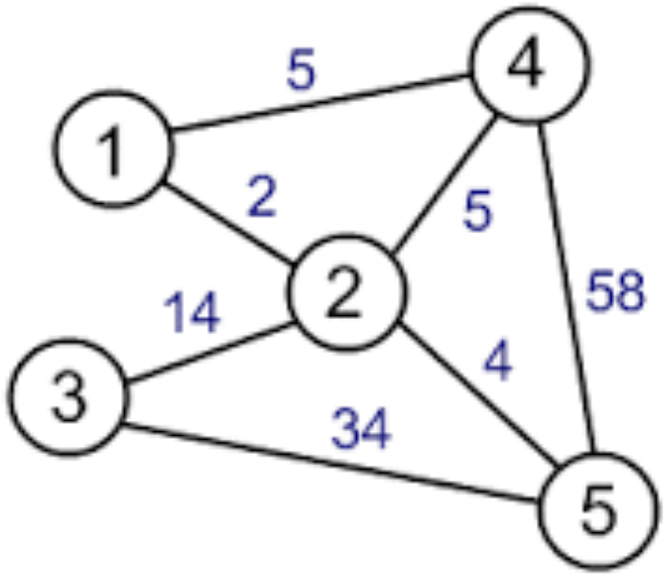
This kind of graphs are usually employed to represent **more complex information regarding the instances** (nodes) of the graph:

- Social Networks
- Any kind of domain that associate to each node additional information

Type of Graphs

Weighted (or edge-labeled) Graphs:

Directed (or undirected) graphs that have a numerical weight (or a discrete label) associate to each edge



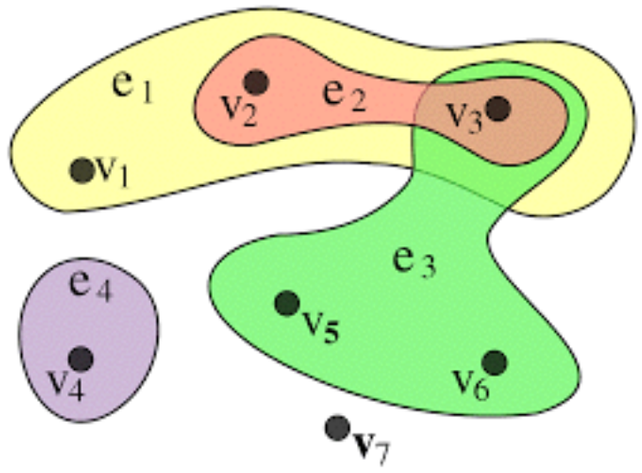
This kind of graphs allows to represent **edge information** such as the **strength** of the connection between two nodes or a **category** of edge:

- Social Networks
- Communication Networks
- Biological Networks
- Any kind of domain in which we are able to quantify the strength between instances (distances or similarities)

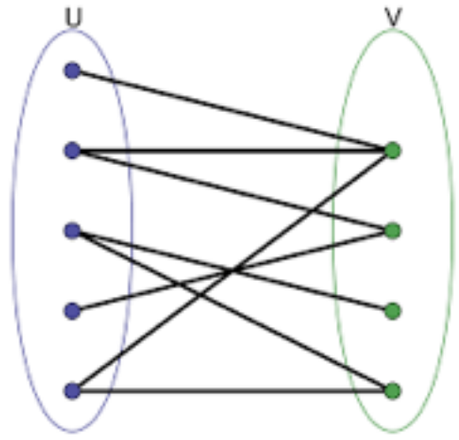
Type of Graphs

HyperGraphs:

Graphs in which an edge can link more than two nodes. A generalization of common graphs where an edge is a connection between two nodes



This kind of graphs allows to represent **complex interactions** among **more than two entities**:

