P.h.D. course
Universitat Autonoma de Barcelona, 2022

(**S**ocial) **N**etwork **A**nalysis

Giulio Rossetti

Knowledge Discovery and Data Mining Laboratory (KDD) @ ISTI-CNR

giulio.rossetti@isti.cnr.it

@GiulioRossetti

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

Lecture 4

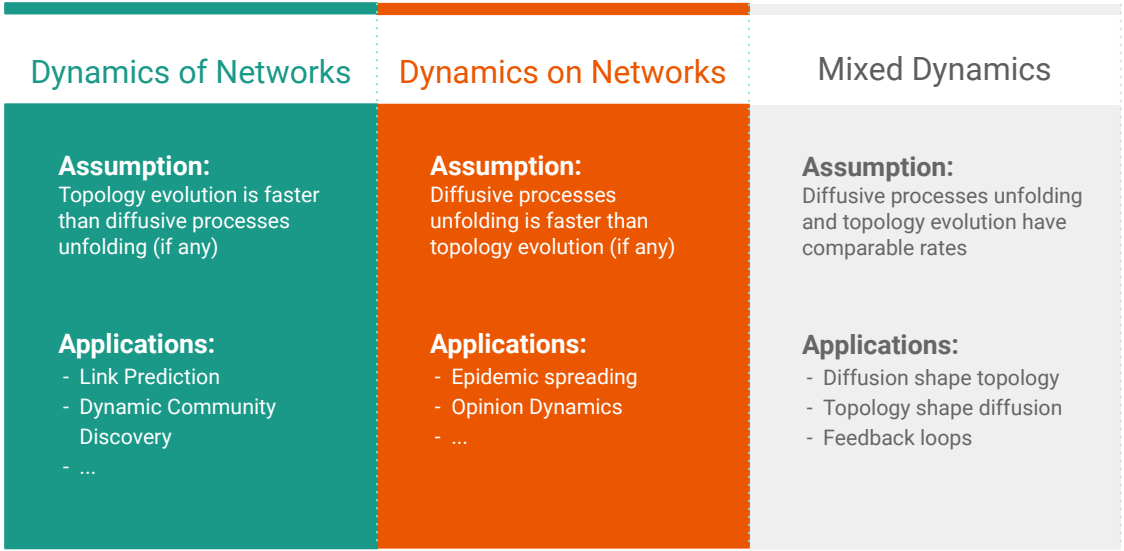
Dynamics of Networks: Topology perturbations

Where next?

Two kind of dynamics:

- Dynamics of Networks
  (topological perturbations)
- Dynamics on Networks
  (diffusive phenomena: epidemics, opinion dynamics…)

Of course they can happen at the same time...



Chapter 8

Representing Dynamic Topologies

Summary

- Representing Dynamic Topologies
- Analyzing Dynamic Networks

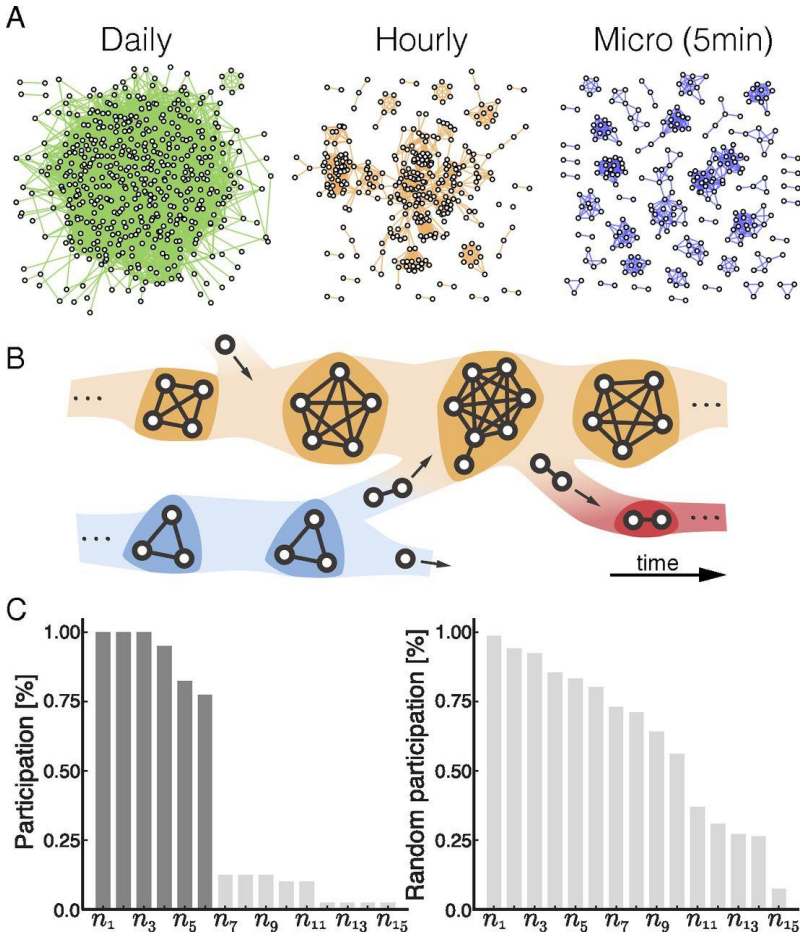
Reading

- “Temporal Networks” Holme et al.

Why bother of time?

Most real world networks are dynamic

- Facebook friendship
  - People joining/leaving
  - Friend/Unfriend
- Twitter mention network
  - Each mention has a timestamp
  - Aggregated every day/month/year => still dynamic
- World Wide Web
- Urban networks
- Protein-protein interactions
- Brain networks
- ...



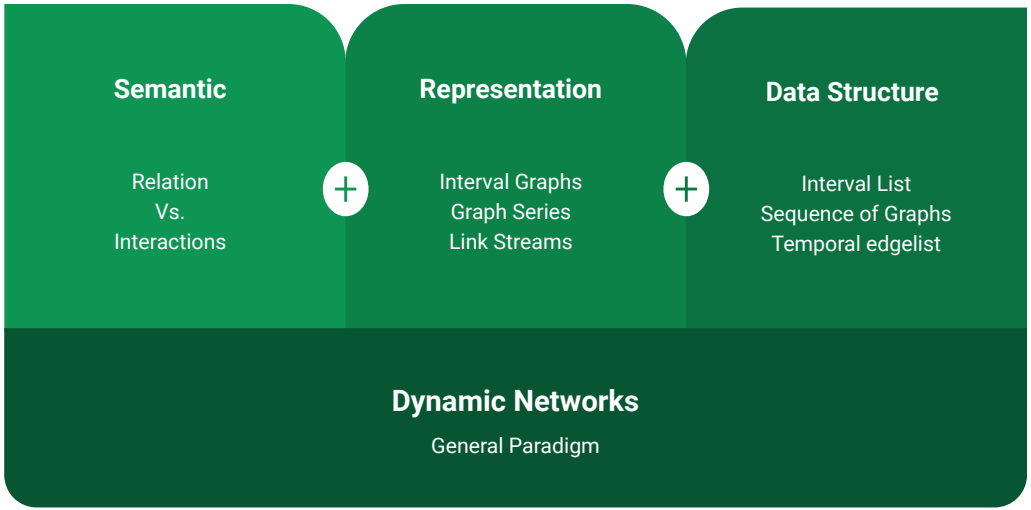
Evolving Topologies

- Nodes can appear/disappear
- Edges can appear/disappear
- Nature of relations can change

How to represent those changes?

How to manipulate dynamic networks?

Three different levels of abstraction



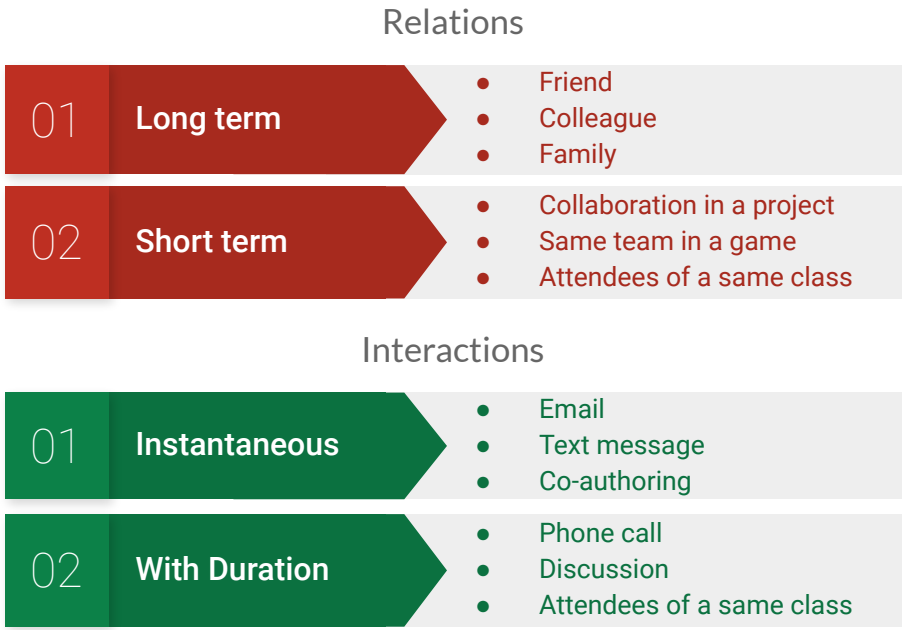
Semantic

Relations Vs. Interactions

Topological perturbations may have different temporal scales depending on their intrinsic semantic value.

