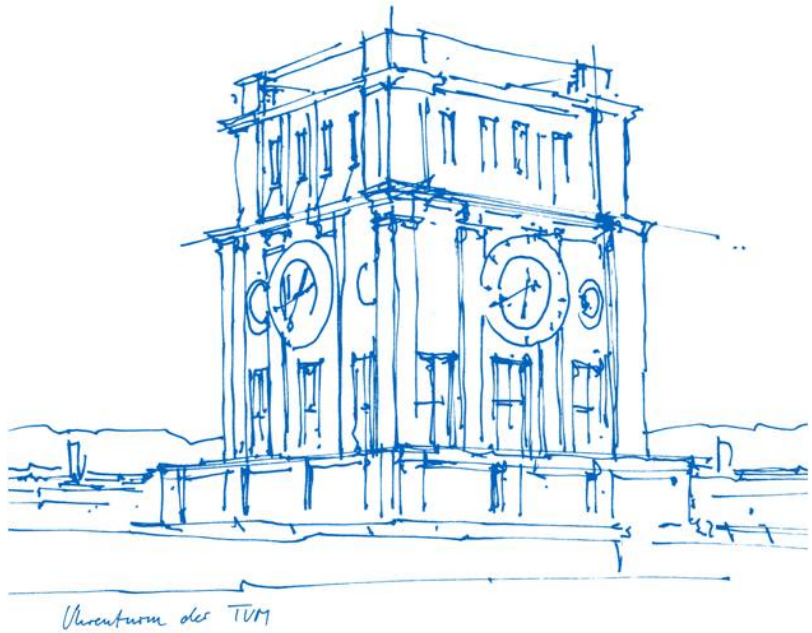
Exercises for Social Gaming and Social Computing (IN2241 + IN0040)

Exercise Sheet 2

Topic: Centralities



Exercise Content

Sheet Number	Exercise	Data Gathering	Deadline
0 (prep)		• Install software (python and libraries) following instructions • Install mobile data app • Register on our platform	
1	• Introduction to Python: basic Python programming language exercises • Graph Drawing using igraph	• mobile data gathering (nothing actively to do) • social context in group rec experiment: complete questionnaires, form groups, rate restaurants etc.	• **Sunday, May 27, 24:00**
2	• Centrality measures	• mobile data gathering (nothing actively to do) • social context in group rec experiment: complete questionnaires, form groups, rate restaurants etc.	• **Sunday, June 3, 24:00**
3	• Recommender Systems as an example for systems using simple forms of social context: Collaborative Filtering	• mobile data gathering (nothing actively to do) • social context in group rec experiment: complete questionnaires, form groups, rate restaurants etc.	• **Sunday, June 10, 24:00**

Exercise Content

Sheet Number	Exercise	Data Gathering	Deadline

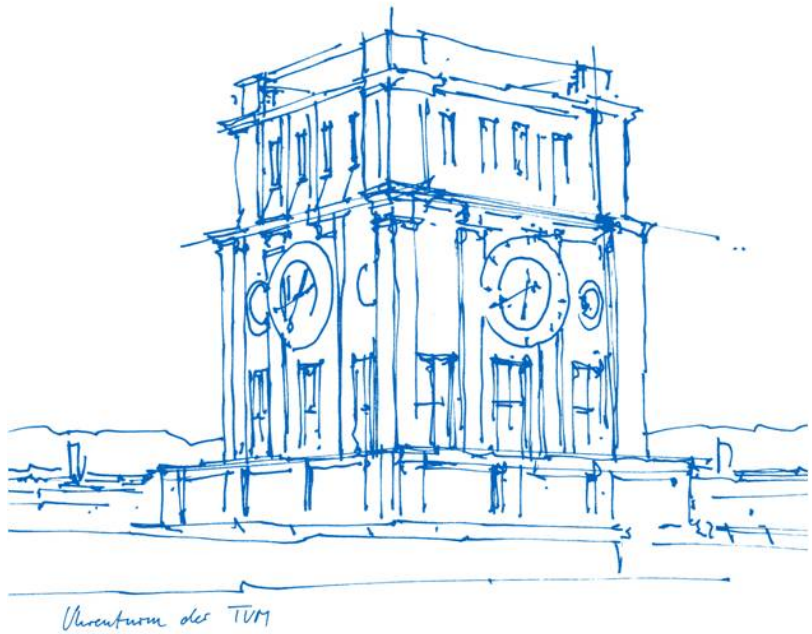
4	• Clustering:
	  • metric: K-means Clustering
	  • networks: Girvan-Newman-Algorithm
		• mobile data gathering (nothing actively to do)
		• social context in group rec experiment: complete questionnaires, form groups, rate restaurants etc.
			• **Sunday, June 17, 24:00**

