Analiza omrežij

1.predavanje (21.2.2024)

Logistics of the course: 2Hws, project, openbook exam

Newman – Networks: An Introduction (2018)

Netwoek science

Graph – points(nodes) connected with edges

Web is the largest graph, second is the human brain

Internet: nodes are class C subnets and autonomus entities (like the faculty) that route by themselves, links are packet routes.

Facebook: nodes – users, links – social connections (online)

Society – firnedships – division based on race and age (offline)

Assortatitve mixing nodes connect to similar nodes (same authors collaborate with similar authors, homosexual relationships in a sex intercourse graph)

Symetric (social friendship) vs. Asymetric connections (citing another paper)

Example from ecology: food web, in topological (?) layers, cannibalism (loop connection from the node to the node itself), we will usually remove those loops (they usually have no meaning)

Hub-spokes arrangement : hub in the middle, links (spokes ? ) out of it

Networks are studied in order to understand the real systems. If they are to complex, we study them trough their structure(what is expected and what is not), evolution (how it came to such structure) and dynamics(dynamic changes traveling troutgh networks).

Small worlds networks

Scale-free networks

Connected – six degrees of separation (documentary)

Historical development of network science :

* it started with graph theory (Eulier - 7 bridges problem – first non trivial proof of graoh theory)
* Hamilton – travelling salesman problme
* Kirchhoff – laws of electrical circuits
* Kekule – Chemical structure of molecules

Operation reserach :

* Dijkstra – shortest path
* Kruskal – spanning tree
* Ford & Fulkerson – maximum flow/minimum cut
* Cartwright – signed graphs (weighted links with positive and negative weights – usually we need a nonnegative weigth) 🡪 we have stable and unstable structures based on the signs of the weights on the links
* Random graph theory –

Sociometry:

* Children sociograms (Moreno)
* Southern Women – Davis: women attending social events (they divide into two groups based on the events)
* University karate club – Zachary (most known graph) :
* Small world experiment – entire world is connected (confirmation of six degrees of separation)
* Strength of weak ties – Granovetter : strong and weak ties, weak ties are more useful (when looking for a new job social circle of me and my best friend are quite similar, while someone you only know as a collegue might know of some more job opportunities)
* Measures of centrality – Freeman : nodes importance

Bibliometrics:

* Scientific paper citations – SCI – Science Citation Index (Derek de Solla Price)
* Political scandals – Mark Lombardi
* Neural wirings – White et. Al – brain of a worm is completely mapped
* Transportation – Pelletier

Around the year 2000 3 things happened that made it possible for this field to evolve (way more citations of old papers from the field, mentioned above):

* Data became avaliable – seven orders larger than whatever exsisted before (before no one collected that many data)
* People realised thar mathematical models of graphs turn out to be useful to understand the structure of real networks (is a certain thing expected in a graph or is it suprising)
* The networks have the same structural properties – there has to be some reason for seeing these properties in science, society etc. It has to be a simple reason 🡪 modern networks from different fields are similar to one another – we can use the same tools and methods on all of them

Impact of network science: management(predicition of conflicts in a company), economic, epidemics,(predicting the spread of diseases), health(drug design), security (detecting insurance fraud), neuroscience (mapping of the human brain) and many others.

Problem – understanding the real networks

Means : study of netowrk properites, design of mathematical models, implementation of efficient algorithms

Goals: knowing the netwirk structure and evolution, knowing the network dynamic processes

**4 Layers** that we will consider when studying networks:

* Layers – whole network
* Clusters (their characteristics) – group of nodes
* Fragments – triangles and hub-spokes
* Individual nodes/links

1. GRAPHOLOGY AND NETWORKOLOGY

Terminology:

* Graph vs network: graph is a representation of a network, while network is some real-world problem.
* Graph is formal mathematical object, while network is a graph with real data
* Different terminology based on field:
  + Network scoence : edges/links
  + Graph theory: edges/relations
  + Social science: agents/brokers/units and ties

Terminology classes:

* Social networks: for example **affiliation networks** (some students are enrolled or affiliated at the university – two types of nodes, edges only appear betwen the nodes of different types), collaboration network is a projection of a bypartite graph, Facebook, offline, online, affiliation, author/actor collaboration
* Infromation networks: nodes are source of information, links show how information flows (they show information flow – usually in the other way than the links are pointed) : web, twitter, citation, communication, peer-to-peer
* Technological networks: human-made infrastructure with technological constraints(Internet, telephone, transportation, power grid, software)
* Biological networks: interaction between genes, cells, neurons in living beings.
* Ecological/lexical/financial/sports

Definition of a graph : n = number of nodes, N = set of nodes, L = set of links, m = number of links

If graph is **undirected**: links are unordered pair of nodes {i,j}

If graph is **directed**: links are ordered pairs of nodes (i,j)

**Adjecency matrix** is a nxm matrix, elements od the matrix tell you wheteher i and j nodes are connected. The edge in row i and column j tells you there is a connection from j to i (not the other way around which is more natural). We do it that way to simplify some formulas with eigenvectors later on. If there are no loops, the diagonal elements are always zero. Adjeceny matrix is symetric in an undirected graph – if we sum the whole matrix we get 2\*m connections. If the graph is directed Aij != Aji, we get the number of links if summing the whole matrix.

* Simple graph : only one link between two nodes
* Multipgraph : multiple links possible between two nodes
* Pseudographs: graph that has loops (diagonal element of the adje matrix is nonzero – 2 if there is one loop, 4 if there are two loops for this node)
* Weighted graph: links have weights, weights must be non-negative (they represent distances, larger the weight, shorter the distance)

Multipartite graph : instead of adjecency matrix we have a incidence matrix B, where one class of nodes is on the rows, and the opther class of nodes in on columns.