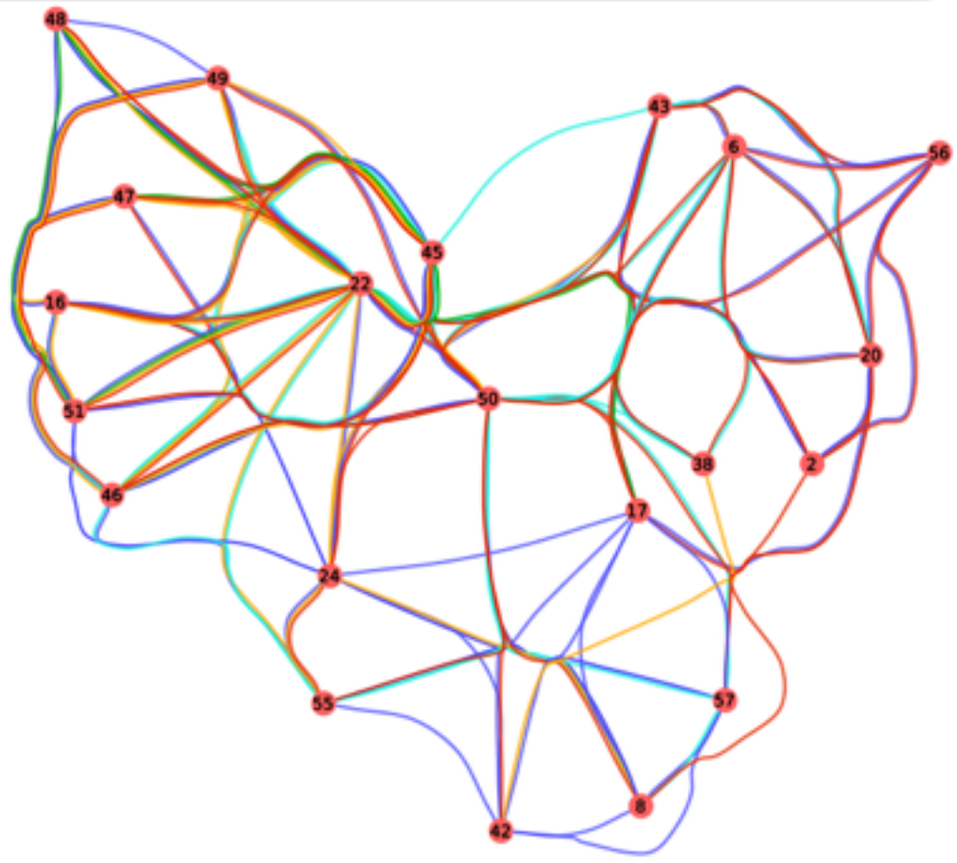


- Social Networks
- Co-authorship network
- Relational Databases
- Any kind of domain in which we need to manage n-ary relationships

Type of Graphs

MultiLayer Graphs:

Directed (or Undirected) graphs that have different layers (edge set). Each layer is associated to a different kind of interaction between nodes.



This kind of graphs allow to represent **multiple (semantically different) interactions** between two nodes:

- Social Networks (with different layers: social, family, work, etc)
- Co-Authorship networks (one layer for each conference)
- Biological networks (gene interactions with pathways spec.)
- Remote Sensing image networks (one layer for each spectral indicators)

Graph-Based approaches and Methodologies

Graph Data Management:

Techniques to store, index and query graph data

Graph Mining:

Techniques and approaches to extract interesting knowledge from graph

(Social) Network Analysis:

Techniques and approaches to supply information about the global network structure but also inspecting the role of the nodes in the network

Knowledge Graphs:

Knowledge bases that can be represented as graph. Metadata and data represented by RDF format or linked open data

Graph Visualization:

Techniques and approaches that can be used to perform Visual Analytics on network data

Graph Data Management

Techniques to **store, indexing** and **query** graph data

Work at database level in order to supply fast implementation of graph primitives and basic graph operations

Example of **Operations**:

- **Exact Subgraph Queries**
- **Approximate Subgraph Queries**
- **Reachability Queries**
- **Regular Path Queries**
- **Etc…**

Settings (How the graph database looks like?):