5	• Group Recommender Systems • Social Context in Group Recommender Systems	• mobile data gathering (nothing actively to do)	• **Sunday, June 24, 24:00**

6	• Analysis of mobile Data: Paper: N. Eagle and A. Pentland: “Reality mining: sensing complex social systems”. Pers. Ubiq. Comp. 10, 4 (2006):
	  • Compute behavioral entropies
	  • Compare mobile network with long-term network
			• **Sunday, July 1, 24:00**

7 (essay)	• Essay (only if not participated in data gathering) :
	  • style of scientific paper / seminar paper (no personal opinions etc.)
	  • >= 2500 words (excluding citations)
	  • topic: Privacy in Social Media
			• **Sunday, July 16, 24:00**

Data Collection Part I and II: Reminder



Data Collection: Part I: Social Context in Group Recommender Systems (data for sheet 5)

deadline: Sunday, June 17

detailed steps and todos: see "exercise0" presentation

- step 1: register at https://vmschlichter24.informatik.tu-muenchen.de . important: check the data collection consent declaration.
- step 2: do Thomas-Kilman Conflict Model test (same platform)
- step 3: individually review 5+ restaurants
- step 4: form a class-internal group (3+ members)
- step 5: each member of internal group: provide social context: rate other group members
- step 6: formally create internal group, elect group persona
- step 7: sit together, review 5+ restaurants (as a group (internal group))
- step 8: form a class-external group (3+ members). you are automatically the group persona for the external group
- step 9: each member of external group: do steps 1, 2 and 3
- step 10: each member of external group: provide social context: rate other group members
- step 11: formally create external group, elect group persona
- step 12: sit together, review 5+ restaurants (as a group (external group))

Data Collection: Part II: Mobile Data (data for sheet 6)

detailed steps and todos: see “exercise0” presentation

1. Install app at Android or iOS device
2. Enable permissions on mobile device (may technically also ask for access to your contacts, app will however NOT collect this data)
3. Scan QR code to participate at user study

4. Register at https://vmschlichter24.informatik.tu-muenchen.de . important: check the data collection consent declaration.
5. Enter generated Device ID during registration

Data Collection

- Personal data will be **anonymized** before any processing
- We gather:
  - **part I:**
    - **Personal Data:** Full name, matriculation number (if student) - Date of birth - country - email address - device ID (mobile data collection experiment) - Coordinates of the main place - Thomas-Kilmann conflict model test data
    - **Social network data:** Trust, Tie strength, relationship strength, personal similarity, social context similarity, level of sympathy, social hierarchy, domain expertise
    - **Individual Restaurant ratings**
    - **Group Restaurant ratings**
  - **part II:**
    - **Location via GPS, network**
    - **Bluetooth environment**
    - **Cell id localization**
    - **Association with Wi-Fi networks**
    - **Environment sensors**
    - **acceleration, air pressure, magnetic field, temperature**

Permission for Data Collection (Part I and II)

This is a voluntary consent to contribute your data to research and teaching activities of Safey Halim, Michael Haus, Leonardo Tonetto, Georg Groh, and Jörg Ott (all Faculty of Informatics, TU-München). With your permission, your data will be collected, processed, and used for the following purposes:

Purpose 1a: Research conducted in the scope of the PhD thesis work of Michael Haus and Leonardo Tonetto on a common volume of data, including, but not limited to, mobility modeling and predictability, and private proximity testing.
Purpose 1b: Research conducted in the scope of the PhD thesis work of Safey Halim on social context in group recommender systems.
Purpose 2: Provide anonymized versions of the data to the registered students in the voluntary exercises of the class IN0040 Social Gaming / IN2241 Social Computing (SS2018, TUM, Faculty of Informatics) to be analyzed in the exercise sheets 5 and 6.

