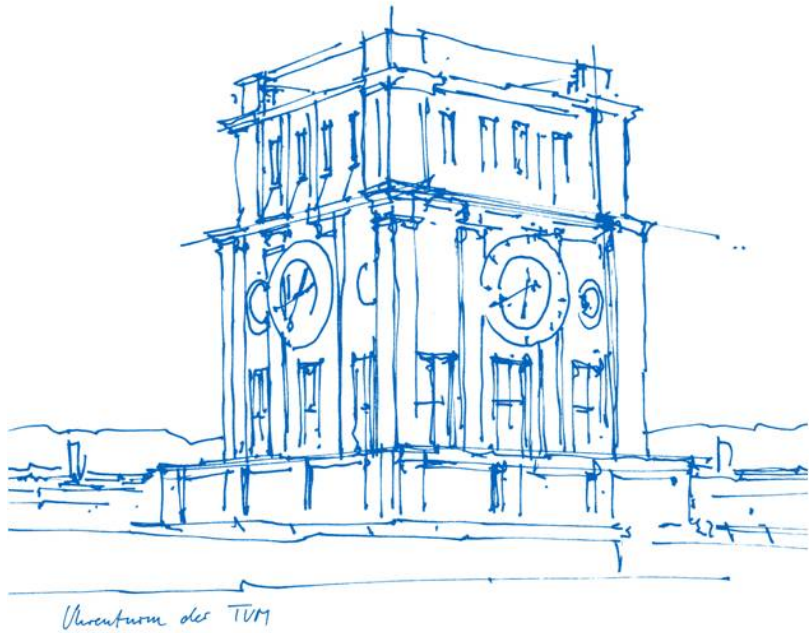
Exercises for Social Gaming and Social Computing (IN2241 + IN0040)

Exercise Sheet 4

Topic: Clustering



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

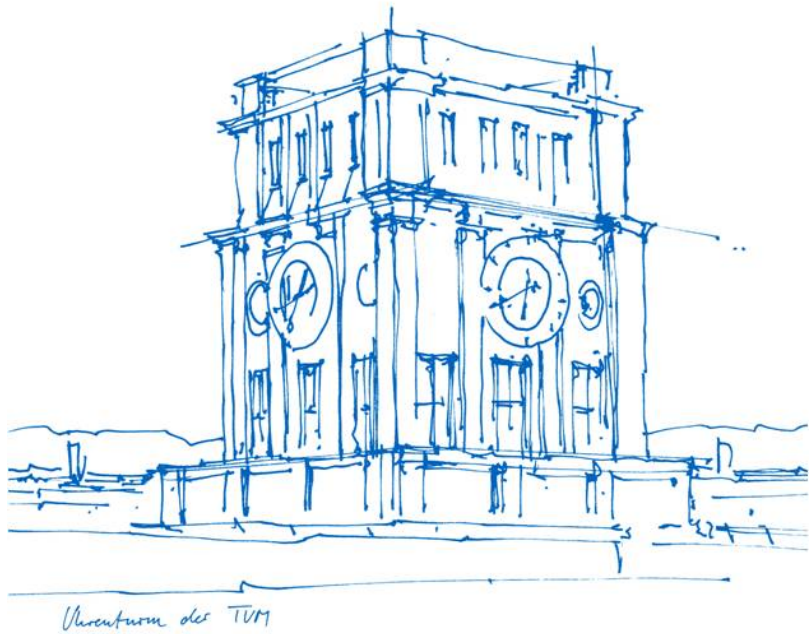
4	• Clustering:
	  • metric: K-means Clustering
	  • networks: Girvan-Newman-Algorithm
		• mobile data gathering (nothing actively to do)
		• social context in group rec experiment: complete questionnaires, form groups, rate restaurants etc.
			• **Sunday, June 17, 24:00**

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.

Problem 4.1

Problem 4.1. K-means Clustering of social network data

Write a Python program that computes K-means clustering for a given social network dataset. The input dataset (file: networkinput.csv) contains anyonymized data from user profiles of a small social networking platform. Each line in the file represents one feature vector and is associated with a social network user. The vectors have a name ("userXYZ") and the following 4 features:

- Number of posts
- Number of comments
- Number of likes (on both posts and comments)
- Number of friends and followers (average)

By clustering the users, you identify users with similar activity patterns. This can be helpful for research but also for advertisers and polling firms.

- use K=4
- you may (optionally) compare your results to library functions from scikit-learn (see remark on sheet)

K-Means Clustering

- General idea (also valid in graph clustering): Optimize objective function that formalizes clustering paradigm.