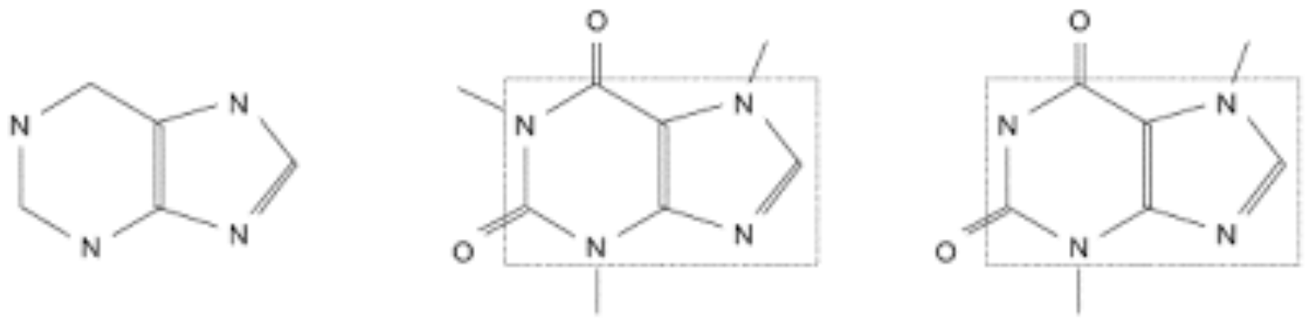
Single Large Graph (Social Networks, Biological Network, Ecological Network)

Collection of Graphs (Chemical networks, Human interaction patterns, Temporal road networks)

Graph Data Management

Subgraph Queries:

Given a query graph Q and a data graph G, find all the embeddings of Q in G



(a) Query (b) Caffeine (c) Thesal

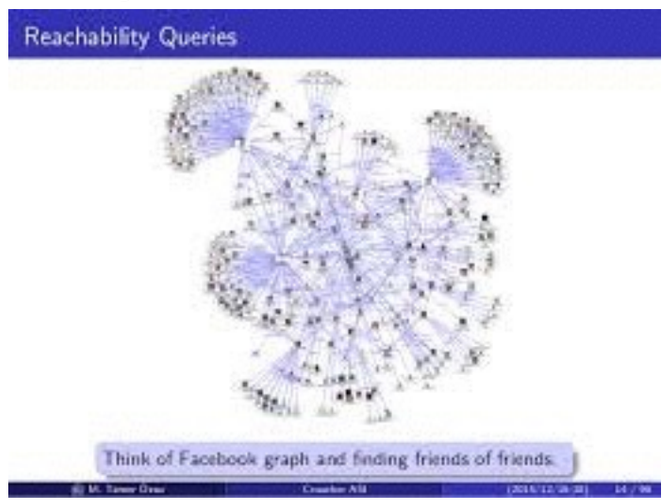
Applications:

Biochemical compounds matching,
Pattern matching in biological or social network,

Graph Database

Reachability Queries:

Given a data graph G and two nodes n1 and N2, verify if n1 (resp.n2) is reachable from n2 (resp. n1).



Applications:

Analyze Web graphs, Link analysis, Biological networks

Graph Mining Techniques

Techniques to **extract** interesting **knowledge** from graph

Exploit graph primitives in order to mine knowledge or patterns that can characterize the graph database

Example of **Techniques**:

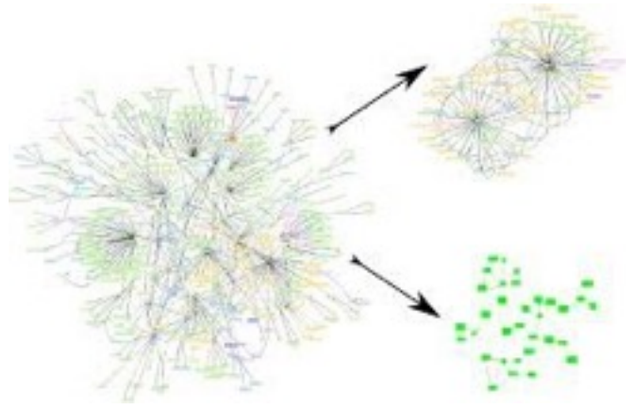
- **Frequent Pattern (Graph) Mining**
- **Enumerate Quasi-Cliques**
- **Graph Clustering**
- **Etc…**