We will collect data during the months May and June 2017 with the help of a mobile phone app. This app collects the following sensor data from your phone and transmits it when a Wi-Fi connection is available:

- GPS location
- Cell id localization
- Bluetooth environment
- Environment sensors: acceleration, air pressure, magnetic field, temperature

We will during the months May and June 2017 further collect

- Personal Data: Full name, matriculation number (if student) - Date of birth - country - email address - device ID (mobile data collection experiment) - Coordinates of the main place - Thomas-Kilmann conflict model test data
- Social network data: Trust, Tie strength, relationship strength, personal similarity, social context similarity, level of sympathy, social hierarchy, domain expertise
- Individual Restaurant ratings
- Group Restaurant ratings

from students within the class and selected persons outside the class which are chosen by students in the class.

Please note that, while we do not store any personal information, this data could bear enough information to make you identifiable.

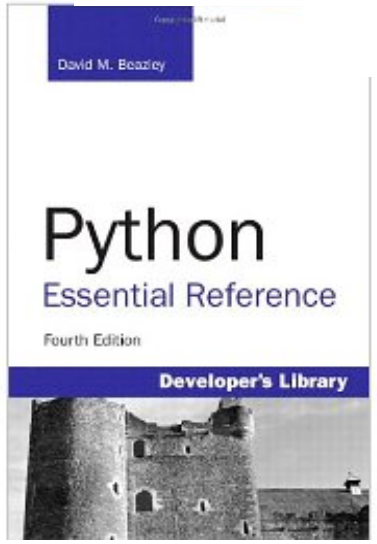
The data will be stored until 30.12.2020. Your personal data will be collected, processed, and used in the context of the aforementioned objectives in accordance with the Bavarian Data Protection Act (BayDSG).

The collection, processing, and use of your data take place on a voluntary basis. You can revoke your consent at any time without any adverse consequences. Please send any notice of cancellation to:
Technische Universität München, Research Group Social Computing I11; Boltzmannstr.3; 85748 Garching, E-Mail: grohg@in.tum.de
In the event of cancellation, your data will be deleted upon receipt of your notice.

Python and IPython Books

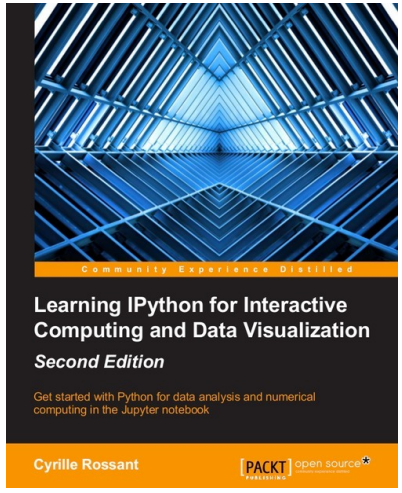
- **Learning Python:**
  Python Essential Reference (2012)
  by David M. Beazley, Safari Books
  (especially **chapter 1: A Tutorial Introduction (25 pages))**

  free eAccess: https://eaccess.ub.tum.de/login



- **Learning IPython / Reference for IPython:**
  Learning IPython for Interactive Computing and Data Visualization (SECOND EDITION) by Cyrille Rossant, 175 pages, Packt Publishing, October 25 2015
  (Especially (free) chapter 1.4. A crash course on Python)

  free access: http://nbviewer.ipython.org/github/ipython-books/minibook-2nd-code/blob/master/chapter1/14-python.ipynb
  (do not try to open this ipynb with Jupyter directly. Instead, download all the ipynb's from the book from Github: https://github.com/ipython-books/minibook-2nd-code → 14-python.ipynb)