Projections: A = B^TB – D1; A = BB^T – D2

Multi-mode: bipartite graph(two-mode network), tripartite graph (three-mode network). Higher orders are rare.

Degree:

* Degree is the number of links from a node ni, we note it with ki.
* In directed networks the number of outgoing links is ougoing degree, and incoming links are incoming degree (ki\_out, ki\_in). The classical degree is a sum of ki\_out and ki\_in.
* Average degree: sum of all degress, divided by the number of nodes (denoted by <K>)
* If the number od edges is m, number of halfedges is m/2. If we count halfedges over the nodes we get Ki (degree of the node), so we just sum over all the nodes to get 2\*m = n\* <K>
* n \*sum(Ki)/n = sum(Ki) = 2m
* in a directed graph the average in degree is the same as the average in degree
* Average degree is rarely larger than 10 (except for Facebook network), the average degree does not scale with the average number of nodes of the network

Density(ro):

* In an undirected graph: number of links divided by number of all possible links in a graph (ro = m/(n over 2) = 2\*m/n\*(n-1) = <K>/(n-1) E [0,1].
* Density ranges from 0 to 1.
* If we have a connected graph, the sparsest graph is a tree with no loops
* Real networks are pretty sparse (cca. 0.1 – 0.2) 🡪 How is this mathematcialy defined? Graph is sparse if the density goes to zero when the n nodes go to infinity. If density converges to some nonzero value, the graph is dense.
* Connection density and network size are linearly linked (higher the density, smaller the network) – adjecency matrix is not a good way to store links because it is mostly zeroes – next week we will discuss a better way to store it.

28.2.2024 – 2.predavanje

(manjka 15min)

Degree distribution:

* Degrees are not independent one of another (if one has more, the other has less) – uncorrelated is main preposition of Gauss distribution.
* Nodes with a lot of connections are called hubs.

Connectivity of a network:

* **Path** is a sequence of links betwen i and j. Links cannot repeat, (if they repeat it is a **walk** not a path).
* **Connected component** is a subset of nodes such that exists a path between each pair of nodes. If we have an edge that if removed the whole graph falls apart, we have a **bridge**.
* **,** has only one connected component.
* What are the sizes of connected component?
* Usually, we have one giant component that contains a nontrivial fraction of nodes (90 – 95%). What remains is »change«.
* If we have a directed graph, path is a sequence of directed links.
* **Weak connectivity**: weakly connected component is connected if we ignore the direction of edges
* **Strong connectivity:** we take into account the direction of the links. If the component is connected with directed edges, we have a strongly connected componnent.
* In practice we like connectivity.
* If we have a giant componnent, we expect to have only one (it is very unlikely to have two giant componnents, because it takes only one connection from the first one to the second one to obtain one giant componnent)

Distances:

* **Length of the path** is number of links of the path.
* **Geodesic path** is the shortest path between nodes i and j (we can have multiple)
* Distance between two nodes is the length of the shortest path between the two nodes.
* Distances are non-negative, withhold the triangle rule and symmetry.
* Diameter of the graph will be the largest distance between any pair of nodes in our network.
* Maximum distance is not a very robust measure, so we instead use the **average distance** (go through all pairs of nodes, count the distances, normalize the paths accordingly)
* If we have a disconnected graph: two nodes that are not connected, their distance is infinite and the average distance is useless. To avoid that we define that distance as zero – or we can only calculate the average distance from the giant connected component (zanemarimo majhen nepovezan del)
* Average distance = <d> = l^-1 (pisani L) -> ni na -1 potenco, to je zgolj kako označimo to mero razdalje
* Distancces are short, close to the average, normally distributed,
* Average distance scales with the size if the graph was just linear nodes connected.
* <d> = 6 <<n

**Clustering coefficient:**

* **Nakopičenost**: friend of a friend is also your friend (that probability is very high)
* We formaly define it as: one node i has ki neighbours. Clustering measures what is the probability that neighbors of node i are connected. That kind of connections form triangles. Clustering coefficient is the number of triangles in the neighborhood divided by the maximum possible number of triangles. Ci = ti/(ki over 2) = 2\*ti /Ki\*(Ki – 1)
* **Average clustering coefficient:** calculate the clustering coefficient for all the nodes and then average,
* Network clustering coefficient is defined as all possible triangles in the whole network \* 3, divided by number of linked triplets/connected triads. The multiplication by 3 serves as normalization, otherwise we would count each triangle three times.
* Calculating takes a while, facebook example is calculated only one a small snapshot, not on the whole network
* For high degree we expect much higher clustering coefficient that for low degree guys

How do we represent a network?

Slika, ki vsebuje besede besedilo, diagram, posnetek zaslona

Opis je samodejno ustvarjen

1. Adjacency matrix (symmetric for undirected graph)
2. Adjacency list
3. Edge list (ordered pairs of nodes for directed graph (2,1), if they are undirected use {2,1})

What do we use in practice? All three, for different purposes.

* A matrix we use if we want to derive things analytically (because of matrix multiplication).
* A list we use when we want to represent our netweok in a computer program (most algorithms require only that we can retrieve neighbours of a given node – optimal time complexity)
* Edge list is used when we want to store the network in a file (easy to add, remove, efficient storing and manipulation).