- K-Means: Optimize intra cluster coherence:

  - Describe cluster C_k by prototype μ_k; prototype need not be an actual pattern (If so, algorithm works with slight modifications as well)

  - Determine cluster for each pattern x_n by nearest neighbour rule:

$$\mathscr{C}(x_n) = k_a \leftrightarrow ||x_n - \mu_{k_a}|| = \min_k ||x_n - \mu_k||$$

K-Means Clustering

- K-Means: Optimize intra cluster coherence:
  - Find prototypes by optimizing objective function modeling intra cluster coherence as mean square error

$$J_{\mathrm{SQE}} = \sum_{k=1}^{K} \sum_{\{n|x_n \in \mathcal{C}_k\}} ||x_n - \mu_k||^2$$

$$\frac{\mathrm{d}J_{\mathrm{SQE}}}{\mathrm{d}\mu_k} \stackrel{!}{=} 0 \quad \Rightarrow \quad \mu^k = \frac{1}{|\mathcal{C}_k|} \sum_{\{n|x_n \in \mathcal{C}_k\}} x_n$$

  - → cluster prototypes are barycenters („centers of gravity“) of their clusters.

K-Means Clustering

- K-Means: Iterative Algorithm: Alternating optimization:

Iterate {Compute set of Prototypes; Assign cluster membership} until convergence criterion is met.