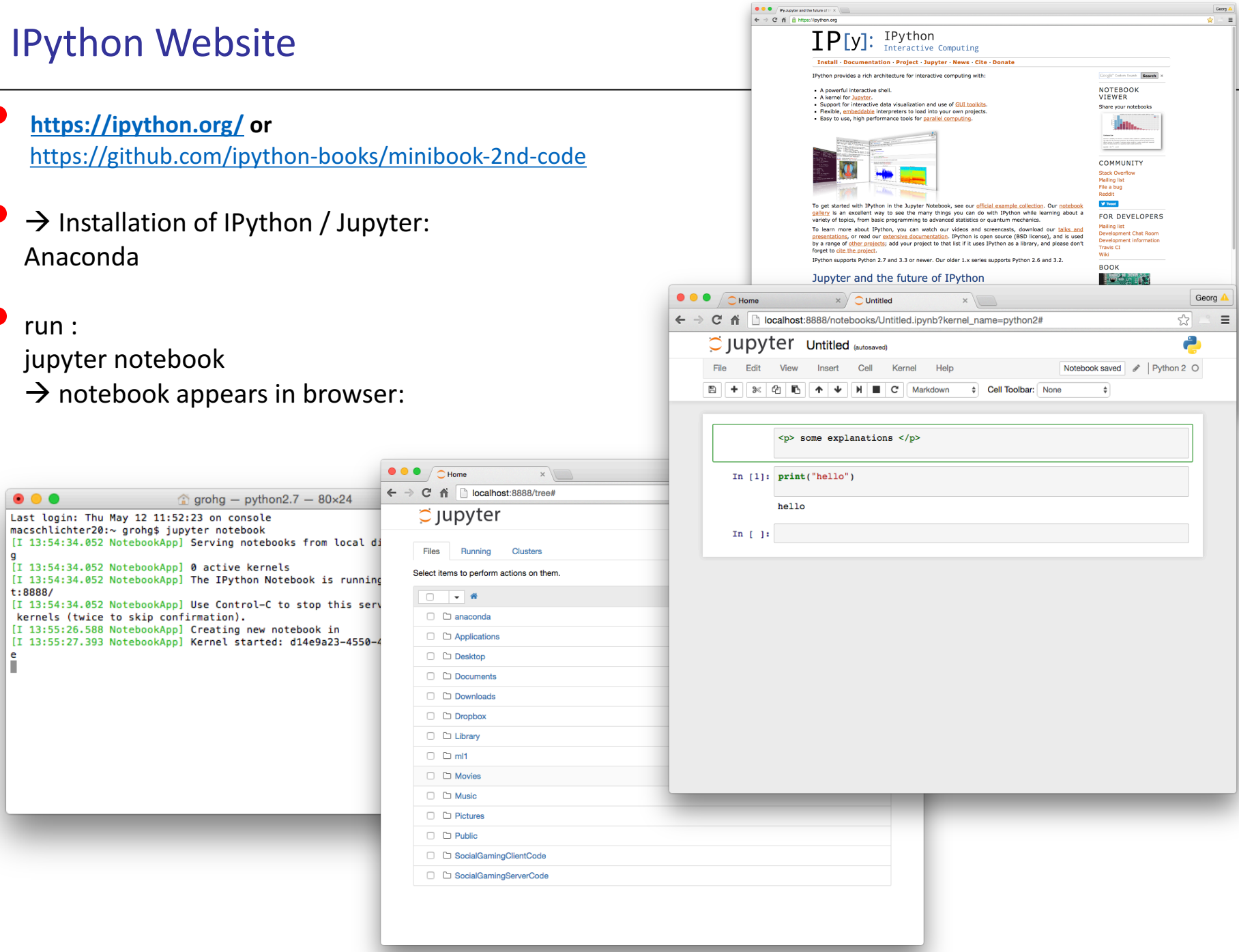


IPython Website

- **https://ipython.org/ or**
  https://github.com/ipython-books/minibook-2nd-code

- → Installation of IPython / Jupyter:
  Anaconda

- run :
  jupyter notebook
  → notebook appears in browser:



Identifying the most important nodes in a graph – brief introduction to centrality measures

- In this week‘s exercise you **calculate centrality measures**
- You will work with **three different graphs** extracted from **our social networking platform**

Degree centrality

Number of connections of a node to other nodes in the graph

$$C_D(v) = \deg(v)$$

Closeness centrality

The inverse average **length of the shortest path** between the node and all other nodes

