

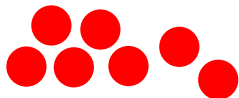
Cosa ci dicono le reti (in Python)

Pietro Battiston

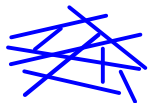
Linux Day 2019
26 ottobre, Pisa

Cos'è una rete?

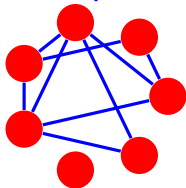
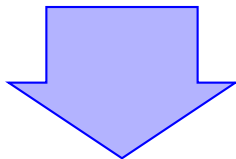
Ingredienti principali



Cose
~~Cose~~
Nodi



Conessioni
~~Conessioni~~
Spigoli



Python: varie alternative

(Vantaggi e svantaggi dipendono dal ciò che si vuole fare!)

- ▶ [networkx](#): semplice da usare (pure Python), non molto efficiente
- ▶ [graph-tool](#): efficiente (C++), affermato
- ▶ [igraph](#): efficiente (C), un poco meno di graph-tool, disponibile anche per R
- ▶ [networkit](#): efficiente (C), forse più di graph-tool, progetto giovane

Diamo un occhio a [networkx](#) and [graph-tool](#).

Primi passi con networkx

Creare una rete (Non-diretta, di default!)

```
import networkx as nx

g = nx.Graph()

g.add_node(0)           # Un oggetto Python (hashable)
g.add_nodes_from(['a', 'this', 42])

g.add_edge(0, 42)
g.add_edge(42, 43) # Aggiunto automaticamente
```

Creare una rete diretta

```
g = nx.DiGraph()
```

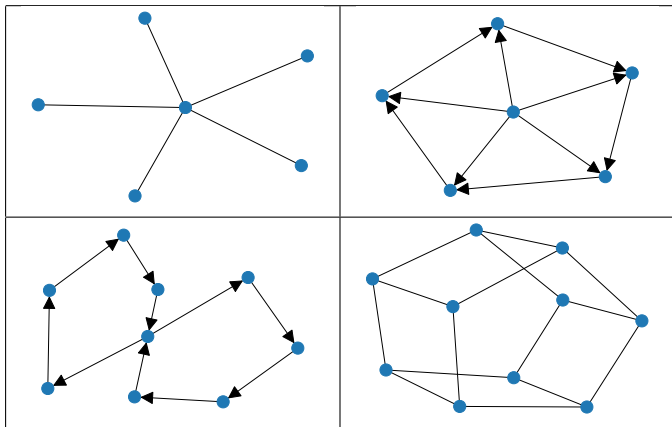
Disegnare una rete

```
nx.draw(g)
```

Utile:

```
nx.draw(g, with_labels=True)
```

Esempi



Getting started with graph-tool

Creare una rete (diretta, di default!)

```
from graph_tool import Graph

g = Graph()

n = g.add_vertex()      # I nodi hanno indici, non nomi
ns = g.add_vertex(10)  # ... aggiungiamone 9

g.add_edge(42, 43)     # Aggiunge automaticamente (tutti)
g.add_edge_list([(2, 3), (7, 8)])
```

Creare una rete non-diretta

```
g = Graph(directed=False)
```

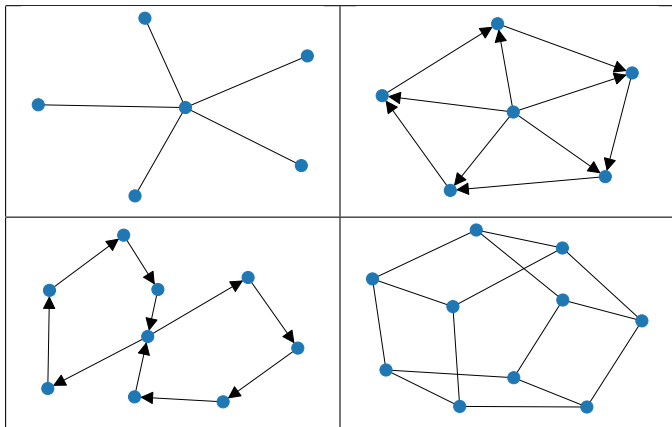
Disegnare una rete

```
from graph_tool.draw import graph_draw
graph_draw(g)
```

Utile:

```
graph_draw(g, vertex_text=g.vertex_index)
```

Esempi



Sei gradi di separazione?