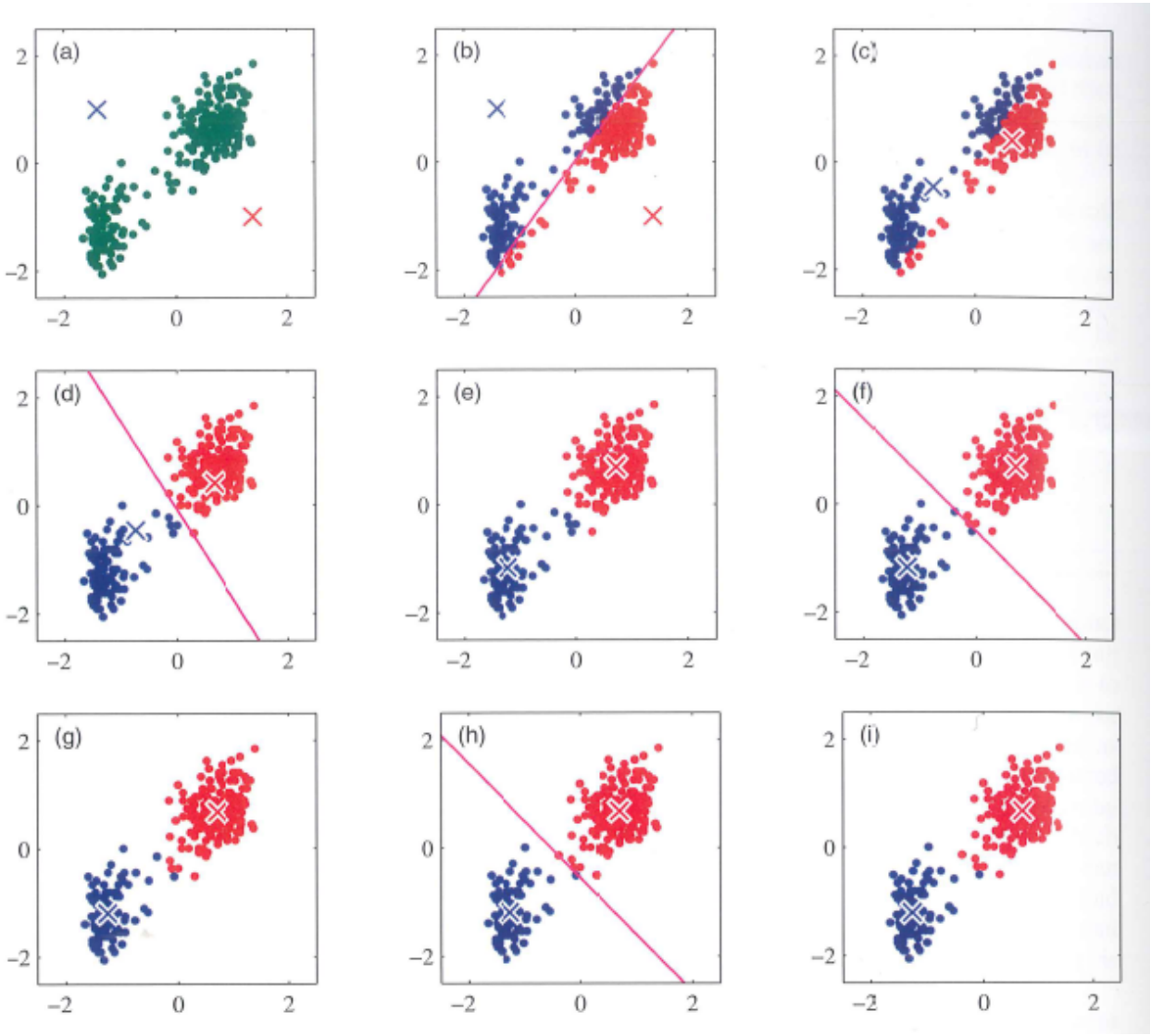
K-Means Clustering

- Input:
  - Number of clusters K.
  - Set of patterns $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$
  - Maximum number of iterations $t_{\max}$
  - stopping criterion bound ϵ
  - Suitable metric $||\,||$
- Initialize:
  - Determine randomly the initial set of prototypes ${}^{(0)} = \{\mu_1{}^{(0)}, \mu_2{}^{(0)}, \ldots, \mu_K{}^{(0)}\}$
- Process:
  - For (step $t := 0$; $t \leq t_{\max}$; t++)
    - Compute the association of the patterns to the clusters with the nearest neighbor rule $\mathscr{C}(x_n) = k_a \leftrightarrow ||x_n - \mu_{k_a}|| = \min_i ||x_n - \mu_k||$
    - Compute the prototypes ${}^{(t)}$ with $\mu^k = \frac{1}{|\mathcal{C}_k|} \sum_{\{n|x_n \in \mathcal{C}_k\}} x_n$
    - If $(\forall i \; ||\mu_i{}^{(t)} - \mu_i{}^{(t-1)}|| < \epsilon)$ break.

  Endfor

[7],[1]

K-Means Clustering



[3]

Optional Problem 4.2: Girvan Newman Method

- Implement Girvan Newman method
- Test it on the Krackhardt Kite graph

Newman Girvan Method: Centrality-based Splitting + Modularity

Last example of this part: bringing it all together (see [3]):

- Observations → critique on agglomerative methods: fail to cluster peripheral nodes correctly [3] → Newman Girvan method: Divisive hierarchical clustering (splitting) + Modularity:

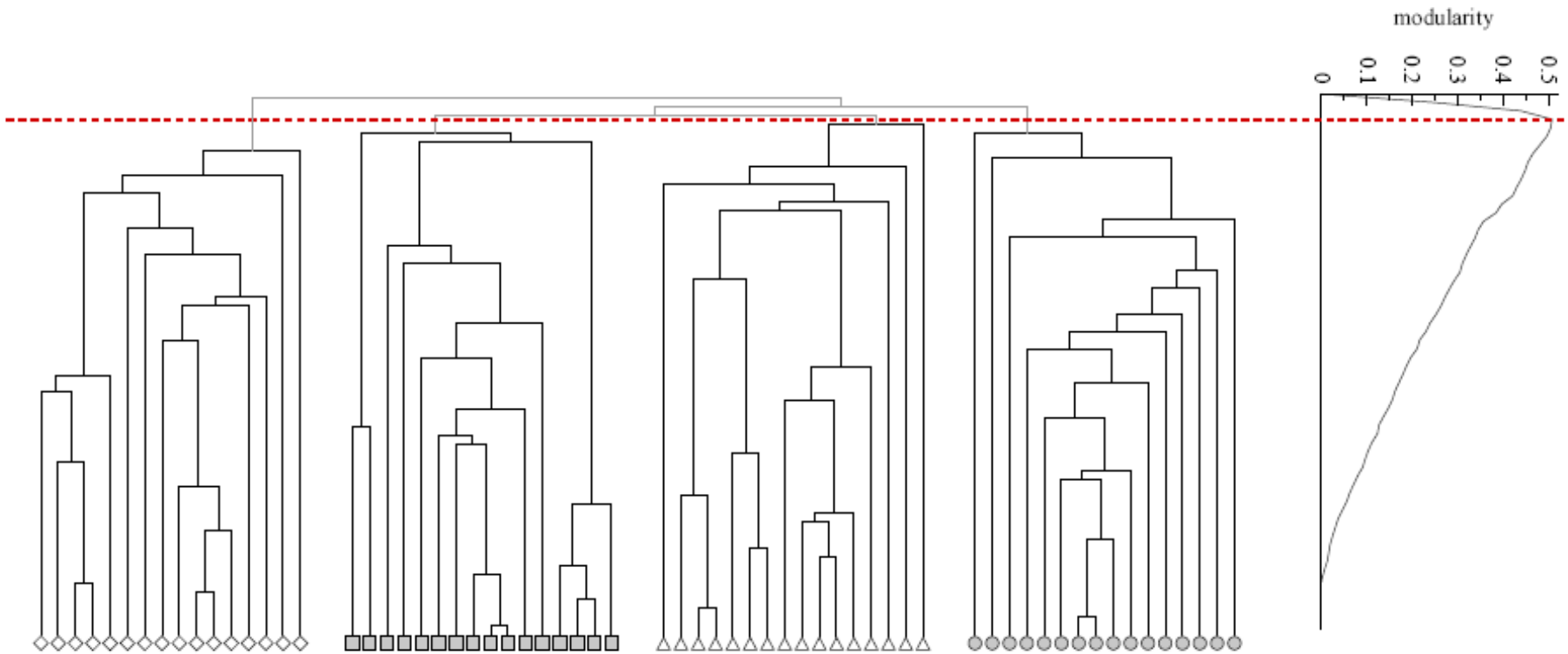
1. Calculate edge betweenness for all edges
2. Remove edge with highest edge betweenness
3. goto 1.