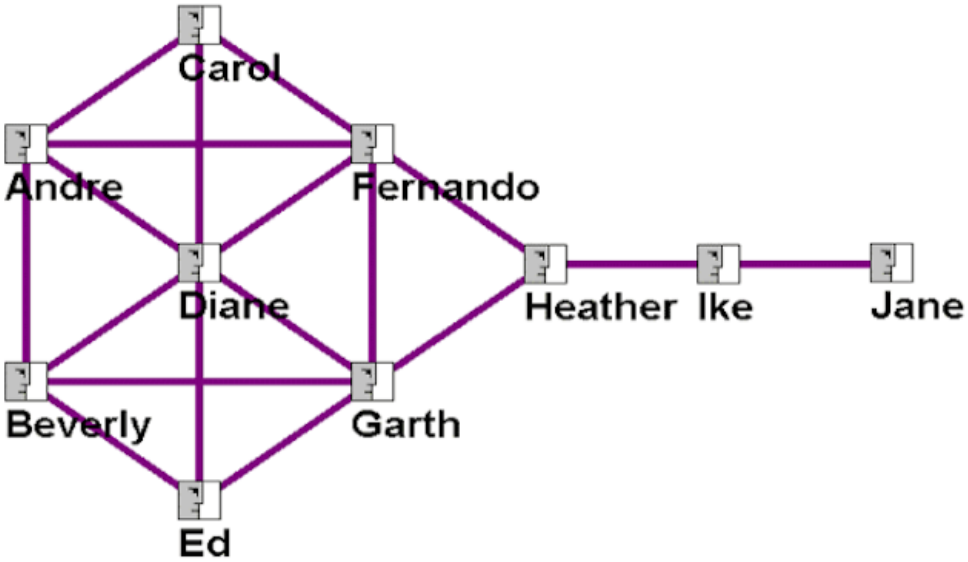
$$C(x) = \frac{N-1}{\sum_y d(y,x)}$$

Betweenness centrality

Number of times a node **acts as a bridge** along the shortest path between two other nodes + Normalization

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

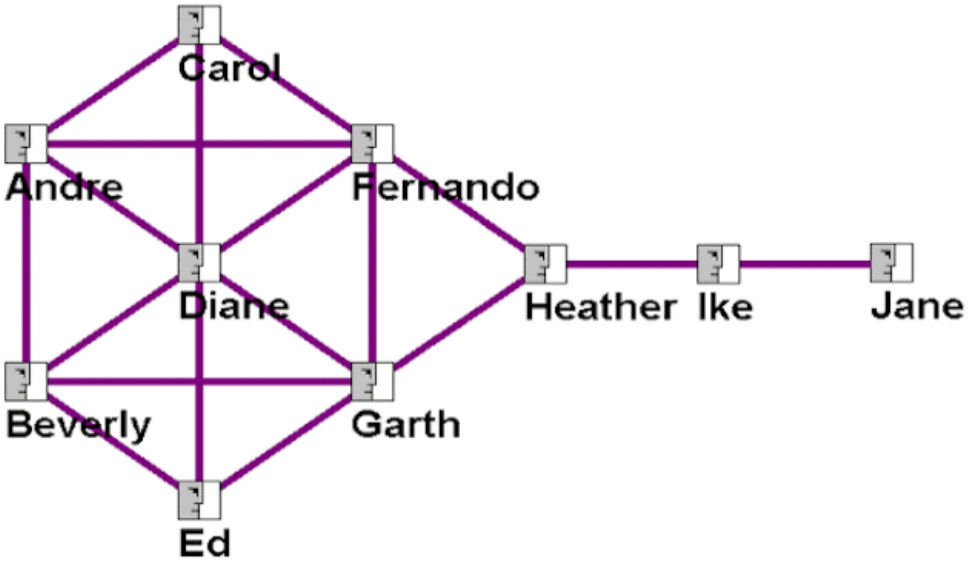
Degree centrality counts the direct connections



Name	Degree
Andre	4
Beverly	4
Carol	3
Diane	6
Ed	3
Fernando	5
Garth	5
Heather	3
Ike	2
Jane	1

- Degree centrality is the **number of connections** a node has
- Diane has the **most direct connections in the network**, making hers the most active node in the network
- However, she only connects people that are already friends with each other