Slika, ki vsebuje besede besedilo, posnetek zaslona, pisava, številka

Opis je samodejno ustvarjen

In practice we search for a format to store both directed and undirected graph in the same form. The best for this is an adjacency list.

Storing a graph in a file:

Slika, ki vsebuje besede besedilo, diagram

Opis je samodejno ustvarjen

Where do we get the data? Online sources, present in many standard datasets, popular network repositiores/collections

## Erdos-Reny random graph

Do determine what the coefficients tell us we need a “zero” to compare to. We get that with random graphs.

Graph model is an ensemble of random graphs. We talk about a graph model with n nodes and m edges. The random graph models is a collection of all the possible graphs with those restrictions. We do reasoning over all the possible graphs.

It is an algorithm, generating a random graph, given some parameters. This gives us a baseline to realize what statistics happen at random and which do not. We also use it for generate random graphs of different sizes and densities.

Erdoš-Reny model is the baseline for random graph.

We differentiate between G(n,m) and G(n,p) models:

**G(n,m):** n is number of nodes, m is number of edges. Randomly place m links between all (n over 2) node pairs. Computationally convenient, analytically hard (O(m) = O(n<k>)).

Slika, ki vsebuje besede besedilo, pisava, posnetek zaslona, bela

Opis je samodejno ustvarjen

**G(n,p):** n is number of nodes, p is some probability that is set in advance. Place a link between all the possible random node pairs based on probability p. Computationally hard but analytically convenient (Complexity is quadratic).

Slika, ki vsebuje besede besedilo, pisava, bela, posnetek zaslona

Opis je samodejno ustvarjen

What is the expected number of edges that we will get with set n and p?

Slika, ki vsebuje besede besedilo, pisava, rokopis, vrstica

Opis je samodejno ustvarjen

What is the probability of m? P(m) =

== we have to consider all possible places \* m coins must have said “yes – make a link” \* all the others must have said “no don’t”

== ((n choose 2) choose m) \* p ^m \* (1-p)^((n choose 2) – m).

Parameter p turns out to be the density of the network. We figure that out with throwing some formulas around.

We are working with binomial distribution – a node can in theory connect to all the other n-1 nodes. Expected number of links is (n-1)\*p. Most of the nodes will have the degree similar to the average. Because of the binomial coefficient this is hard to work it. We use Poissons distribution as an approximation. It is much easier to work with, but we have to make sure that the number of nodes is much higher than the average degree. On the picture we can see how we can adjust the top formula to get the second formula which represents the Poissons distribution. There is no n in the formula! That means that we expect the same distribution, no matter the size of the network (as long as n is big enough), it is only reliant on the degree.

Slika, ki vsebuje besede besedilo, posnetek zaslona, pisava, vrstica

Opis je samodejno ustvarjen

Degree distribution plotted on a log scale is a line. According to degree distribution, real networks are not random graphs. We will discuss the reason behind this later on. In a random graph there are no hubs (nodes with a very high degree), while we do observe them in a real network.

We are analytically trying to derive what is the size of the biggest connected component. What is the probability of a random node to be a part of this connected component. **S = ns/n**

In order for node I to be a part of the largest connected component there needs to be a link to the rest of the component. There has to be a node j that will connect to the i. This link must have formed with probability p. That node must also be part of the largest connected component – the probability is S. **S = p\*S**.

There are also other nodes that we have to consider. J’ is a random node, and be careful to not “double count the porbabilities”. It becomes a nasty long formula. Instead of computing

**S = p\*S + p\*s - …..**

that I am the part of the largest connected component, I rather compute what is the probability that I am not connected to the largest connected component. **1 – S = ((1-p)+p\*(1-S)^(n-1)**

**1 – S = (1 – p\*S)^(n-1)**

**Log(1 – S) = (n-1)\*log(1 – p\*S)** 🡨 use the Taylors approximation

**Log(1 – S) = -(n-1)\*p\*S**

**Log(1 – S) = - <K>\*S**

**1 – S = e^(-<K>\*S)**

**S = 1-e^(-<K>\*S)**

The formula seems to be recursive. Again, it does not depend on the network size. We cannot isolate S out of the formula here. The only thing to do is to resort to numerical simulations. We can consider S to be two functions. f(S)=g(S). Plot the functions and see where they intersect. That is the value we are looking for. F(S) is just a linear function. G(S) is a bit more complex. First graph show how f(s) and g(S) are plotted. If G starts above the F we have a solution (intercept), otherwise we do not. When K becomes larger than one, the large connected component emerges.

Slika, ki vsebuje besede besedilo, diagram, vrstica, grafični prikaz

Opis je samodejno ustvarjen

Slika, ki vsebuje besede besedilo, posnetek zaslona, vrstica, diagram

Opis je samodejno ustvarjen

The increase is not gradual! It is (semi)instant when K reaches the value of one.