Techniques are **available for both** graph database **settings** (Single / Collection) but, techniques for one setting do not directly work for another setting

Graph Mining Techniques

Frequent Graph Mining:

Given a data graph G, find all the patterns (subgraphs) that have a frequency greater than a certain threshold t

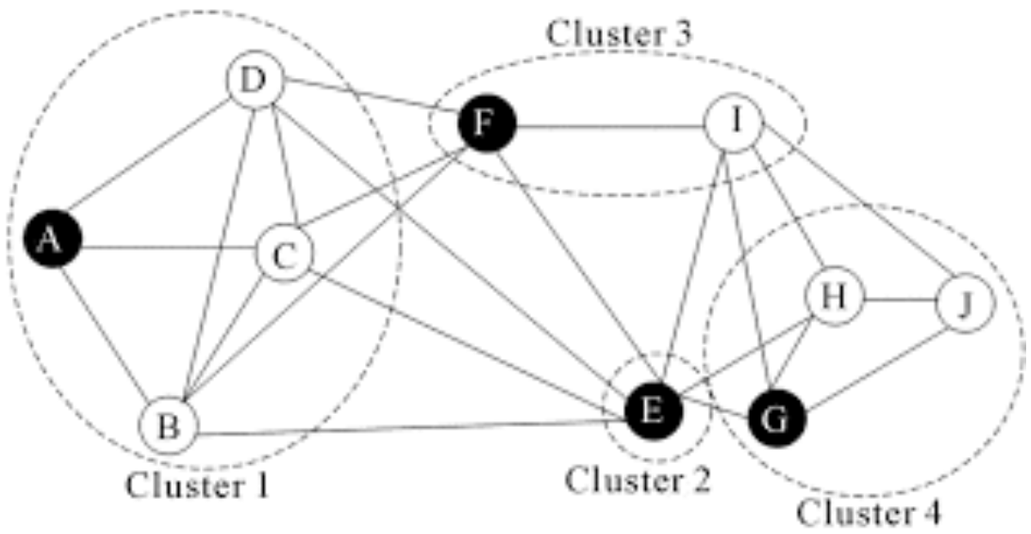


Applications:

Mine frequent behavior in any kind of network (biological, ecological, co-authorship, remote sensing image graph, social network)

Graph Clustering:

Given a data graph G, find k groups of nodes such that the nodes belonging to a group are structurally and semantically similar



Applications:

Network Partitions, Module identification in biological networks (gene-gene interactions)

(Social) Network Analysis

Techniques and approaches to supply information about the **global network structure** but also **inspecting the role of the nodes** in the network

Describe structural properties of the network at global and local scale

Example of **Techniques**:

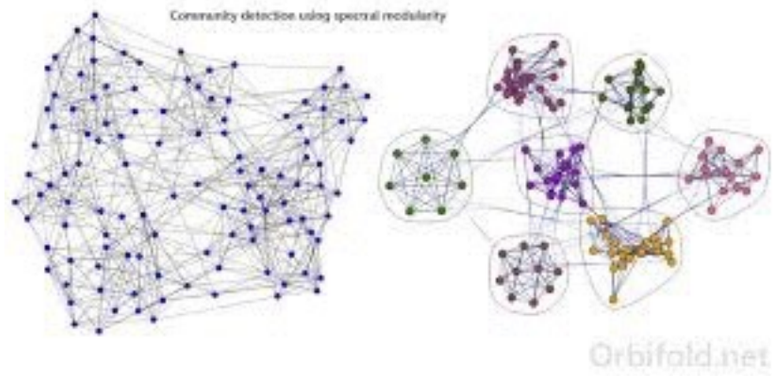
- **Community Detection**
- **Link Analysis**
- **Role Identification (hubs, outliers, etc..)**
- **Etc…**

Techniques are mainly exploited to analyze single large graphs since they were introduce in the context of social network analysis but, now, they are exploited to analyze any kind of network

(Social) Network Analysis

Community Detection:

Given a data graph G, find a set I of (overlapping) group such that each group contains nodes linked each other.