Closeness centrality shows how close nodes are to others

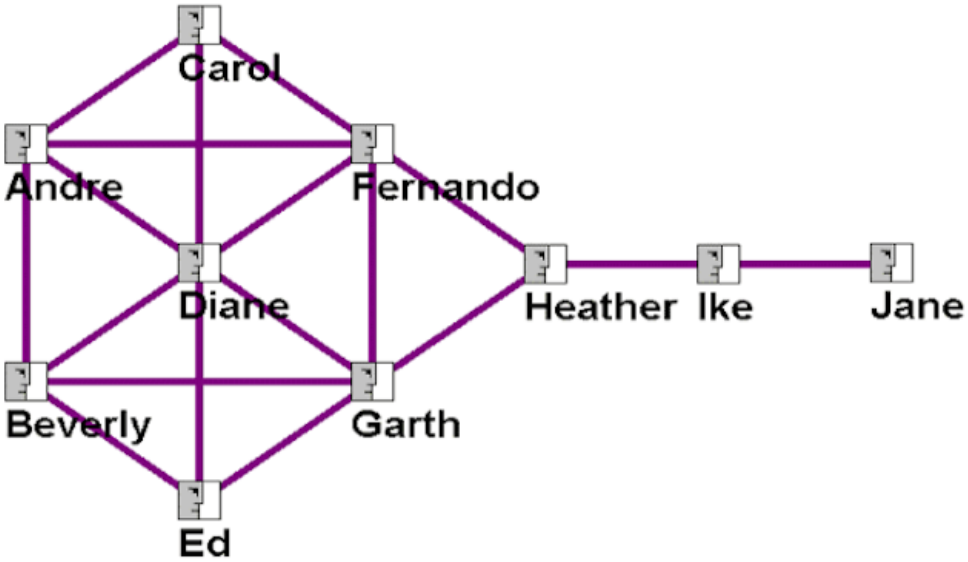


Name	Closeness
Andre	0.529
Beverly	0.529
Carol	0.500
Diane	0.600
Ed	0.500
Fernando	0.643
Garth	0.643
Heather	0.600
Ike	0.429
Jane	0.310

- The closeness centrality is the inverse of the average **shortest path to all other nodes** in the graph
- Fernando and Garth can **access all the nodes in the network more quickly** than anyone else
- They have the shortest path to all users

SOURCE: http://www.orgnet.com/sna.html

Betweenness centrality takes the position along shortest paths into account



Name	Betweenness
Andre	0.833
Beverly	0.833
Carol	0
Diane	3.667
Ed	0
Fernando	8.333
Garth	8.333
Heather	14
Ike	8
Jane	0

- The betweenness centrality measures the **number of times the shortest path between two nodes goes through the investigated** node, divided by the total number of shortest paths between the two nodes
- Heather has few direct connections, yet she has an important role for Ike and Jane, who wouldn't be connected to the network without her
- She has **high control of information flow**

Exercise 2 – Introduction Problem

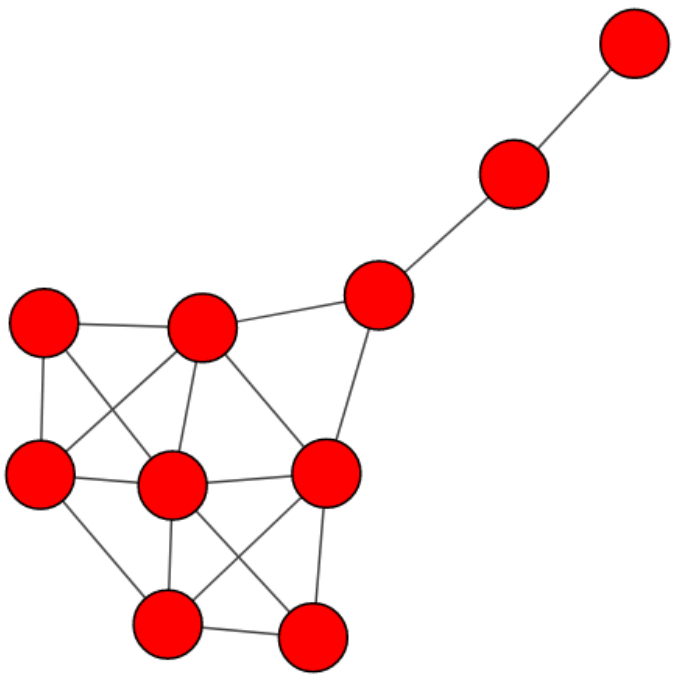
Introduction Problem: The Krackhardt Kite Graph

We will use the Krackhardt Kite for the first exercise. As you know from exercise 1, the Krackhardt Kite is a simple connected graph: unweighted and undirected. The following figure illustrates the Krackhardt Kite.

Plot the graph to make sure that all packages are correctly installed. Then calculate the degree centrality of the Krackhardt Kite Graph (just a list of 10 values - one for each node). You can use the implemented function of the igraph library.