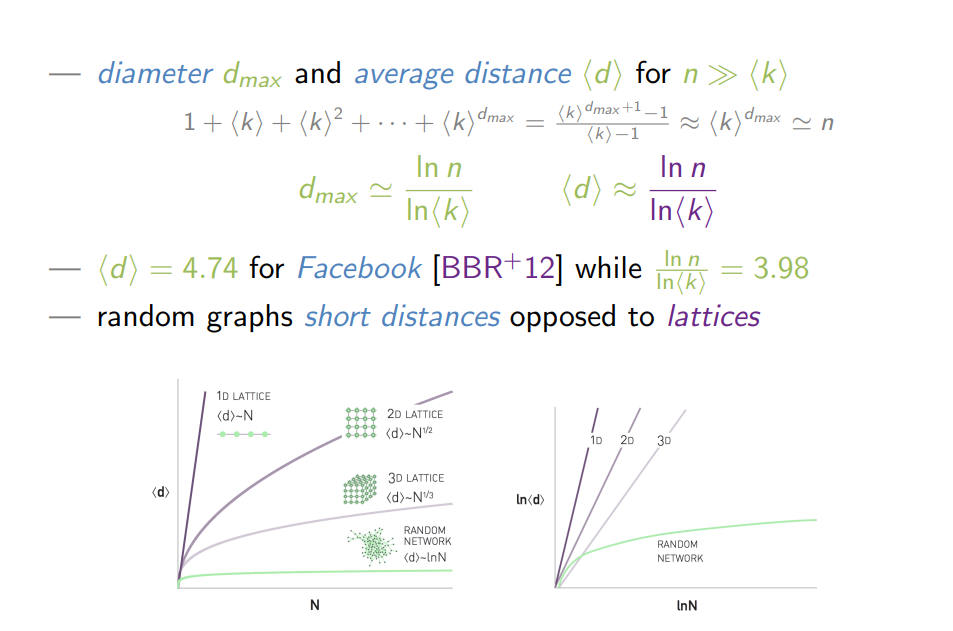
1St homework: 7 excercises, specified what you need to submit.

1. **Mathematical** induction for the first part, for the second part use logical reasoning
2. Next week
3. How to **efficiently** (in constant time) sample
4. **Weak** and strong connectivity
5. Next two weeks
6. Figure out which network is which one (by structures)
7. Who to vaccinate? Wait until next weeks

3.predavanje – 6.3.2024

\*\*manjka prvih 15min – glej ppt in slike\*\*

Graph diameter (average distance) and distance:

* We have an estimate of a diameter – very non robust measure. If we have a “ball” network with a diameter 3, and we add a long tail, the diameter will change quite a lot.
* Different scaling based on number od dimensions 🡪 average distances scale based on the dimensions 🡪 <d> ~ n^(1/D) where D is number of dimensions.
* For better approximation we use log scale
* 

Clustering coefficient in an Erdos-Reny graph:

* In G(n,p) every edge forms with probability p, which means that the expected clustering coefficient of node I is also **p regardless of the degree of the node** 🡪 all the averages and statistics are p (density of a graph)
* What is the estimated number of triangles 🡪 every triangle is formed with a probability of p.
* Random graphs do not reproduce high clustering compared to real networks.
* High degree nodes do not have high clustering compared to low degree nodes,
* Clustering for high degree nodes is underrated and overatted for low degree nodes 🡪 not a good model for ocenjevanje tega

### Configuration model

Slika, ki vsebuje besede besedilo, posnetek zaslona, pisava

Opis je samodejno ustvarjen

Another important graph model. We know that in random graphs we have Poissons distribution, however in real networks we observe power-law degree distribution.

Generate a random graph with an arbitrary degree distribution.

If we fix the degrees of the nodes, we also fix the number of nodes and number of edges. In contrast to Erdos-Reny, in the configuration model we also fix the degrees, and after that everything is random.

We are given a degree sequence, make a node with “stubs” attached to them to match the degree of the node. Then randomly select two stubs to make an edge. Nodes are paired randomly this way, however we have the number of nodes, edges and degree distribution that we wanted.

In theory we might also get a multigraph with loops. Here it’s impossible to say for certain that we will not create a multigraph. We cannot restrict ourselves to simple graphs.

Sequence of degrees is graphical: that means that it’s possible to construct a graph with nodes with such degrees. For example, it is not possible to construct a graph with degrees {1,2}. (“Handshake lemma”).

How do we derive probability p for this model of graphs? Now nodes with higher degree are more likely to be connected. Probability of nodes I and j connecting depends on the qualities of those two nodes – numbers are usually small, if it becomes more than 1, then you can think of this number as the expected number of edges between I and j not that this is a probability that a link will form.