→ dendrogram

- Use Modularity as intra cluster coherence (f) cluster validity measure (g=0) to optimally cut dendrogram:

$$Q = \sum_i (e_{ii} - a_i^2) = \mathrm{Tr}\,\mathbf{e} - \| \mathbf{e}^2 \|$$

Newman Girvan Method: Centrality-based Splitting + Modularity



[3]

Newman Girvan Method: Centrality-based Splitting + Modularity

Which edge centrality?

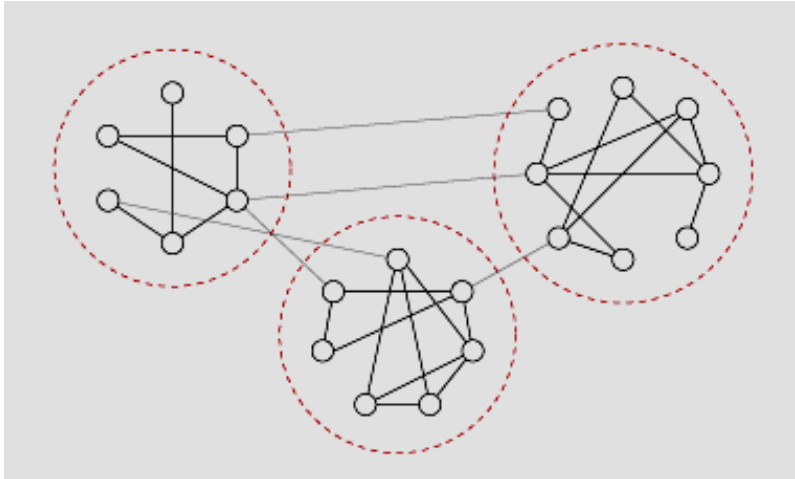
- Shortest Path Betweeness (works best for most cases [3])

  (naive: $O(n^2m)$ (breadth first ($O(m)$) for each pair of vertices) →
  better: $O(nm)$ Alg. by Brandes or Newman [3])

- Electric Network based == Random Walk based (see [3])

Newman Girvan Method: Centrality-based Splitting + Modularity

Modularity:



[3]

- k clusters → k x k symmetric matrix **e**: e_{ij} = |E(C_i,C_j)| / |E| : fraction of edges between communities

- $\mathrm{Tr}\,\mathbf{e} = \sum_i e_{ii}$: fraction of edges within communities

- $a_i = \sum_j e_{ij}$: fraction of edges that connect to cluster C_i

- Random network (keep a_i fixed): $e_{ij}^{\mathrm{rnd}} = a_i a_j \rightarrow e_{ii}^{\mathrm{rnd}} = a_i^2$

- f: Compare (→difference) real with rnd → $Q = \sum_i (e_{ii} - a_i^2) = \mathrm{Tr}\,\mathbf{e} - \| \mathbf{e}^2 \|$

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 17, 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)

(4) Desrosiers, C., & Karypis: A Comprehensive Survey of Neighborhood-based Recommendation Methods, 2011 in: Ricci et al (eds.) "Recommender Systems Handbook", Springer 2011

(5) U. Brandes, A faster algorithm for betweenness centrality. Journal of Mathematical Sociology 25, 163–177 (2001)

(6) M. E. J. Newman and M. Girvan: Finding and evaluating community structure in networks. Arxiv, 2003