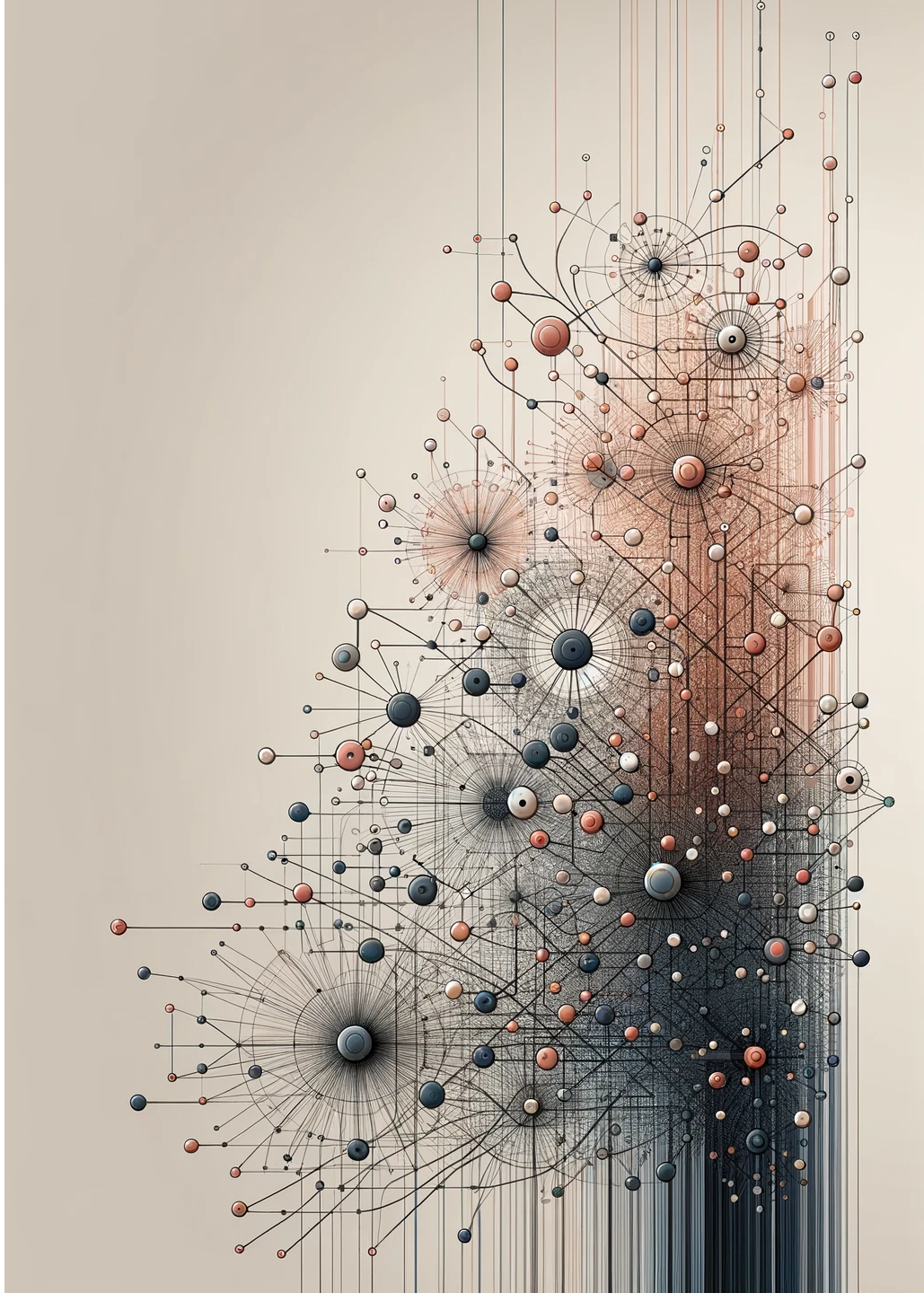
UNIVERSITÀ DI TORINO

Analisi e Visualizzazione delle Reti Complesse

NS13 - Communities - Benchmarks and Tutorial

Prof. Rossano Schifanella



UNIVERSITÀ DI TORINO

Outline

- Method evaluation
- Artificial benchmarks
- Real benchmarks
- Partition similarity
- Chapter 6 Tutorial (Python)