Two families:

- Relations (stable ties)
- Interactions (unstable ties)



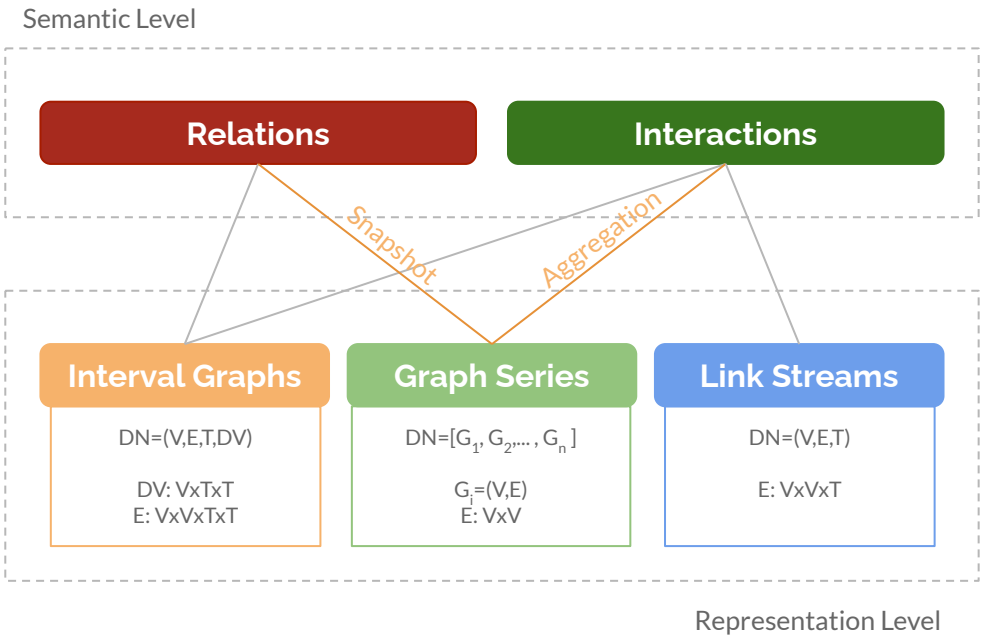
Semantics and how to represent them

Relations

The graph is more and more stable, until most observations are completely similar to previous/later ones (frequency faster than change rate)

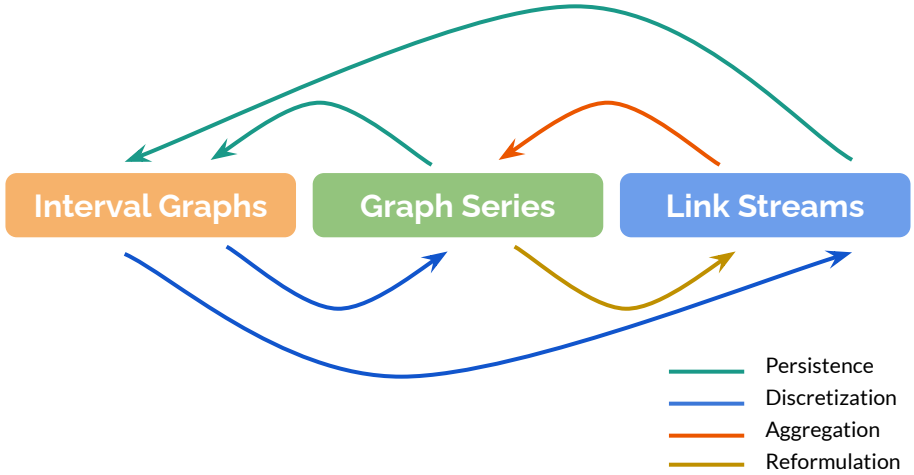
Interactions

The graph is less and less stable, until each observation is a graph in itself, thus completely different from previous/later ones (frequency faster than observed events rate)



Changing Representation

Alternative representations can be, to some extent, converted among them by applying appropriate data transformations



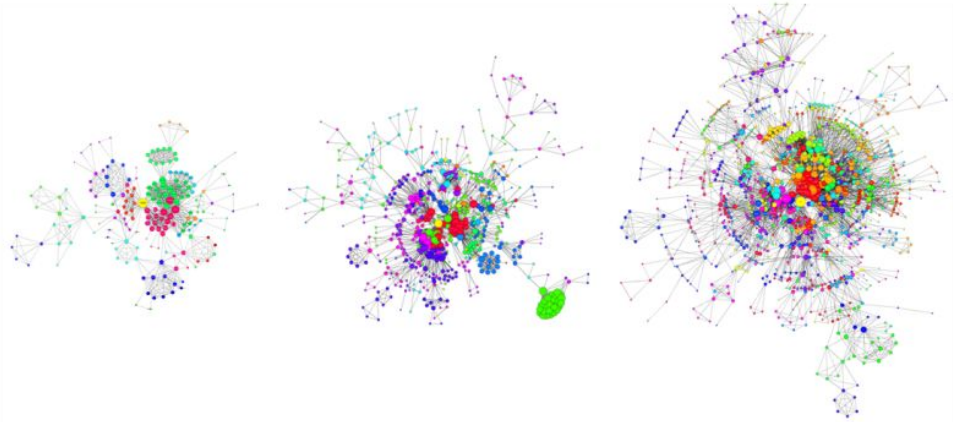
Analyzing Dynamic Networks

A brief Introduction

Unstable Snapshots

The evolution is represented as a series of a few snapshots

- Many changes between snapshots
  (Cannot be visualized as a “movie”)
- Each snapshot can be studied as a static graph
- Evolution of node properties can be studied “independently”
  (e.g., node i had low centrality in snapshot t and high centrality in snapshot t+n)



Stable Network

Edges change (relatively) slowly

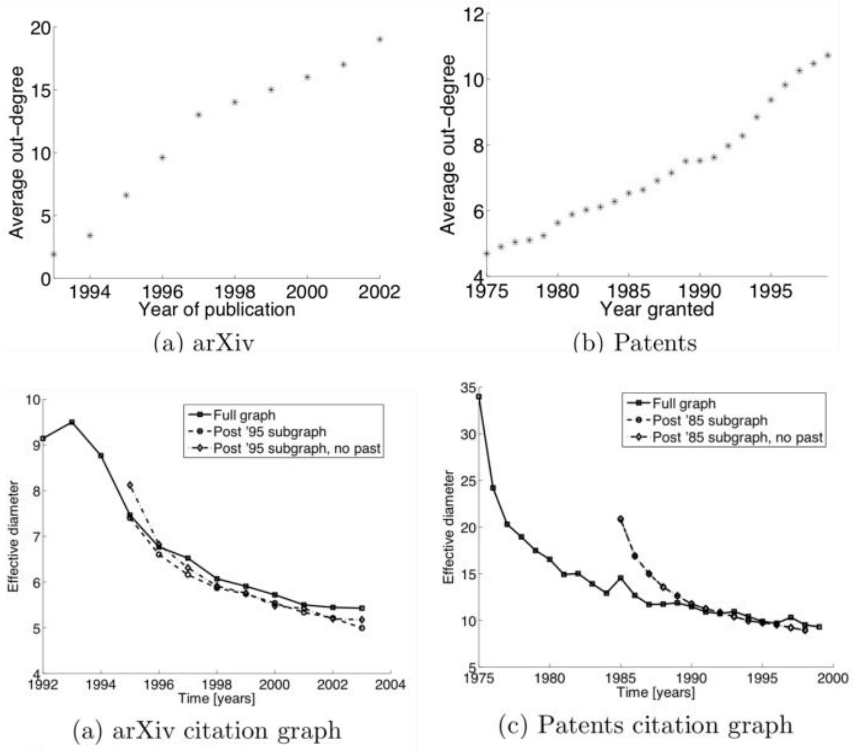
The network is well defined at any t

- Temporal network: nodes/edges described by (long lasting) intervals
- Enough snapshots to track nodes

A static analysis at every (relevant) t gives a dynamic vision

No formal distinction with previous case
(higher observation frequency)

Properties can be analyzed as time series



Leskovec, Jure, Jon Kleinberg, and Christos Faloutsos. "Graph evolution: Densification and shrinking diameters." ACM Transactions on Knowledge Discovery from Data. (2007)

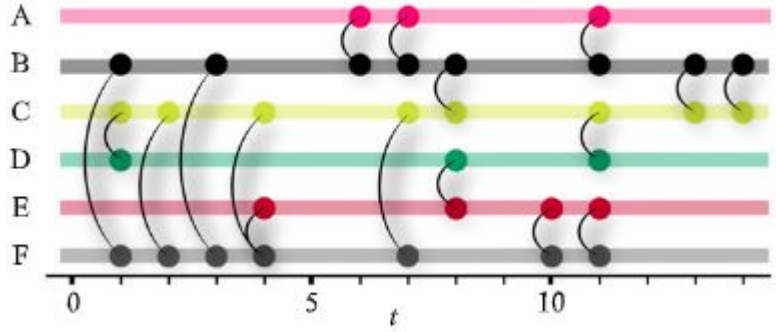
Unstable Temporal Network

The network at a given t is not meaningful

How to analyze such a network?