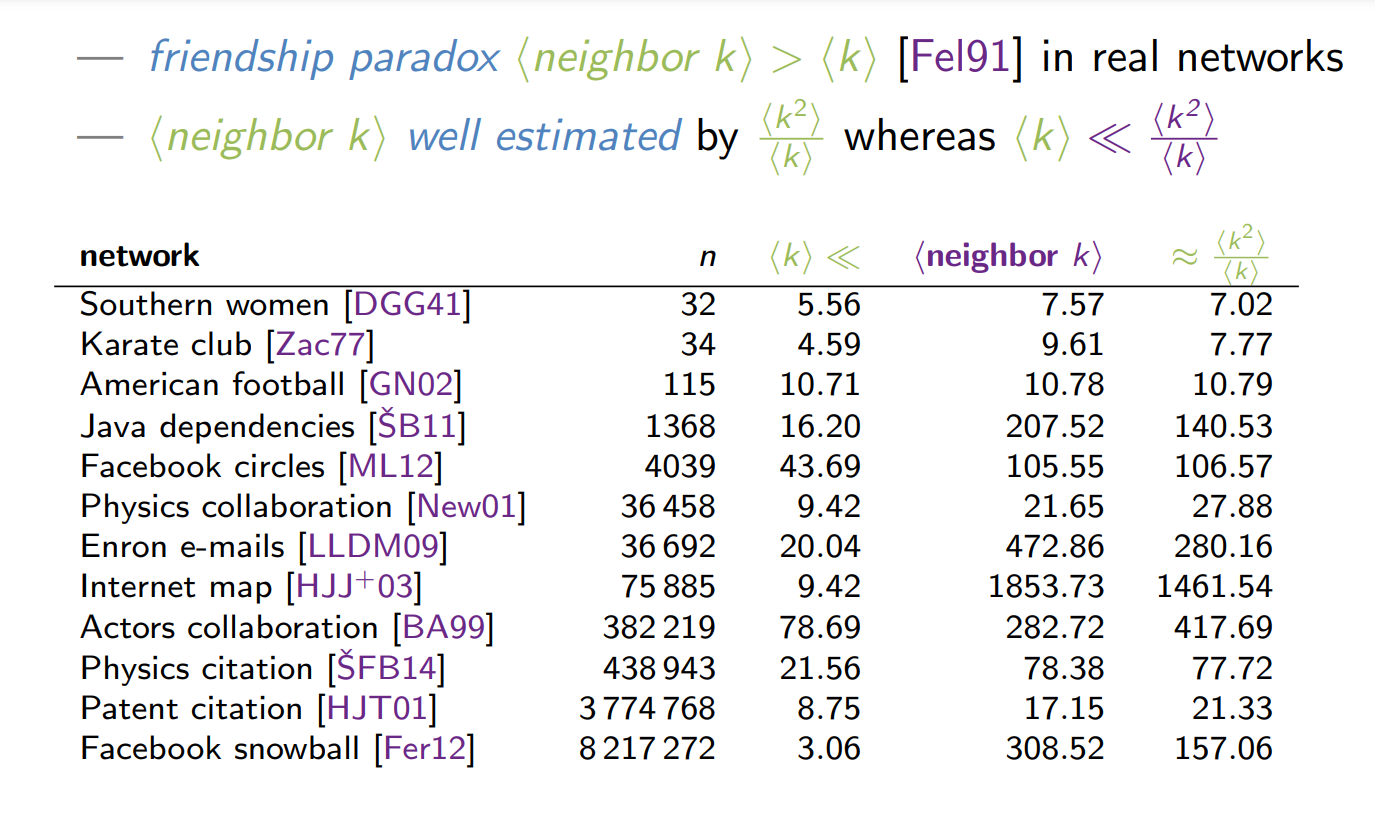
Slika, ki vsebuje besede besedilo, posnetek zaslona, pisava, rokopis

Opis je samodejno ustvarjenm pairs of half edges need to be matched.



Neighbors:

* When forming en edge with a stub from node I, the probability that the edge forms with the node with degree K is P. However there might be many nodes with such degree K. This is how we get the degree distribution of a randomly selected neighbor.
* Slika, ki vsebuje besede besedilo, pisava, posnetek zaslona, rokopis

  Opis je samodejno ustvarjen
* Standard formula for variance pops up when we turn around the inequality in the frame. When is this number larger than zero? Can it be equal to zero? Zero is only when all the values (degrees) are the same. If at least two degrees differ, then variance is larger than zero, the whole izraz is larger than zero, so the inequality holds. What does that mean? 🡪 Friendship paradox: your friends on average have more friends that you have. High degree nodes are more likely to appear in our neigborhood. 🡪 it holds in many different networks
* 

In order to derive the clustering coefficient, we introduce The excess degree distribution. What is the probability of neighbors connecting? We need to be careful 🡪 one of the links of the node is already used for the sosedstvo. **Excess degree** is simply the degree of the node minus one = Ki -1.

If we consider excess degrees instead of normal degree, the derivation is the same, only the degree is subtracted by one.

# Small-world networks

(omrežja malega sveta)

**High clustering + small distances**

(so far, we only had high clustering with long distances, or low clustering with short distances)

There was no model to reproduce what was observed in real networks - **High clustering + small distances 🡪 small world networks.**

First model that shows it is possible with such characteristics is a Small-world model. It is nor a realistic model, it only explains that these two qualities can co-exist.

We have a ring with n nodes, each node is connected to some nodes (look at the picture). Randomly rewire p % of the links. P \* m links are rewired.

Slika, ki vsebuje besede krog, besedilo, posnetek zaslona, diagram

Opis je samodejno ustvarjen

Deriving the network clustering coefficient of the network: C = 3\*triangle/number of possible triangles

* How many connected triplets we have
  + Triangles to the right
  + …..

Navigation: Does the existence of short paths imply navigable small-world decentralized search?

We need to make local decisions, because we cannot use centralized search algorithms.

Triadic closure – social sciences

# Scale-free graphs

Real networks contain hubs. Scale-free networks are networks with a power-law distribution.

The main difference between Poisson and power-law distribution is in the tail. Exponentially fast vs. polynomial fast (how fast the tail goes to zero).

Slika, ki vsebuje besede besedilo, pisava, rokopis, posnetek zaslona

Opis je samodejno ustvarjen

\*\*missing two lectures\*\*

Predavanje 3.4.2024

Communitiy detection

z-scientos slides pack

forest fire, butterfly model, copying model, citation model

**Models dynamics:** separate nodes that we visited and nodes we linked to (green ones on the slide). With this we can model more general dynamics. (look up ambassador nodes)

**Model comparison:** burning probability (how much neighb would we explore), linking probability, all models are giving sparse small world networks. Look at degree mixing (corealtion of the degrees between linked nodes), negative mixing – mixing coefficient r - (negative r means that high degree nodes connect to low degree nodes (can only be reproduced with citation model (check?)).