UNIVERSITÀ DI TORINO

Method evaluation

Method evaluation

- **Problem:** How can we tell how good a clustering algorithm is?
- **Solution:** The natural way to evaluate a method is to check whether it is able to find clusters in benchmark graphs, i.e., networks known to have a natural community structure
- **Two classes of benchmarks:**
  - **Artificial benchmarks:** computer-generated networks, built via some model
  - **Real benchmarks:** real networks in which the communities are suggested by the history of the system or by attributes of the nodes

Planted partition model

- Stochastic block models (SBMs) are often used to generate artificial benchmarks

```
# network with communities with sizes in the list S
# and stochastic block matrix P

G = nx.generators.stochastic_block_model(S,P)
```

- Special version of SBMs regularly used in method evaluations: **planted partition model**
- The planted partition model has only two link probabilities: the probability p_{int} that nodes in the same community are connected and the probability p_{ext} that nodes in different communities are connected
- If $p_{int} > p_{ext}$ the groups are communities, as two nodes are more likely to be connected if they are within the same group than if they are in different ones

Planted partition model

- q groups of identical size $\frac{N}{q}$
- Expected internal degree of the nodes:

$$\langle k^{int} \rangle = p_{int} \left(\frac{N}{q} - 1 \right)$$

- Expected external degree of the nodes:

$$\langle k^{ext} \rangle = p_{ext} \frac{N}{q} (q - 1)$$

- Expected total degree of the nodes:

$$\langle k \rangle = \langle k^{int} \rangle + \langle k^{ext} \rangle = p_{int} \left(\frac{N}{q} - 1 \right) + p_{ext} \frac{N}{q} (q - 1)$$

In NetworkX:

```
# network with q communities of nc nodes each
# and link probabilities p_int and p_ext

G = nx.generators.planted_partition_graph(q,nc,p_int,p_ext)
```

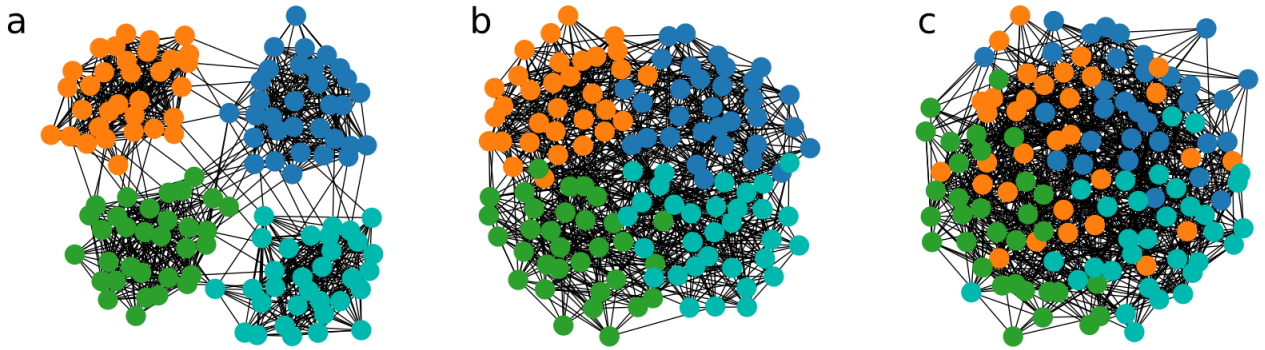
UNIVERSITÀ DI TORINO

GN benchmark

- Specific implementation of the planted partition model, in which the network size, nodes degree, and number and size of the communities are set to particular values:
  - $N = 128$, $q = 4$, $\langle k \rangle = 16$
- Since
  - $31 p_{int} + 96 p_{ext} = 19$
- Then
  - p_{int} and p_{ext} are not independent parameters
- GN benchmark networks are constructed with a procedure similar to the one adopted for Erdős–Rényi random graphs:
  - we go over all pairs of nodes and connect each with probability p_{int} or p_{ext}, depending on whether or not the nodes are in the same community