Until recently, network was transformed using aggregation/sliding windows

- Information loss
- How to choose a proper aggregation window size?



Holme, Petter, and Jari Saramäki. Temporal networks. Physics reports 519.3 (2012)

Chapter 8

Conclusion

Take Away Messages

1. Real phenomena evolve through time
2. Network modeling can fill such gap

Suggested Readings

- "Temporal Networks" Holme et al.

What's Next

Chapter 9:
Dynamic Community Discovery

Chapter 9

Dynamic Community Discovery

Summary

- Communities in dynamic networks
- Evaluation & Benchmarking
- Visualization

Reading

- *"Challenges in community discovery on temporal networks."* Cazabet & Rossetti

Communities
In Dynamic Networks

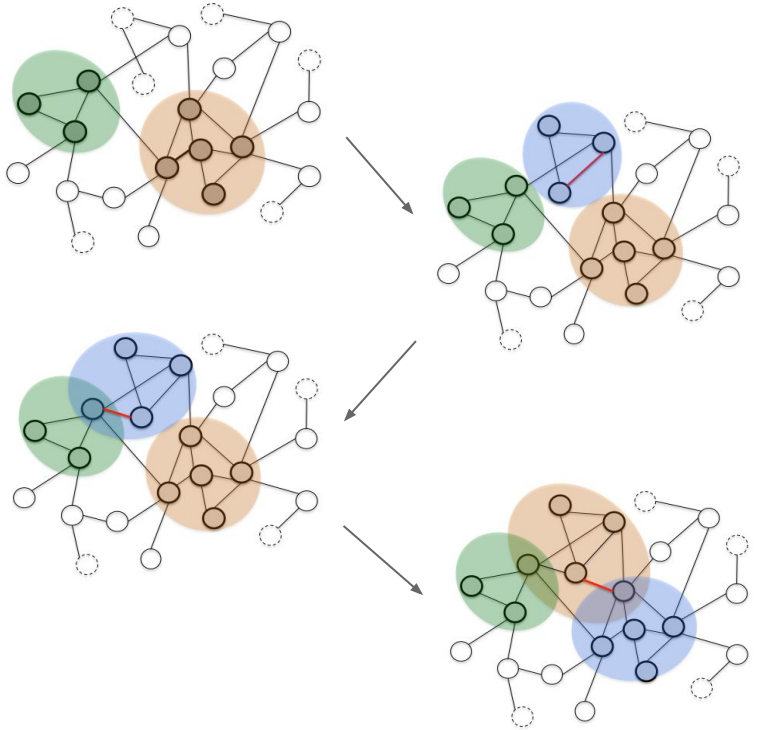
Networks change with time…

- Nodes appear and vanish
- Edges appear and vanish

…communities must change too!

DCD:

identify/track changes in community structure



Cazabet, Remy, and Giulio Rossetti. "Challenges in community discovery on temporal networks." Temporal Network Theory. Springer, Cham, 2019. 181-197.

A Novel Problem:

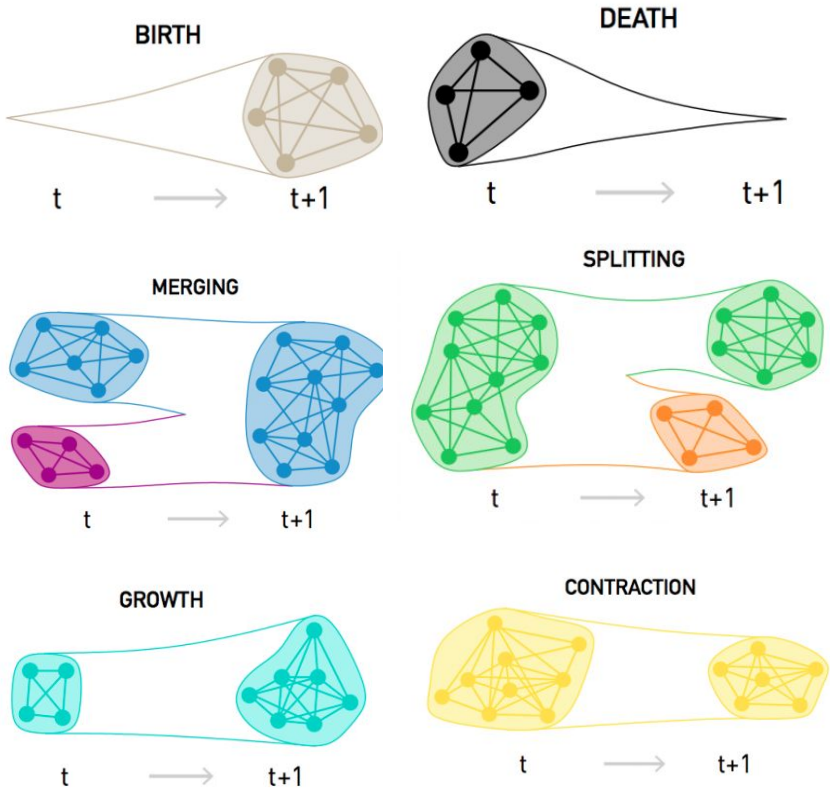
Community life-cycle tracking

As time goes by the rising of novel nodes and edges (as well as the vanishing of old ones) led to network perturbations

Communities can be deeply affected by such changes

Three main strategies:

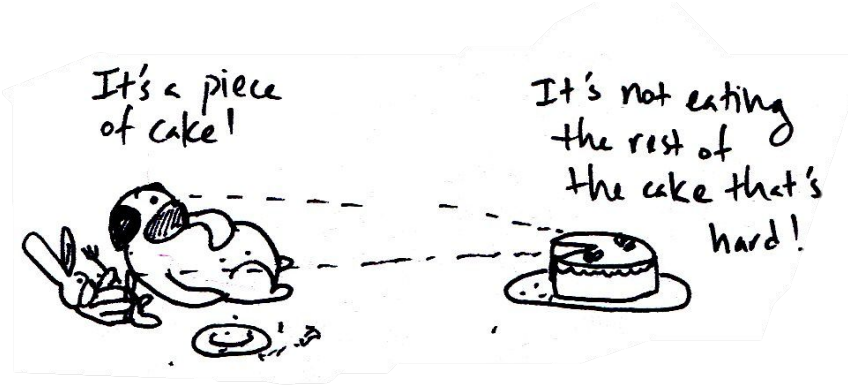
- Identify & Match
- Informed Iterative algorithms
- Stable Identification



The Optimist:

“Ok, It’s a piece of cake!”

1. Find communities at each network observation (using a static algorithm)
2. Match communities across consecutive network observations
3. Observe differences



Two major issues:

- Community Smoothing
- Theseus’ Ship Paradox

Community Smoothness

Communities are arbitrarily defined
(same issue of static CD)

Most “efficient” algorithms are stochastic

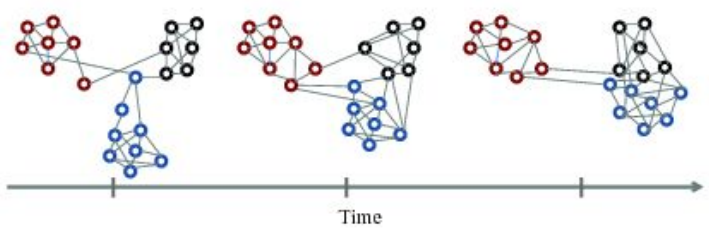
- Change in communities might be due to structural changes OR to arbitrary choices of the algorithm
- The same algorithm ran twice on the same graph *might yield* different results

Desiderata:

- a “simple” (parsimonious) model
- a trade-off between quality and simplicity (smoothness)

No Smoothness:
Partition at each t should be the same as found by a static algorithm

Smoothness:
Partition at t is a trade-off between “good” communities for the graph at t and similarity with partitions at different times