R is the slope of a line fitted to a plot of node degree

Slika, ki vsebuje besede besedilo, posnetek zaslona, pisava, vrstica

Opis je samodejno ustvarjen

**Directed models:** we now have 4 probabilities instead of 2. We have a forward and a backward burning probability. We also have two linking probabilities (are we linked to the ambassador nodes or not and how many neighbors of the ambassador we will link to).

Random model and citation model work on same princips, but there is a difference (check what it is).

Slika, ki vsebuje besede besedilo, diagram, grafični prikaz

Opis je samodejno ustvarjenAll models fit the real network quite well (following the power law with in-degree distribution

Biggest difference can be observed in the clustering coefficient (what % of nodes has a certain clustering coefficient). Citation model fits the real data surprisingly well. However the forest fire doesn’t. Only the citation network fits the real data on all parameters (degree distribution, clustering coefficients …).

Now our model looks like G(n, p\_forward, p\_backward, q\_ambassador, q\_link). We can fit these parameters to fit the real data. The parameters are meaningful and they tell us some interesting info:

Slika, ki vsebuje besede besedilo, posnetek zaslona, pisava, številka

Opis je samodejno ustvarjen

* We are citing more papers than we used to (Number of papers cited is average out degree)
* We only read 20% of the papers we cite
* How do we discover new papers? Following links in the right direction to get predecesors of the paper. If we follow the links of the parent in the wrong direction we get papers who cite the same papers that we cite
* Slika, ki vsebuje besede skica, oblikovanje, oprema za kuhinjo, postelja

  Opis je samodejno ustvarjen
* Blue we read, green we just cited, we read only a small fraction, however they are connected so the situation is not as bad. It is true we only read 20% of the papers we cite, but in truth for each two papers cited we will read one.

**COMMUNITY structure**

*How to cluster/group nodes in a network in a meaningful way – into communities.*

* Zachary karate club network: who is friends with who. Club is split into two part because of an falling out between two members. We can see that two groups were forming even before the argument. The most famous network in the literature for community/graph partitioning algorithm – a test if the algorithm finds the two groups. The nodes are colored based on the color of the majority of the neighboring nodes. If we have a tie you randomly choose one.
* Most real networks (especially social networks) contain communities.
* Communities cannot be found in technological networks (due to limitations) for example roads or Java classes (one class can only have one parent class etc we cans till see some clusters : algorithms, data representation, input/output, visualization).
* Why do we believe that clustering is important for real networks

Slika, ki vsebuje besede besedilo, posnetek zaslona

Opis je samodejno ustvarjen

* Assume weights on edges, strong ties have bigger weights, more information flow on those ties (family vs. acquaintances).
* Bridges: formally a bridge is a link, if I remove it, the network falls apart. If I remove link I and j , distance between them becomes infinite 🡪 link I j was a bridge. That is very strict, we introduce a local bridge, if removed the distance between I and j becomes strictly larger than 2 🡪 this link was not a part of the triangle, which makes it a local bridge.
* Weak ties are local bridges that span communities
* Triadic closure 🡪 triangles with strong connections will close 🡪 local bridges must be weak, otherwise it will form a triangle under triadic closure, if it Is a part of a triangle it cannot be a local bridge.
* Homophily: similar people are more likely to connect (birds of feather flock together)
* Assortative mixing: connected nodes have similar properties

Slika, ki vsebuje besede besedilo, posnetek zaslona

Opis je samodejno ustvarjen

Slika, ki vsebuje besede besedilo, posnetek zaslona

Opis je samodejno ustvarjen

Defining a community as a clique is too strict : we would need too many connections, because all the nodes must be connected to all other nodes. We must relax the definition.

* K-plex: group of n nodes 🡪 k-plex says that everybody connects to at least n-k other nodes. 1-plex is the same as a clique.
* K-cores: set of nodes where everybody has at least k neighbours. How would we detect a 10-core? Eliminate all nodes that have degree < 10. When we are removing these nodes, the degree of other nodes will also decrease. What is left is a 10-core. Cores are nested – k+1 core is a subset of a k-core.
* K-cliques: in a clique everybody is on distance 1. In a k-clique everyone is on distance of at most k.
* K-clubs: k-clique where all the shortest path is included,
* K-klans: k-clique where all the shortest path is included, they differ based on the way they are made.

**Community is a dense subgraph of a sparse network.**

Is a community weak or strong? Internal and external degree of a node. Imagine a node in a group. It has some links to nodes also in a group and some to nodes outside of the group. The number of links of a node going out of the group is the external degree, while the number of links of the node to the nodes in the group is called and internal degree. A strong community is where all the nodes have higher internal degree than external degree. A weak community only looks at the average of internal and external degrees (not at every node, but at average).