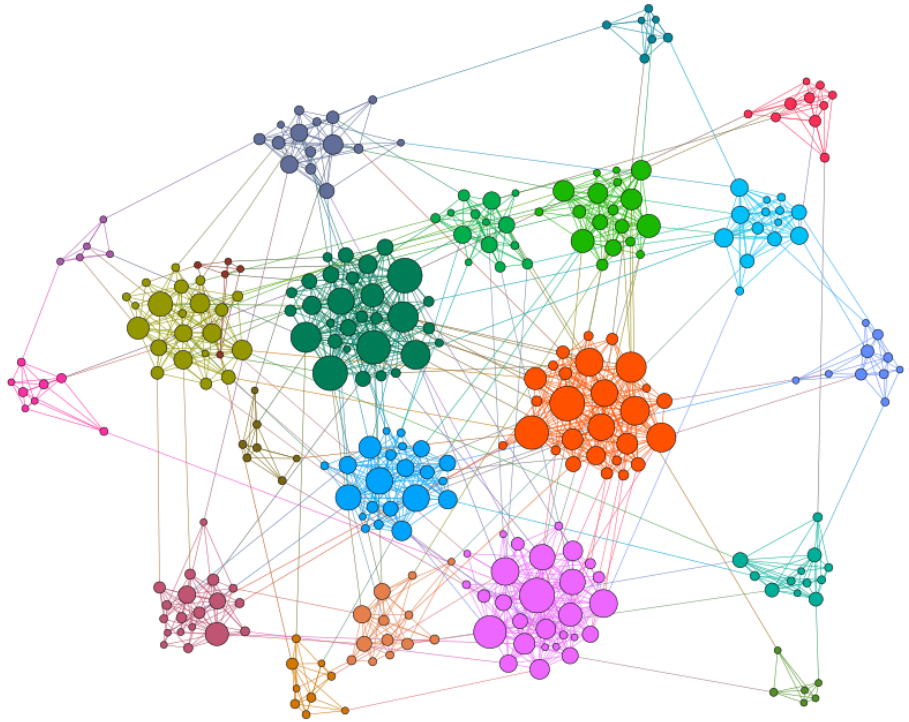
GN benchmark

- The higher the expected external degree and the lower the expected internal degree, the more difficult it is to detect the communities
- **Expectation:** communities should be detectable as long as $p_{int} > p_{ext}$, i.e., $\langle k \rangle$ smaller than (about) 12
- **Surprise:** the detectability threshold is quite a bit lower, around 9, due to random fluctuations in the placement of the links (detectability limit)



LFR benchmark

- The GN benchmark is not realistic!
- **Problem 1:** nodes have approximately the same degree
- **Problem 2:** communities have approximately the same size
- In real networks the distributions of node degree and community size are quite heterogeneous
- The **LFR benchmark** produces networks having heavy-tailed distributions of degree and community size