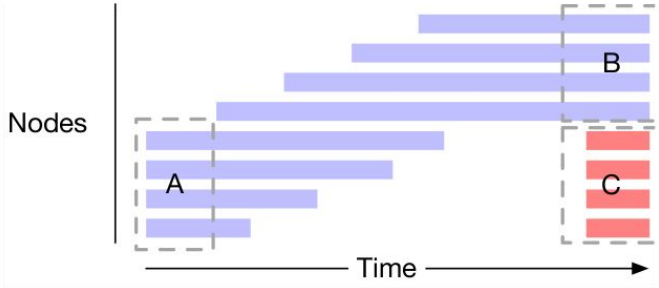
Theseus' Ship Paradox

I. Theseus killed the Minotaur in Crete and came back to Athens on his boat

II. His boat was conserved as memory during a very long time

III. The boat was deteriorating, so pieces of it were gradually replaced.

IV. Until one day, all original parts were replaced



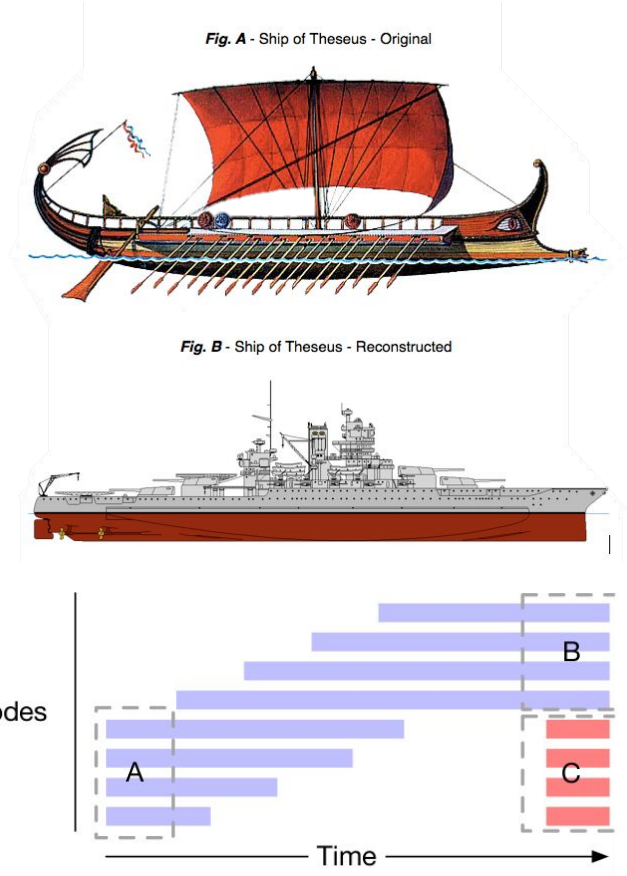
Plutarch 75

Theseus' Ship Paradox

A. Is this ship still the same as Theseus boat ?

B. If another boat was built using all pieces of the original boat, which one would be the "real" Theseus boat ?

Community evolution/identity is an arbitrary concept



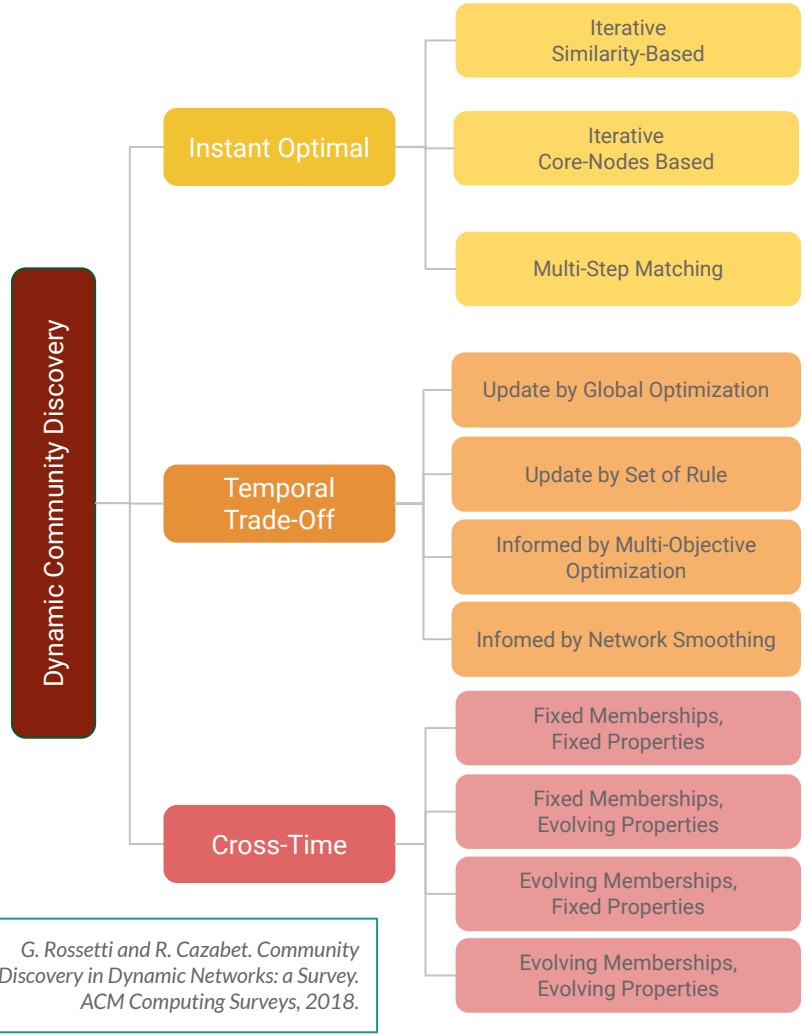
DCD Algorithms Taxonomy

Hierarchical categorization

First Level:
Increasing degree of smoothness *(none -> complete)*

Second Level:
Algorithmic Approach *(how to deal with Theseus)*

-



G. Rossetti and R. Cazabet. Community Discovery in Dynamic Networks: a Survey. ACM Computing Surveys, 2018.

Taxonomy

Instant Optimal

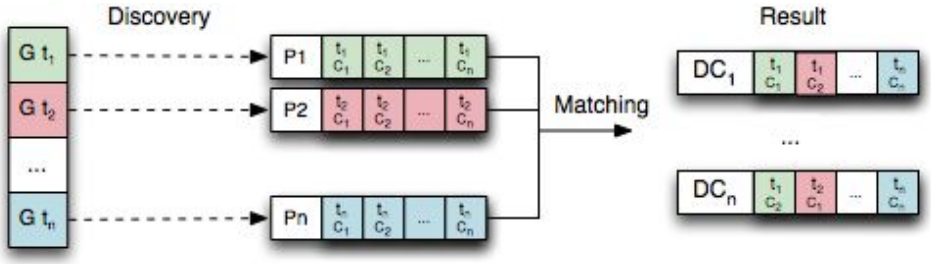
"Communities found at time t are optimal for the network at time t"

Strengths

Definition consistent with static CD, parallelisation

Drawbacks

Lack of smoothness, only Snapshot Network repr.



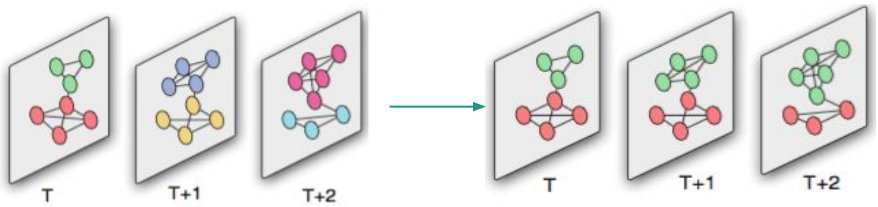
Taxonomy

Two-Step

1. Communities are detected at every step using a static algorithm (e.g. Louvain Algorithm)
2. Similarities are computed between communities in consecutive steps (at t and t+1 (e.g., Jaccard index))

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

3. Most similar communities are matched between t and t+1



Advantages:

- Easy to model, can extend smoothly static approaches

Drawbacks:

- The reduction to static scenarios trough temporal discretization is not always a good idea
  - How to choose the temporal threshold?
  - To what extent can we trust the obtained results?

Greene, et al. "Tracking the evolution of communities in dynamic social networks." 2010 international conference on advances in social networks analysis and mining. IEEE, 2010

Taxonomy

Temporal Trade-Off

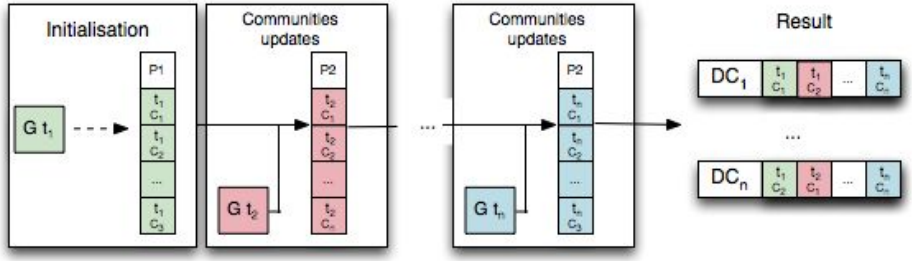
"Communities found at time t represent a trade-off between the graph at t and its previous states"

Strengths

Online, incremental, natural smoothness

Drawback

Iterative, risk of avalanche effect



Taxonomy

Tiles

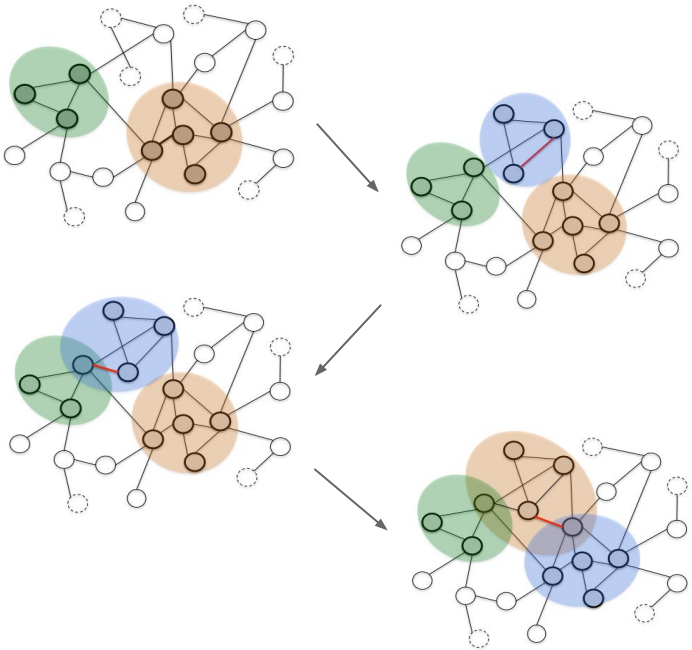
1. Social Interactions define the communities a user belongs to
2. Dynamic graphs as *edge streams*
3. Online updates of communities as nodes/edges appear/vanish

Advantages:

- Punctual updates of the community structure
- Low computational complexity

Drawbacks:

- Ad-Hoc model



Rossetti, et al. "Tiles: an online algorithm for community discovery in dynamic social networks." Machine Learning 106.8 (2017): 1213-1241.