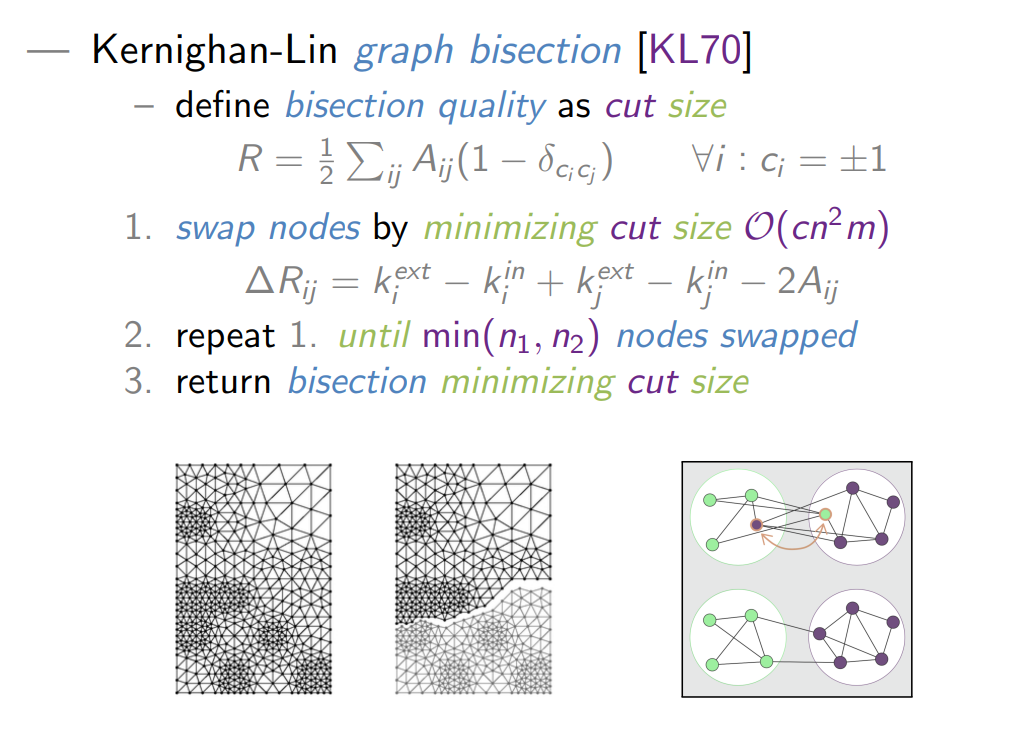
In a network we found some partitioning – 3 groups. We would like to measure how good this partitioning is. Measure the number of links inside of the communities and substract the expected number of links in those communities based on a random graph model. Q = #links inside - #links expected inside. Definition of modularity. Ci = group identifier of group i. Aij is an element in the adjacency matrix. Go thru all the pairs, see how many edges we have inside and how many we would expect at random. For this we usually use a configuration model. To keep the Q number on some good value, we need to normalize it by number of links. Q € [-1/2, 1] 🡪 negative value is anticommunity structure, 1 is a community (look at examples below).

Slika, ki vsebuje besede besedilo, pisava, rokopis, vrstica

Opis je samodejno ustvarjen

\*\*missing 25mins\*\*

Graph partitioning is not the same as community detection! In partitioning we give a parameter how many groups we want, while with community detection we give no parameters.

**Bisection: Kerighan-Lin graph bisection:** We have a graph that we want to divide into two groups with given sizes n1 and n2, where n1+n2= n, with cutting as little links as possible. Cut size R = 5 means that 5 links were cut to divide the graph into two partitions. Inefficient because we need to check all the links and swapped nodes. Into three groups: bisection two times.

Slika, ki vsebuje besede besedilo, posnetek zaslona, pisava, oblikovanje

Opis je samodejno ustvarjen

* **Spectral partitioning: Fiedler graph bisection:**
* **Block modeling partitioning: standard equivalence blockmodeling**

Hierarchical clustering

Slika, ki vsebuje besede besedilo, posnetek zaslona, diagram

Opis je samodejno ustvarjen

* Modern approaches to community detection:
  + Hierarchical clustering:

Slika, ki vsebuje besede besedilo, posnetek zaslona, diagram, pisava

Opis je samodejno ustvarjen

* Slika, ki vsebuje besede besedilo, diagram, posnetek zaslona, vrstica

  Opis je samodejno ustvarjen
* Slika, ki vsebuje besede besedilo, posnetek zaslona, pisava, diagram

  Opis je samodejno ustvarjen
* For each node check the color of the neigbors, color the node to maximize modularity. Merge the groups into supernodes. Now optimizing modularity is the same as in the original graph – we can simply project the partitioning back on the original graph.
* Slika, ki vsebuje besede besedilo, posnetek zaslona

  Opis je samodejno ustvarjenLeiden is faster, returns with higher modularity, it has quality guarantees that it will return connected communities.
* Slika, ki vsebuje besede besedilo, posnetek zaslona

  Opis je samodejno ustvarjen

8th lecture:

Popular approaches for community detection:

* Hierarchical clustering:
  + Bottom up: you need node similarity (in this case the node similarity is the degree of the node), define similarity between clusters – average linking: average similarity between nodes in the two clusters (average the similarity of all pairs) 🡪 slow, you need to decide how many clusters you want (hard problem in practice)
  + Top down: compute the edge between all links, cut the link with the highest betweenness, if the network falls apart, we have two clusters. 🡪 works well, however the complexity Is cubic.
* Luvain algorithm: for each node locally decide in which group you put it, to optimize the modularity. In a locally stable partition, merge every cluster/group into a super ndoe/super network, where the comuunities are now nodes. Then run it again on this second pass. Watch that the number of links between super nodes is appropriate to the initial network. 🡪 Linear, can be used for real/big networks.
* Slika, ki vsebuje besede diagram, zemljevid, vrstica, besedilo

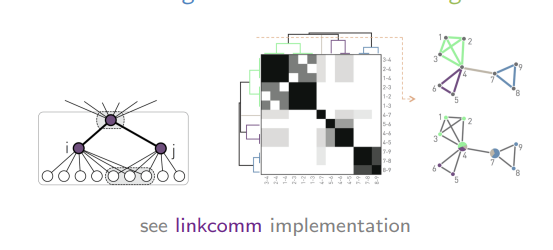
  Opis je samodejno ustvarjen
* Leiden algorithm: Improved Louvain algorithm (it sometimes returned disconnected graphs). 🡪 it offers guarantees.
* Slika, ki vsebuje besede risanka, ilustracija, vzorec, umetnost