Optional: Look at the graph and the list with the degree centrality values. Can you identify which node has which degree centrality?

```
Degree Centrality Kite: [4, 4, 3, 6, 3, 5, 5, 3, 2, 1]
```



Degree centrality

- Number of links of a node

Your code

- Plots the kite (already given)
- Prints degree centrality of kite (1 line)
- *Optional*: also print closeness and betweenness centrality

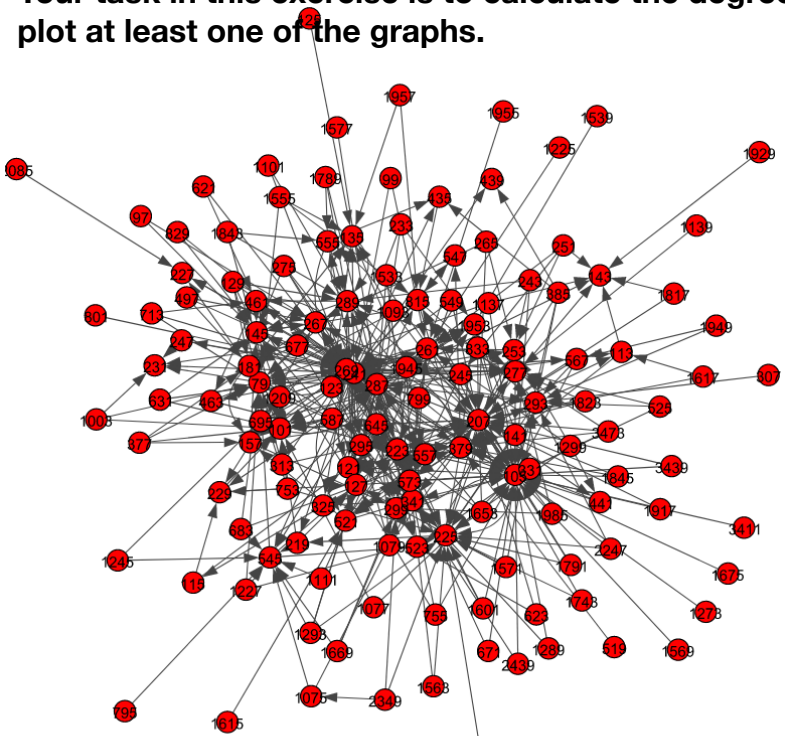
Problem 1.1 – Degree centrality of a social network

We work with an anonymized social network represented by the three files:

FriendshipNetwork.graphml,
FriendshipNetwork_Like.graphml,
FriendshipNetwork_Comment.graphml

The nodes in all of these graphs are user profiles. The edges are the friend relationships (FriendshipNetwork.graphml), the exchanged likes (FriendshipNetwork_Like.graphml), and the comments written to others (FriendshipNetwork_Comment.graphml).

Your task in this exercise is to calculate the degree centrality of all the nodes in the three graphs. Also plot at least one of the graphs.



Comment graph

The graphs

- Friendship graph
- Comment graph
- Like graph

Your code

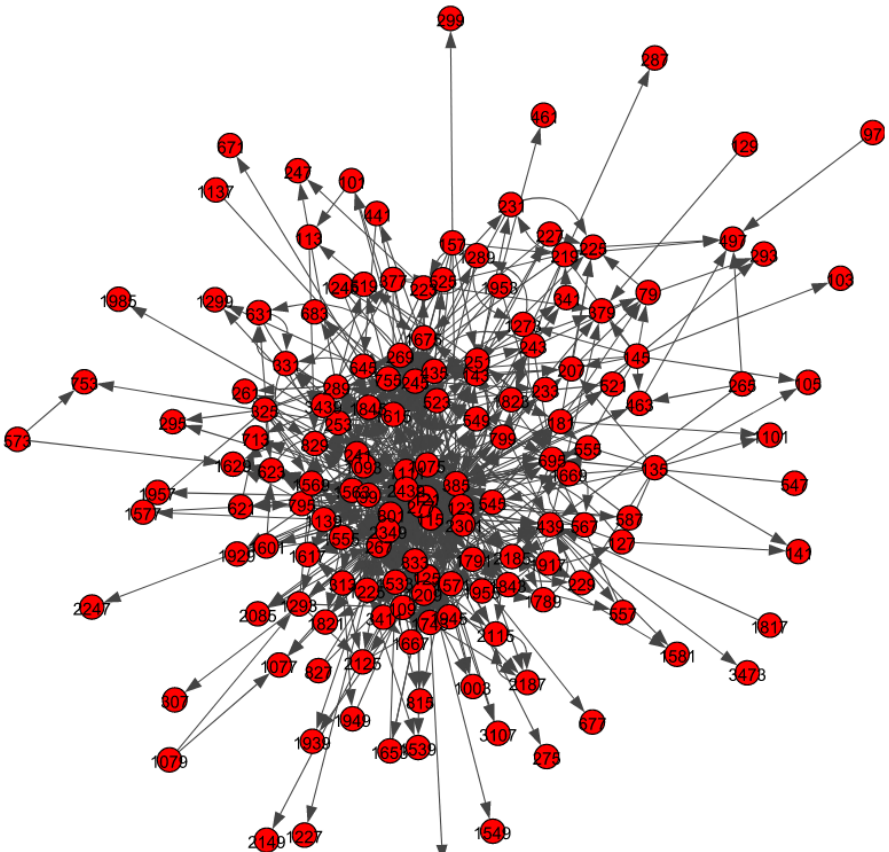
- Print degree centrality of all three graphs
- Plots one of the graphs