Domanda di ricerca con una lunga storia (Karinthy, 1929; Milgram, 1967)

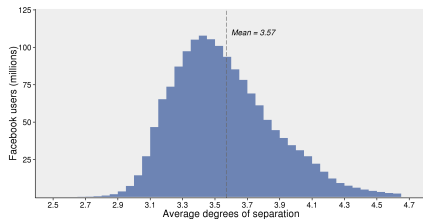
Per la prima volta nella storia, è relativamente facile (Per gli engineer di Facebook) calculate i gradi di separazione. . . e l'hanno fatto:

“Each person in the world (at least among the 1.59 billion people active on Facebook) is connected to every other person by an average of three and a half [3.57] other people.”

(Da <https://research.fb.com/three-and-a-half-degrees-of-separation>)

Diminuito dal 2011, quando era 3.74. All'interno degli USA: 3.46

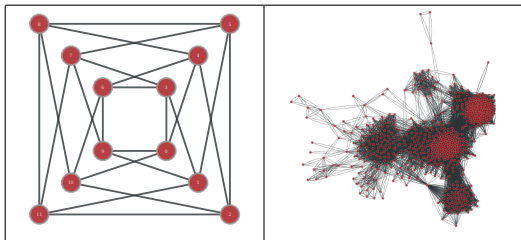
“La maggior parte della gente su Facebook ha una media tra 2.9 and 4.2 gradi di separazione.”



Comunità

1. *cosa* è una comunità
(cosa *condividono* i nodi in una comunità?)
2. *come* sono strutturate le comunità tra di loro
(formano comunità più grandi/contengono comunità più piccole?)
3. *quali* approcci sono computazionalmente fattibili
(non possiamo provare tutte le combinazioni!)

Cavie



Hierarchical Clustering

- ▶ *cosa* è una comunità: *nodi che sono strutturalmente simili*
- ▶ *come* sono strutturate le comunità tra di loro: *come una gerarchia di strutture sempre più piccole*
- ▶ *quale* approccio computazionale:
 - ▶ Calcoliamo una misura della somiglianza dei nodi
 - ▶ Fissiamo una soglia
 - ▶ Due nodi sono simili se la loro differenza è meno della soglia
 - ▶ ... o se sono connessi a nodi simili

CONCOR

(Breiger, Boorman, Arabie, 1975)

- ▶ *cosa* è una comunità: **nodi che sono strutturalmente simili**
- ▶ *come* sono strutturate le comunità tra di loro: **come partizione di comunità più grandi**
- ▶ *quale* approccio computazionale:
 - ▶ Oggi nodo ha un vettore di nodi
 - ▶ i nodi “simili” sono quelli i cui vettori hanno una alta correlazione: **calcolarla** in una nuova tabella C^0
 - ▶ i nodi “simili”... sono quelli i cui vettori hanno descrizioni simili in questa nuova tabella...
 - ▶ ... e così via.

CONCOR = “**CON**vergence of iterated **COR**relations”

Edge removal

(Girvan & Newman, 2002)

- ▶ *cosa* è una comunità: **nodi con tante connessioni tra di loro**
- ▶ *come* sono strutturate le comunità tra di loro: **come sottocomunità di comunità più grandi**
- ▶ *quale* approccio computazionale:
 - ▶ Calcolare quanto è *centrale* ogni spigolo (quanto spesso ci passo per andare da un nodo all'altro)
 - ▶ rimuovere lo spigolo più centrale
 - ▶ ... e così via

Stochastic Blockmodeling (SBM)

- ▶ *cosa* è una comunità: **nodi con tante connessioni tra di loro**
- ▶ *come* sono strutturate le comunità tra di loro: **più o meno vicine**
- ▶ *quale* approccio computazionale:
 - ▶ Approccio probabilistico
 - ▶ Provare tante combinazioni (Markov chain Monte Carlo)