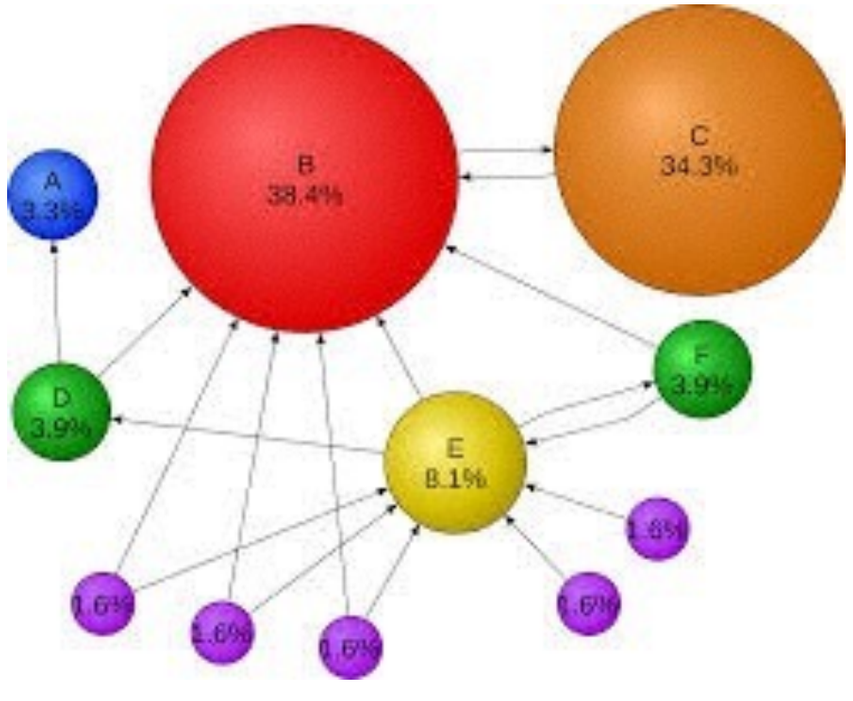


Applications:

Community extraction in Social Network, Extract plant group in Phytosociology Mine gene (or protein) modules in biological network

Link Analysis:

Given a data graph G, find a rank of the graph nodes related to their importance in the graph topology



Applications:

Leader identification in network, Node ranking by structural importance

Knowledge Graphs

Knowledge bases that can be represented as graph. Metadata and data represented by **RDF format** or **linked open data**

Useful to store data and Metadata on which reasoning can be done

Example of **Techniques**:

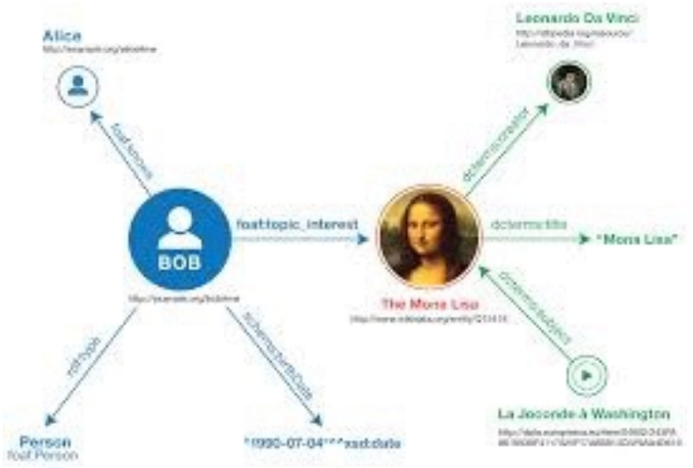
- **SPARQL Query**
- **Reasoning**
- **Etc…**

The knowledge (ontology) is represented as graph and can be navigated and browsed by logical operator and reasoning is possible

Knowledge Graphs

RDF format:

Data and Metadata are represented with triple (subject, predicate, object).