Taxonomy

Cross-Time

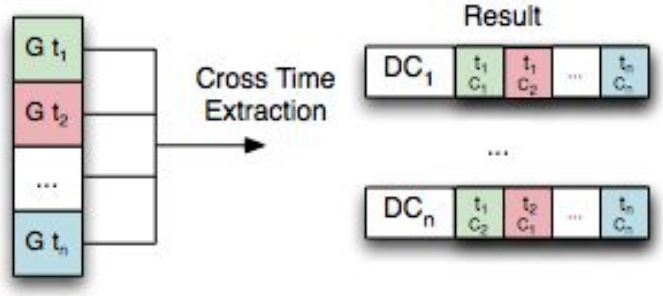
"Communities at t are defined relatively to all other steps"

Strengths

Perfectly smoothed, stable, solution

Drawback

Non online, batch computation, lacks incrementality

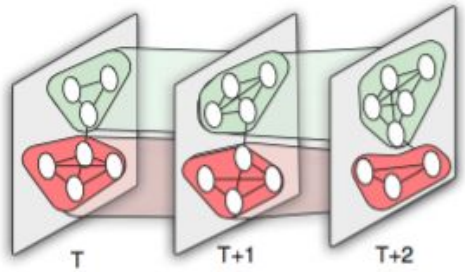


Taxonomy

Transversal Network

1. A transversal network is built: nodes are couples (nodes, time), edges link the same node in adjacent snapshots
2. A community detection algorithm is run on this transversal network

   (Note: modified Modularity to avoid overestimating expected edges between nodes in different time steps, i.e., custom random graph)



Advantages:

- Maximal smoothing and stability

Drawbacks:

- No Community Events are detected
- All the network history needs to be known in advance

Mucha, Peter J., et al. "Community structure in time-dependent, multiscale, and multiplex networks." science 328.5980 (2010): 876-878

Community Discovery in Dynamic Networks

Evaluation Strategies

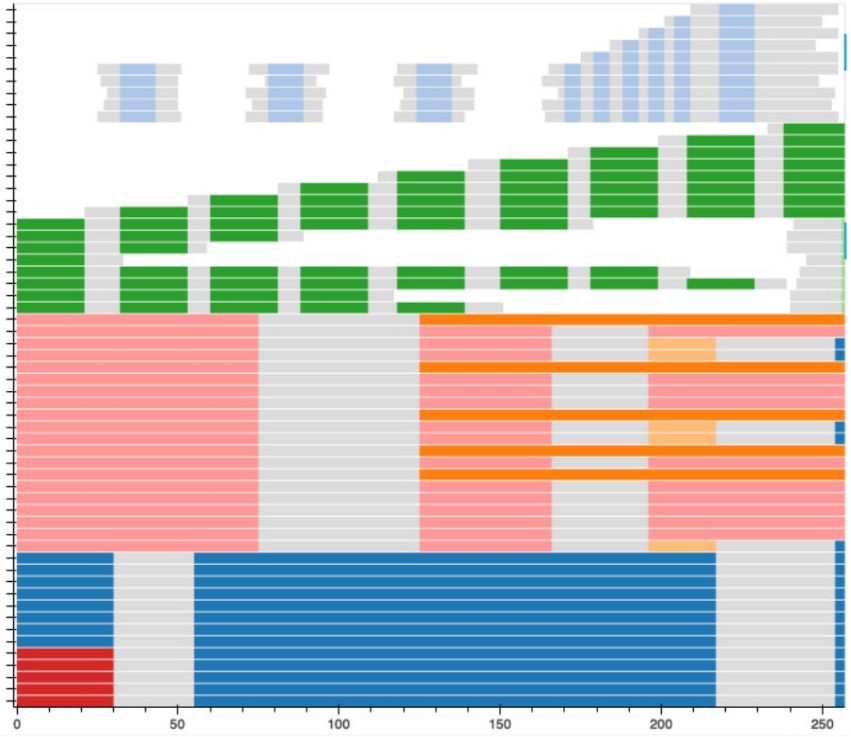
Strategies

Internal Evaluation

- Partition quality function
  (i.e., modularity, conductance, density…)
- Community characterization
  (i.e., size distribution, overlap distribution…)
- Execution time and Complexity

External Evaluation

- Ground truth testing
  (or partitions comparison)



Ground truth testing: Issues

- Few real world datasets with annotated ground truth partition are available
  (mostly static networks)
- Reliability of partition labelling
  (semantic partitions not always reflect topological ones)
- Scarcity of network generators handling community dynamics
  (i.e. birth, death, merge, split)

"I think we're past the point where rebooting will help."

Benchmark

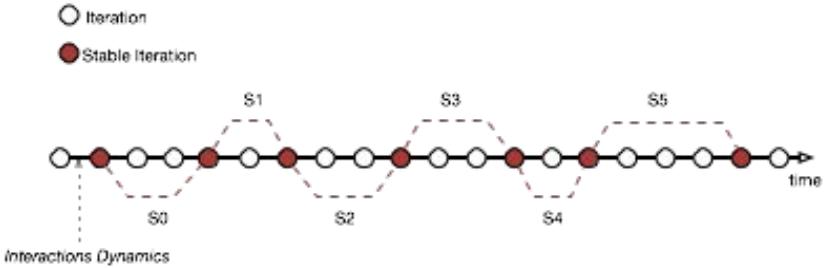
RDyn

Dynamic network generator that

- Guarantee power law degree distribution
- Guarantee small-world effect
- Exploit planted communities
  (having power law size distribution)
- Handle node/edge rise/fall
- Handle Community Dynamics (merge/split)
- Generate tunable-quality time-aware network partitions
  (i.e. conductance/modularity/density)

Expected outputs:

- Synthetic graph (TN/SN)
- Temporal communities with planted events



Events are handled by:

1. "Semantically" planting updated communities;
2. Converging to the final stable state by leveraging intra-community edge probability side effects

A state (iteration) is called stable when a minimum partition quality is reached (i.e. modularity/conductance/density)

G. Rossetti, RDyn: graph benchmark handling community dynamics. Journal of Complex Networks (2017)

Summarizing

Mesoscale Evolutions

Node/edge local dynamics affect community structures

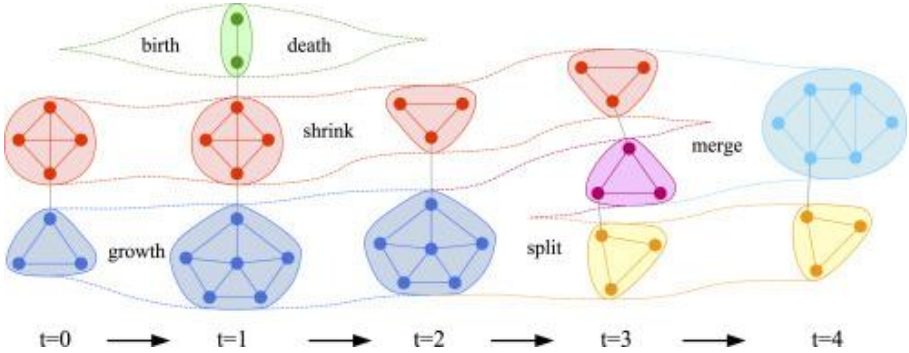
- Communities are subject to events/operations
- Life-cycles can be identified and studied

The complexity behind such ill posed problem grows

- Stability/Persistence
- Smoothness

Every family of approaches depend on

- Specific analytical needs
- Dynamic Network Representation adopted



Chapter 9

Conclusion

Take Away Messages

1. As topology evolve, community do too
2. Smoothness and stability are key issues
    a. Theseus ship paradox
3. Communities are subject to events
    a. Life-cycle tracking

Suggested Readings

- *"Challenges in community discovery on temporal networks."* Cazabet & Rossetti
- *"Community Discovery in Dynamic Networks: a Survey."* Rossetti & Cazabet

What's Next

Chapter 10:
Link Prediction

Chapter 10

Link Prediction

Summary

- Predicting Network Evolution
- Unsupervised approaches
- Supervised approaches
- Evaluation

Reading

- Liben-Nowel & Kleinberg paper

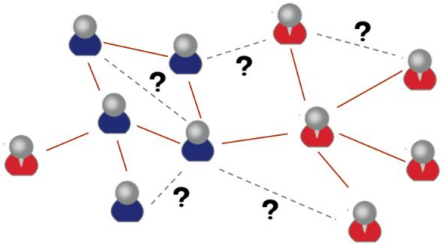
Link Prediction

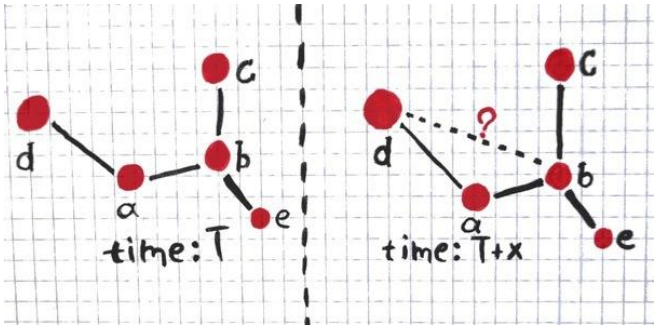
Goal

Understanding how networks evolve

Problem definition

Given a snapshot of a network at time **t**, (accurately) predict the edges that will appear in the network during the interval **(t, t+1)**





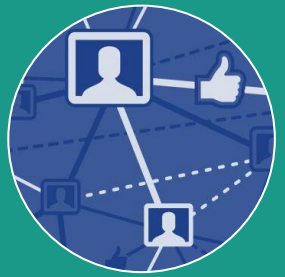
Liben-Nowell, David, and Jon Kleinberg. "The link-prediction problem for social networks." Journal of the American society for information science and technology 58.7 (2007): 1019-1031.