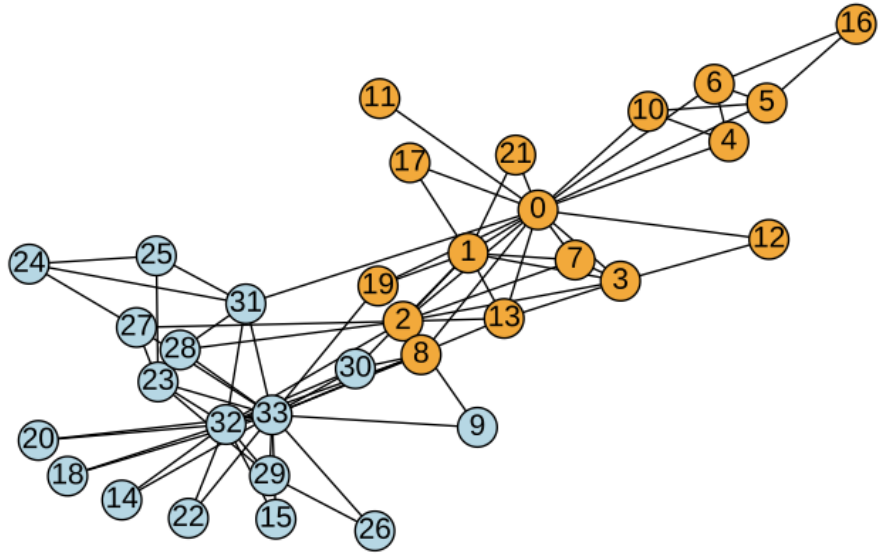
Negative tests

- **Negative test:** is the algorithm finding communities also in networks without communities?
- **Example:** if a method finds non-trivial partitions in random networks, it is unreliable
- **Trivial partitions:** one cluster including everything or N clusters of one node each
- Any non-trivial partition would signal the method's inability to distinguish actual communities from subnetworks with high concentrations of links generated by random fluctuations

Real benchmarks

- Classic example: Zachary's karate club network
- Network of social relationships between members of a karate club in the US
- Following a disagreement between the instructor (node 0) and the president (node 33), the club split in two different groups
- The task is to identify those groups (identified by the node colors) using a clustering algorithm

```
G = nx.karate_club_graph()
```



Real benchmarks

- Many other networks, whose nodes can be classified based on their attributes, are available for testing community detection algorithms
- **Examples:** in many social networks there are groups that users can decide to join, in citation networks papers can be grouped according to their publication venues, Internet routers can be categorized by country, etc.
- **Question:** do clusters of nodes with similar attributes match the communities found by clustering algorithms based only on the network structure?
- **Answer:** if nodes with similar attributes are strongly linked to each other, the attributes can be revealed through community detection. If instead the attributes do not play a role in the build-up of the network, they remain invisible to clustering methods

Partition similarity

- **Question:** how can we tell how close the partition found by the algorithm is to the planted partition of a benchmark network?
- **Answer:** partition similarity measures!
- Different classes of measures, all of them with pros and cons
- Here we discuss two of them:
  - **Fraction of correctly detected nodes**
  - **Normalized mutual information**

Fraction of correctly detected nodes

- A node is **correctly identified** if it and at least half of the other nodes in the same community in the detected partition are also in the same community in the benchmark partition
- **Caveat:** if the detected partition has communities obtained by merging two or more groups of the benchmark partition, all the nodes of those clusters are considered incorrectly classified by this measure
- The number of correctly detected nodes is then divided by the number of nodes N of the network, yielding a number between zero and one
- **Problem:** the recipe to label nodes as correctly or incorrectly classified is somewhat arbitrary

Normalized mutual information

- **Alternative criterion:** if we know one partition, how much information do we need to infer the other one?
- If the two partitions are very similar, little information is needed to transition from one to the other. The more extra information is needed, the less similar the partitions
- The **normalized mutual information (NMI)** is a measure that quantifies such additional information

Normalized mutual information

- Two partitions: X and Y
- Probability that a randomly chosen node belongs to cluster x (with size N_x) of partition X

$$P(x) = \frac{N_x}{N}$$

- Probability that a randomly chosen node belongs to cluster x of partition X and to cluster y of partition Y:

$$P(x, y) = \frac{N_{xy}}{N}$$

- where N_{xy} is the number of nodes shared by x and y
- Shannon entropy of X:

$$H(X) = -\sum_x P(x) log(P(x))$$

Normalized mutual information

- Conditional entropy of X given Y:

$$H(X|Y) = \sum_{x,y} P(x,y) log \left[\frac{P(y)}{P(x,y)} \right]$$

$$NMI(X,Y) = \frac{2H(X) - 2H(X|Y)}{H(X) + H(Y)}$$

- $NMI = 1$ if and only if the partitions are identical
- NMI has an expected value of zero if the partitions are independent, as for example when two random partitions are compared
- **Problem:** Detected partitions with more clusters may yield larger values of the NMI even though they are not necessarily closer to the benchmark partition

Your new iPython notebook

Chapter 6 Tutorial.ipynb

- Contents:
  - Partitions
  - Modularity
  - Zachary's Karate Club
  - Girvan-Newman clustering algorithm

UNIVERSITÀ DI TORINO

Reading material

References

[ns1] **Chapter 6 (Communities)**

UNIVERSITÀ DI TORINO

Q&A