Applications:

Web of data representation language, Linked Open data, data publishing, metadata exchange and data description in any domain (social, biological, chemical, environmental, ecological)

SPARQL:

Language to query and manipulate graph RDF. The answer will be a set of triples that match the query variable and structure

```
PREFIX java: <http://evolizer.org/ontologies/seon/2009/06/java.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?url ?name
WHERE{
   ?url rdf:type java:Package .
   ?url rdfs:label ?name
}
```

Applications:

Knowledge based Question Answering, Knowledge-Based Reasoning

Graph Visualization

Techniques and Approaches that can be used to perform **Visual Analytics** on network/graph data

Techniques developed to visualize graph structure highlighting particular properties of these graphs.

Example of **Techniques**:

- **Vertex Positioning**
- **Edge Bundling**
- **Etc…**

These kind of techniques allow interaction to navigate and browse the visual graph structure and perform zoom-in/zoom-out operations. The general schema involves visual metaphors to abstract network properties

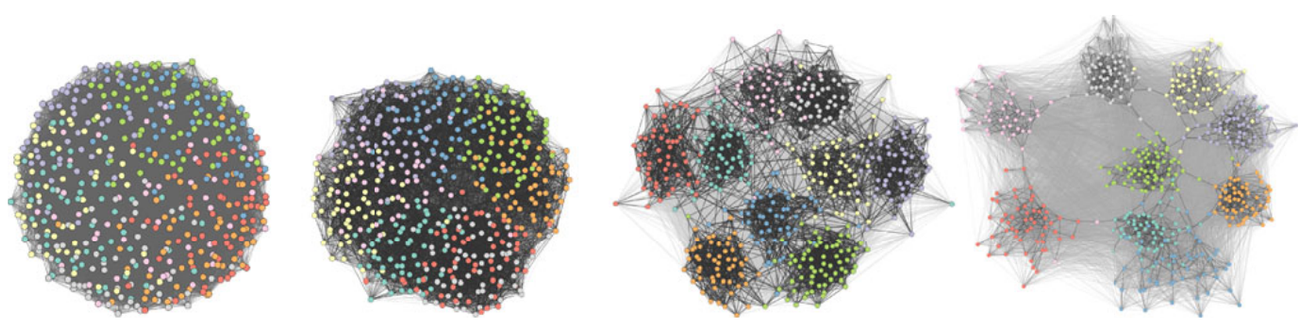
Graph Visualization

Vertex Positioning:

Given a graph G, locate the vertex V of the graph in a two (or three) dimensional space in roder to preserve topological properties and highlight graph specificity

Applications:

Graph Visualization, Visual inspection of any kind of graph



(b) 100% (c) 34% (d) 25% (e) 5%

Edge Bundling:

Given a graph G, draw graph edges to highlight graph specificities and improve the visual result

Applications:

Highlight common interaction patterns, improve visibility, support visual analysis of communities

Conclusions

Graphs are an **ubiquitous structure** to model many type of data

Many useful tool to **manage, analyze** and **visualize** graph data

The Data Science field is working heavily on graph-based approaches but sometimes **the different communities do not communicate each other**

Main issue

Each field (social network, biology, ecology, remote sensing, semantic web, etc..) has its own specificity and needs specific graph primitives

Conclusions

What about **Ecology** and **Graphs**?

Many tools already exist, a possible road map to follow:

1) Understand which kind of graph representation I need

2) Understand which are the analysis I want to do on my data (query, structure summarization, role identification, frequent structure mining, classification, etc..)

3) Search for algorithms or methods that can fit my needs

4) Deploy and Adapt such methods to my problem

5) Maybe…come back to point 1) to adjust my previous considerations

People I'm working with on Graph Analysis

MCF A. Sallaberry
Lirmm - France
(**Graph Visualization**)

MCF F. Ulliana
Lirmm - France
(**Knowlege Graph & Reasoning**)

PhD Student V. Ingalalli
Lirmm - France
(**Graph Data Management**)

Post-Doc R. Interdonato
Univ. Calabria - Italy
(**Social Network Analysis**)

Prof. Pascal Poncelet
Lirmm - France
(**Graph Data Mining**)

Post-Doc F. Nor-Guttler
Dynafor - France
(**Remote Sensing**)

Researcher A. Tagarelli
Univ. Calabria - Italy
(**Social Network Analysis**)

PhD Student L. Khiali
MTD - France
(**Data Mining**)

Thank you for your attention

Questions