Examples of uses of

Link Prediction

Monitor terrorist networks – deducing possible interactions/missing links between terrorists (without direct evidence)

Suggest interactions or collaborations that haven't yet been exploited within an organization

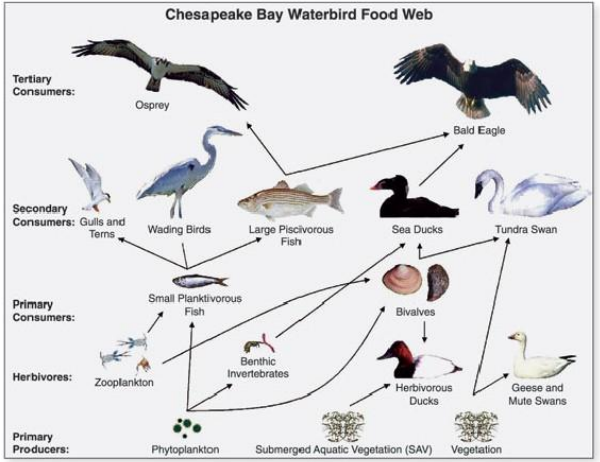


Friendship prediction (i.e., Facebook)

Link Prediction

Link prediction is used to predict future possible links in the network (e.g., Facebook).

Or, it can be used to predict missing links due to incomplete data (e.g., Food-webs)



facebook

tinder

amazon

PLOS ONE

RESEARCH ARTICLE

Link Prediction in Criminal Networks: A Tool for Criminal Intelligence Analysis

Giulia Berlusconi[1], Francesco Calderoni[1]*, Nicola Parolini[2], Marco Verani[2], Carlo Piccardi[3]*

1 Università Cattolica del Sacro Cuore and Transcrime, Milano, Italy, 2 MOX, Department of Mathematics, Politecnico di Milano, Milano, Italy, 3 Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milano, Italy

* francesco.calderoni@unicatt.it (FC); carlo.piccardi@polimi.it (CP)

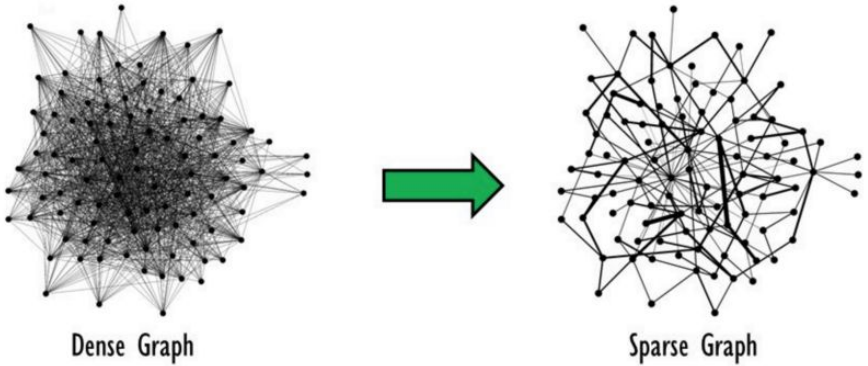
Link Prediction

Task Complexity

1. Given a graph G = (V,E) the set of possible edges to be predicted is O(|V|^2);

2. Real networks tend to be sparse

False Positive prediction issue
(i.e., we are likely to predict edges that will never appear)



Concretizing an Intuition...

Scientists who are close in the network
(i.e., have common colleagues)

→ will likely collaborate in the future

Goal:

- make this intuitive notion precise and understand which measures of "proximity" leads to accurate predictions

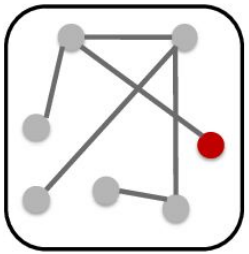
Link Prediction

Workflow

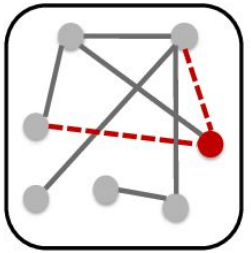
1. Consider as input a graph G at time t
2. Consider all the possible pairs of nodes (u,v)
3. Compute a link formation scores:

   score(u,v)
4. Build a list of all possible edges ordered by scores (from highest to lowest)
5. Verify, following that ordering, the predictions on the graph at time t+1

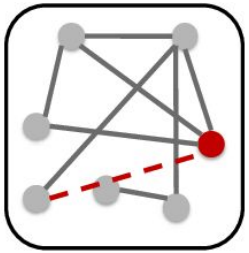
score is a measure of *proximity*



Time t



Time t+a



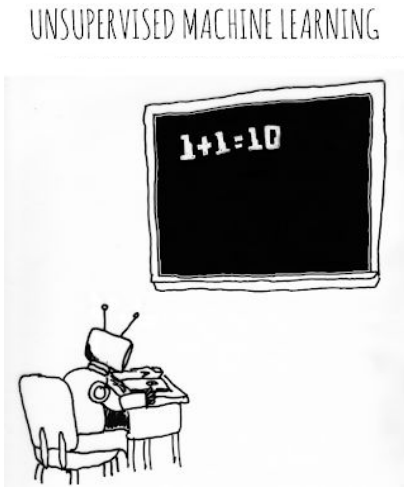
Time t+2a

Link Prediction

Approaches

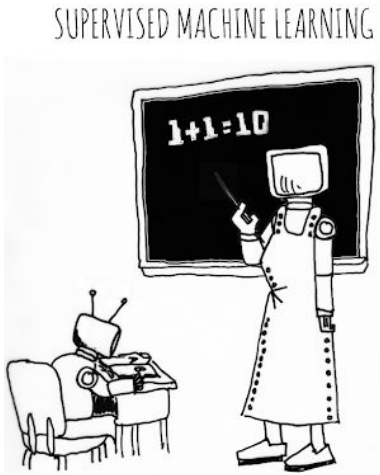
Unsupervised

Define a set of proximity measures unrelated to the particular network



Supervised

Extract knowledge from the network in order to improve prediction accuracy



Unsupervised Link Prediction

Unsupervised Link Prediction

Unsupervised measurements rely on different structural properties of networks

Neighborhood measures

- Common Neighbors, Adamic Adar, Jaccard, Preferential Attachment

Path-based measures

- Graph distance, Katz

Ranking

- Sim Rank, Hitting time, Page Rank

Liben-Nowell, David, and Jon Kleinberg. "The link-prediction problem for social networks." Journal of the American society for information science and technology 58.7 (2007): 1019-1031.

Unsupervised Link Prediction

Neighborhood measures

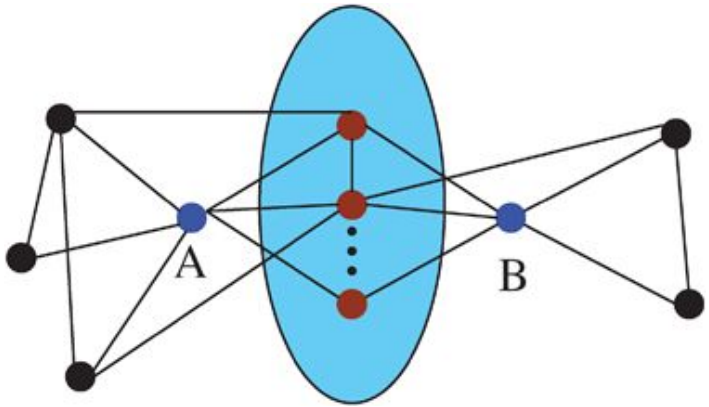
How many friends we have to share in order to become friends?

Common Neighbors:
the more friends we share,
the more likely we will become friends

$$\text{score}\,(u, v) = |\Gamma(u) \cap \Gamma(v)|$$

Jaccard:
the more similar our friends circles are,
the more likely we will become friends

$$\text{score}\,(u, v) = \frac{|\Gamma(u) n \Gamma(v)|}{\Gamma(u) \cup r(v)|}$$



Unsupervised Link Prediction

Neighborhood measures

How many friends we have to share in order to become friends?