  Opis je samodejno ustvarjen s srednjo stopnjo zanesljivosti
* Infomap: describe the walk of a random walker, you know when you enter and when you leave any community. We are suing Huffman coding
* Slika, ki vsebuje besede besedilo

  Opis je samodejno ustvarjen
* Label propagation: Raghavan label propagation: super fast, not robust or reliable, returns different strctures on different runs. For every run, every node determine their neighbourhood. Randomly shuffle, traverse again, until nothing is changed, until every node is locally optimally assigned. Groups can be disconnected in theory.
* Slika, ki vsebuje besede besedilo, risanka, oblikovanje

  Opis je samodejno ustvarjen
* We are optimizing the number of links in the communities.
* Slika, ki vsebuje besede diagram, origami, oblikovanje

  Opis je samodejno ustvarjen**Palla clique percolation:** Set a parameter k (3,4,5), find all the three nodes clicks in the netwrosk (find all the triangles). Merge the cliques into super nodes, connect them if they are adjecend (if they share k – 1 nodes). Now in the “new” graph connect the cliques, if they are adjecened in the original graph. Return these communities. Project them back to the original graph. Now the groups can overlap. Not computationally efficient (all k node clicks must be enumerated). Probability of a link between two nodes in ER graph is p = <k>/n-1. If <k> is larger than 1, one large connected component is ensured. Links are cliques on two nodes. We are actually looking for connected components. We don’t want just one conn.comp., we want more communities. If k is too small, we will only find one group (it will percolate). However if k is too big, we will not find anything (check!).
* Ahn link clustering: first link clustering algorithm 🡪 cluster based on links not on nodes. Clusters on nodes can be overlapping. Simple hierarchical agglomerative clustering. We only consider links that share a single node, define the similaritiy between these two links, as the ajccard coefficient of the neighbourhoods of I and j. gamma I + is the union od the neighbour hood of I (gamma i) plus the I node itself. Now to calculate the number of similarity between the clusters: single linkage (max similarity between two clusters). Bottom- up merging links into cluster, merge clsuters, get dendogram and cut it at some level. Solves overlapping and one other thing, not timely efficient (we must copmute the similarity between all the pairs od links).
* Slika, ki vsebuje besede besedilo, pisava, vrstica, posnetek zaslona

  Opis je samodejno ustvarjen
* 
* Many algorithms because it is a poorly defined problem.
* Community measures: check the degree of I inside the community (internal degree K, average internal degree of the node, we want it high), external degree of node I (connections outside the community – expansion E). Flake score F combines K and E into a single score 🡪 how many nodes have more external links than internal links. We want F to be very small (around 5%).
* 
* NMI: normalized mutual information: mutual information divided by the average of the entropies of the averages of the posamezne communities.
* NVI : normalized variation of information [0, logn] , measure of distance, smaller is better. It is a proper emasure of distance, nonegative, symmetric, triangle equality holds.
* Other measures exists: ARI-adjusted running index
* Benchmarks: construct syntetic graphs, plan communities so we know what the algorithm should detect. Girvan-Newman synthetic graph, mi mixing parameter is the fraction of links that are outside. If mi is 0, the cimmunities are not connected. When increasing mi, at some point the algorithm cant detect the communities anyxmore. It is not realistic: groups are the same size (not like that in real network s)and the degree distribution doesn’t follow a poer law.--> solution: Lanchichinetti synthetic graphs: constructed graph has a powerlawe degree distribution, size distribution of the communities follows a power law of the power of 1 or two.

Approaches that detect general groups of nodes (not necessarily comunnities):

* Blockmodeling
* Stochastic block model
  + Plot the adjacency matrix, dot where there is a link, rearrange the rows and columns so that we find a nice structure. There will be a structure on the diagonal if the groups are communities.
  + Standard structural equivalence must be normalized (divided by the number of nodes) 🡪 how can we do that without penalizing the nodes with few neighbours? Consider the neighbourhood of the nodes as a vector (take two node vectors from adjacency matrix, make a dot product)
  + Slika, ki vsebuje besede pisava, besedilo, rokopis, vrstica

    Opis je samodejno ustvarjen
  + Expected number of common neighbours of I and j, that have external degrees Ki and Kj. Kj / n-1 \* Ki 🡪 Kj \* Ki /n
  + Slika, ki vsebuje besede besedilo, pisava, posnetek zaslona, rokopis

    Opis je samodejno ustvarjen

Regular equivalence: they don’t need to have the same neighbours, however their neighbours must be equivalent.

Chronical delta of I and j prevents node not being equivalent to itself (which we believe should be equivalent).

SBM (stochastic block model) - a couple of ER graphs, extended to groups, connected

We want to use this approach to see how likely is a certain partition, observing a particular network

With nested methods we lower the threshold where we are not able to find groups anymore.

Structures:

* Communities
* Bipartite
* Core-periphery
* Random
* Slika, ki vsebuje besede posnetek zaslona, oblikovanje

  Opis je samodejno ustvarjen
* Slika, ki vsebuje besede besedilo, diagram, oblikovanje

  Opis je samodejno ustvarjen
* Slika, ki vsebuje besede besedilo

  Opis je samodejno ustvarjen
* K-shell are nodes in k.core that are not part of k+1 core.
* Kshells forma partititioning of a network.
* Which k to take? K\* such that it maximizes the closeness centrality