You will see that the degree centralities for all three graphs is quite different because the behavior of the nodes varies (different comment, like patterns)

Problem 1.2 – Closeness centrality of course network

Write a python program that computes the closeness centrality for each node in the Like-Graph (FriendshipNetwork_Like.graphml) from our social network!

The output should be a list where each item contains the value of the closeness centrality of a node.



Like graph

The graph

- Like graph

Your code

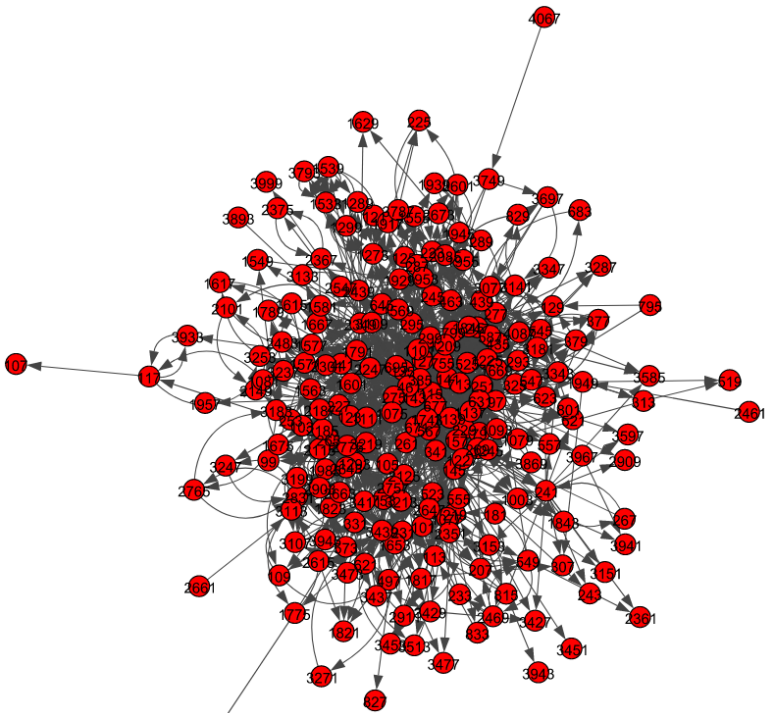
- Prints closeness centrality of like graph
- Does not use the function closeness()
- Takes the weights of the edges into account

Problem 1.3 – Betweenness centrality of course network

This time you are working with the FriendshipNetwork.graphml.

Calculate the betweenness centrality with the help of the pre-defined functios of the igraph library. Interpret the resulting values based on two exemplary nodes.

To do that pick two nodes and explain how their betweenness centrality links to the graph structure. Name the two nodes that you discussed (and their betweenness centrality). (Do not write more than 5 sentences)



Friendship graph

The graph

- Friendship graph

Your code

- Calculate betweenness centrality
- You can use predefined functions
- Discuss/explain the betweenness of two nodes

Exercise – Submission via Moodle

- Submit your finished .ipynb iPython notebooks via Moodle

- there is one Moodle course instance for IN0040 (Social Gaming) and another Moodle course instance for IN2241 (Social Computing)

- Registration in Moodle ←→ register in TUM-Online as a participant for IN0040 (Social Gaming) (Games Engineering students) or as a participant for IN2241 (Social Computing) (other students).

- **Deadline: Sunday, June 3, 24:00**

Citations

(1) [Beazley 2013) David Beazley: Python Essential Reference, Safari Books 2013, E-Book available via www.ub.tum.de

(2) [Rossant 2015] Learning IPython for Interactive Computing and Data Visualization (SECOND EDITION) by Cyrille Rossant, 175 pages Packt Publishing, October 2015

(3) Nathan Eagle and Alex (Sandy) Pentland. 2006. Reality mining: sensing complex social systems. Personal Ubiquitous Comput. 10, 4 (March 2006)