Adamic Adar:

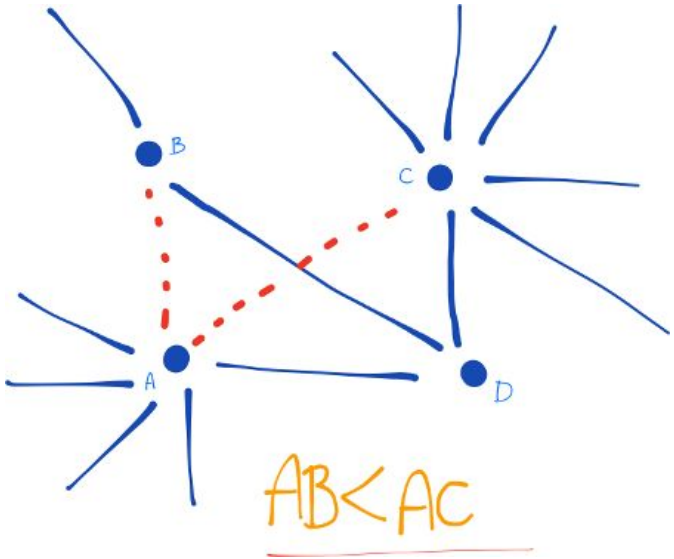
the more selective our mutual friends are,
the more likely we will become friends

$$\text{score}\ (u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log(|\Gamma(z)|)}$$

Preferential Attachment:

the more friends we have,
the more likely we will become friends

$$\text{score}\ (u, v) = |\Gamma(u)| * |\Gamma(v)|$$



Unsupervised Link Prediction

Path-based measures

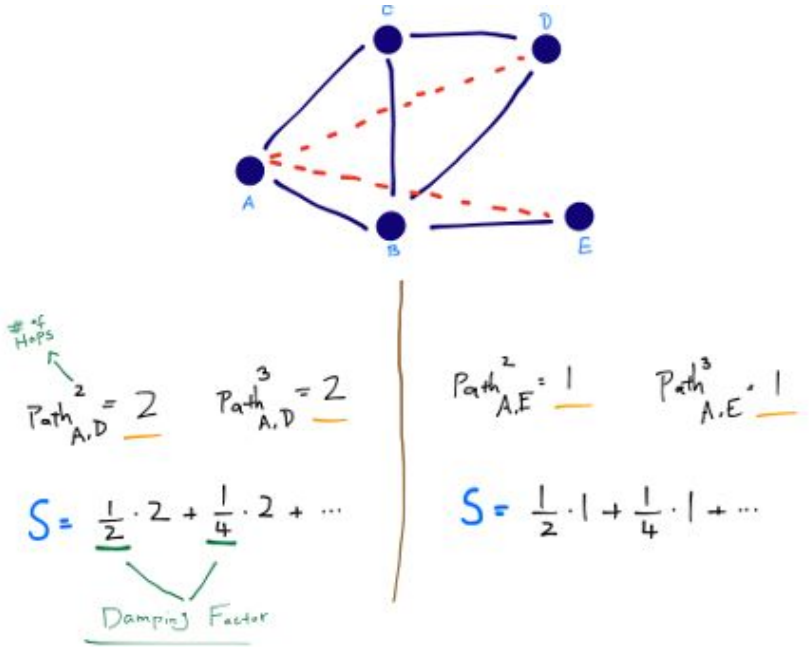
How distant are we?

Graph Distance:
(negated) length of the shortest path between two nodes

Katz:
weighted sum over all the paths between two nodes

$$\text{score}\,(u,v) = \sum_{l=1}^{\infty} \beta^l |\,\text{paths}_{u,v}^{\langle l \rangle}|$$

where: $paths_{u,v}^{\langle l \rangle}$ ={paths of length exactly *l* from *u* to *v*}



Unsupervised Link Prediction

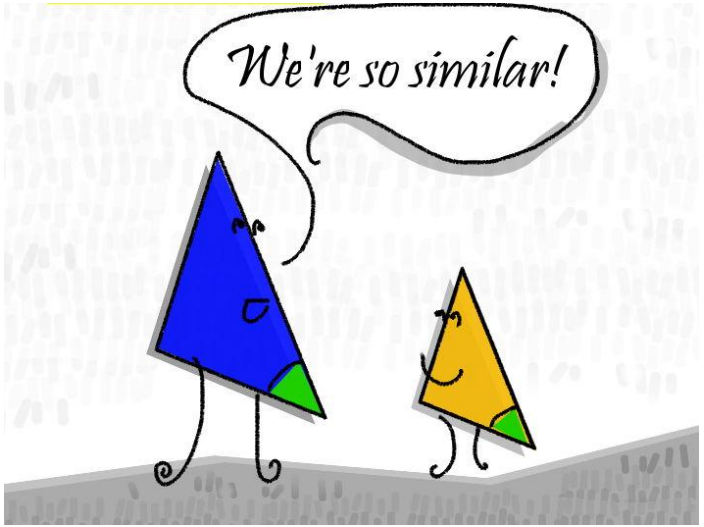
Ranking

How similar are we?

SimRank:
two nodes are *similar* to the extent that their neighborhoods are *similar*

$$\text{similarity}\ (u,v) = \gamma * \frac{\sum_{s \in r(u)} \sum_{n \in \Gamma(v)} \text{similarity}\ (a,b)}{|\Gamma(u)| * \Gamma(v)|}$$

$$\text{score}\ (u,v) = \text{similarity}\ (u,v)$$



Unsupervised Link Prediction

Limits

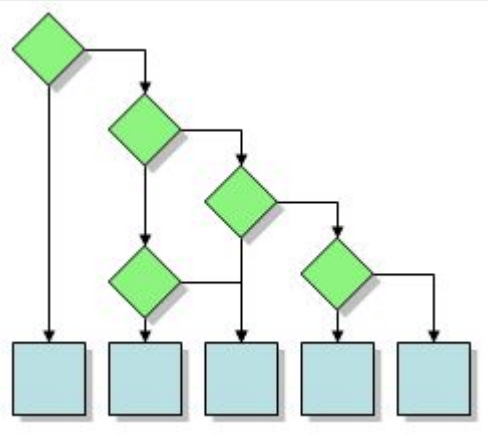
- Different kinds of networks are described by the same general closed formula
- An average overall performance between 6% and 12%

Measure comparison

- No single winner
- Almost all predictors outperform the random predictor
  ⇒ there is useful information in network topology

Supervised Link Prediction

Supervised Link Prediction



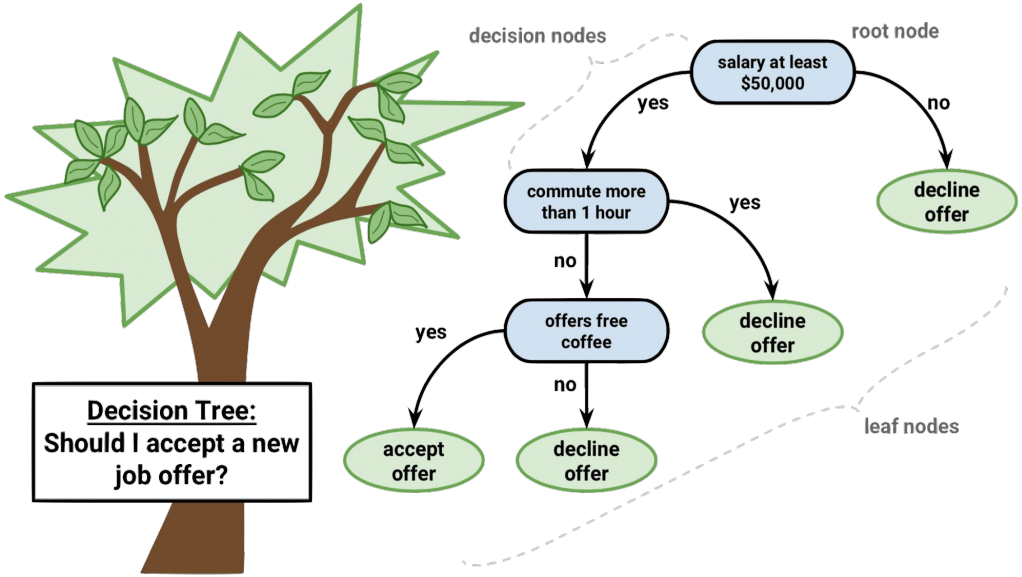
The process is now organized in 4 steps:

1. Split the data in train/test
2. **Learning a model on the train**
3. **Use the model for prediction**
4. Compare the prediction with the test

A natural way to do it:
build a "*classifier*" over a set of *network features*.

Staking Unsupervised Scores

Learn a Classifier (i.e., a Decision Tree) over unsupervised LP scores to generalize the assumption they made on the network growth model



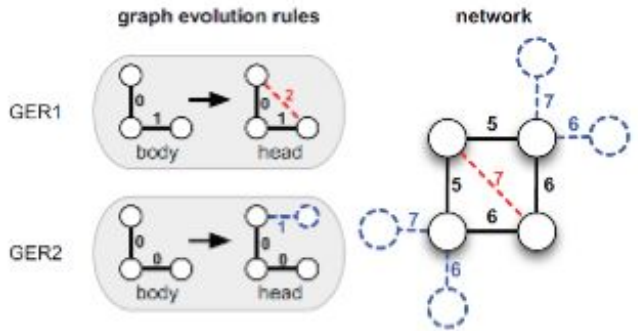
Supervised Link Prediction

Frequent Pattern Mining

GERM

Evolution rules can be extracted from the network history and used to identify/predict recurrent patterns

- e.g., generalization of triadic closure



Berlingerio, Michele, et al. "Mining graph evolution rules." joint European conference on machine learning and knowledge discovery in databases (2009).

Supervised Link Prediction

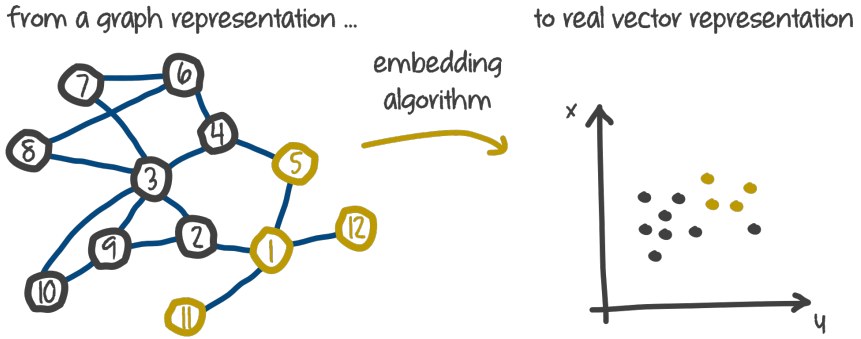
Network Embedding

Idea

Graphs can be *mapped* into vector spaces

- Node/edge similarity scores can be used to define metric spaces
- Metric spaces enable a more natural application of approaches from DM/ML

NB: Different “mappings” facilitate the solution of different classes of problems



Supervised Link Prediction

Limits

- No Free Lunch
- Model construction is often complex and, usually, more time/resource *demanding* than directly applying unsupervised scores.

Results: Higher performances w.r.t. unsupervised approaches

Embedding is not The Answer,
only a different way to reason on graphs...

Evaluation

Given a predictor p is there a way to decide if it is a "good" one?

First Step:

verify that p outperforms the *random predictor*.

Random Predictor

each edge has the same probability to appear in the network

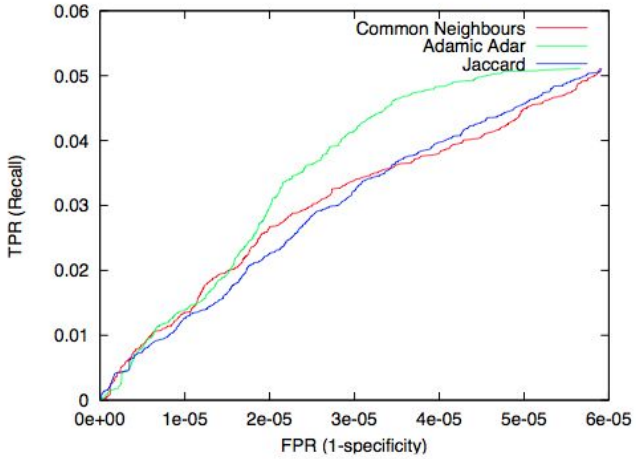
$$\text{performance}\ (p) = \frac{TP}{TP + FP}$$

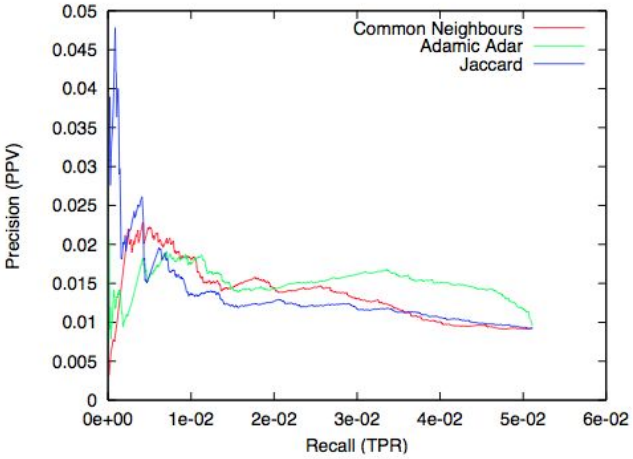
$$\text{ratio} = \frac{\text{performance}\ (p)}{\text{performance}\ (\text{prandom})} = \frac{\text{performance}\ (p)}{\frac{\mathbb{V}(-(\sqrt{V}-1)|}{2} - |E_{\text{old}}|}$$

if ratio > 1 then p is meaningful

Evaluation: Comparing Predictors

We need to analyze either the performances ratio, ROC and/or Precision Recall curve.





Evaluation: ROC and PR curve

Precision Vs. Recall

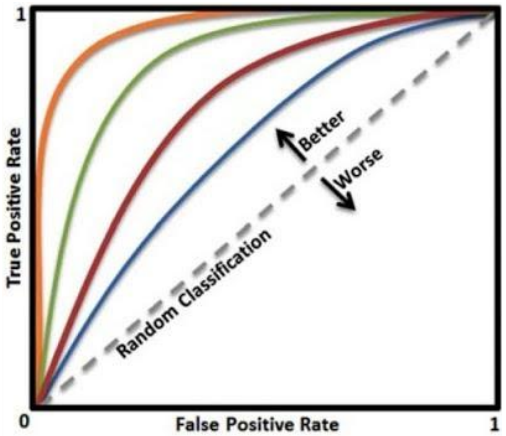
- Precision: PPV = TP/(TP+FP)
- Recall: TPR = TP/(TP+FN)

ROC (Receiver operating characteristic)

- 1-Specificity: FPR = FP/(FP+TN)
- Recall: TPR = TP/(TP+FN)

Note:

- ROC and PR spaces are isomorphic (the use of ROC is more widespread)
- Numerical comparison can be done using the AUROC (area under the ROC curve)



	p'	n'
p	TP	FN
n	FP	TN

Confusion Matrix

Link Prediction

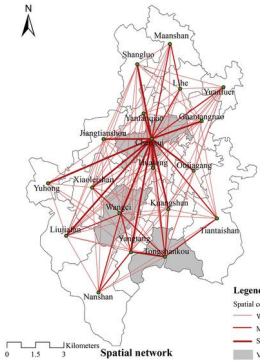
Something more...

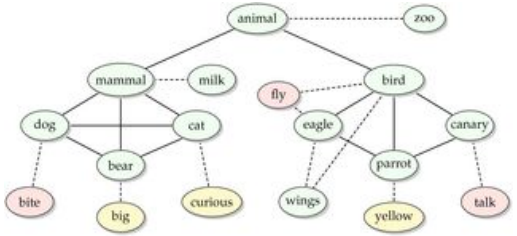
Accuracy could be improved extending simple models with more complex (even semantic) informations:

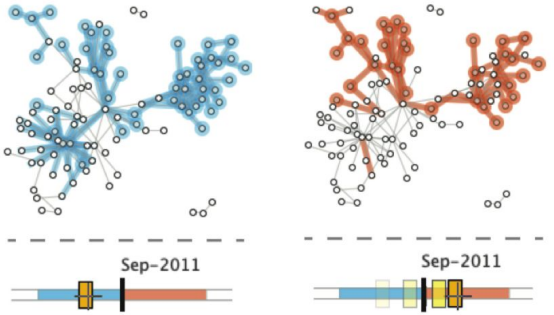
- Link strength
- Geographical information
- ...

Link Prediction needs to be revised while in some scenarios:

- **Dynamic Networks**
- Multiplex networks
- ...







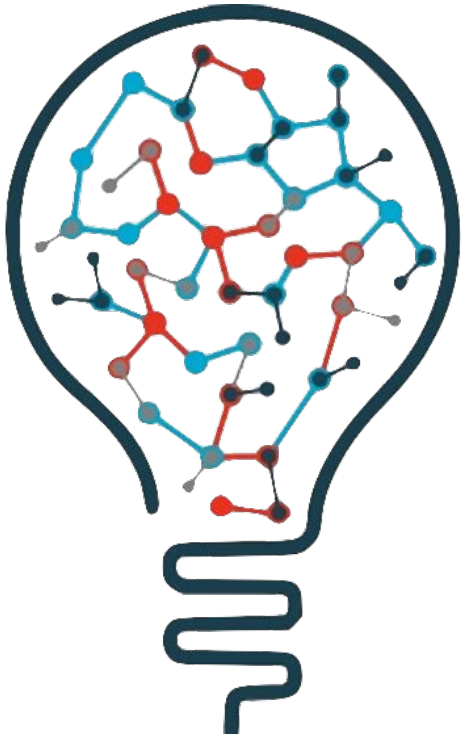
Key Messages

Predict new links that will arise in a network is **not easy**:

1. Networks are, usually, **sparse**
2. **Cold Start** Problem
    - What if I don't have enough information?
        - *Can I predict bridges?*

3. Huge **False Positive** prediction
    - *Bridges* !?!

4. Simple approaches are **"too simple"**
5. Complex approaches are **costly**



Chapter 10

Conclusion

Take Away Messages

1. Link wiring patterns define how network topology evolves
2. Predicting new links is a complex task due to network sparse topologies
3. Static and dynamic formulation of the problem account for different peculiarities

Suggested Readings

- Liben-Nowell & Kleinberg paper

What's Next

Lecture 5:
Dynamics on Networks