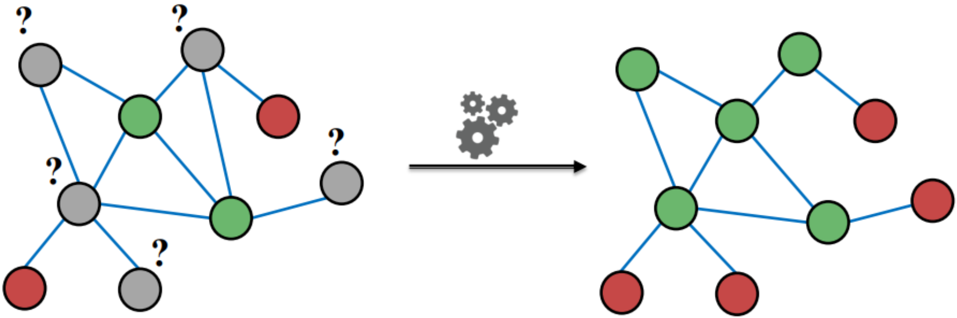


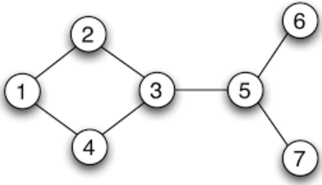
Label Propagation Algorithm for Detecting Communities in Directed Acyclic Networks

Suzana Antunović

5th Croatian Combinatorial Days, Zagreb

Graph vs. Network

"Network is a graph with meaning!"

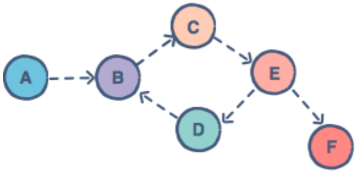


(a) Graph

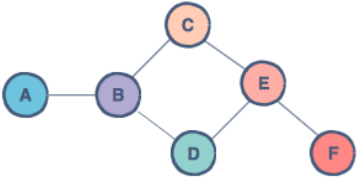
(b) Network

Directed vs. Undirected

Directed

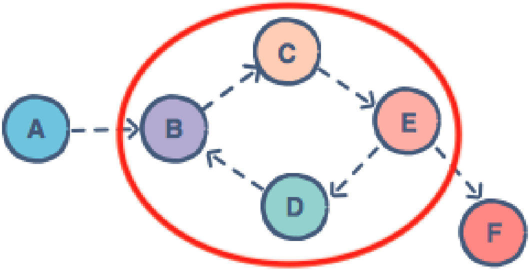


Undirected



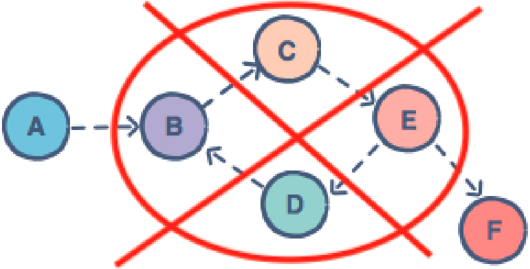
Directed acyclic network

***D**irected **A**cyclic **N**etwork*



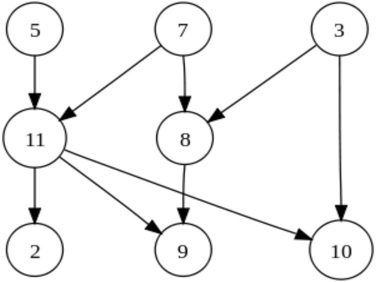
Directed acyclic network

***D**irected **A**cyclic **N**etwork*



Topological ordering

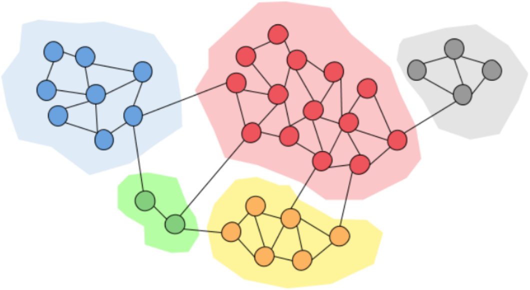
"Every DAN has a topological ordering."



- 5, 7, 3, 11, 8, 2, 9, 10
- 3, 5, 7, 8, 11, 2, 9, 10
- 5, 7, 3, 8, 11, 10, 9, 2
- 7, 5, 11, 3, 10, 8, 9, 2
- 5, 7, 11, 2, 3, 8, 9, 10
- 3, 7, 8, 5, 11, 10, 2, 9

Community detection

"The problem of ***community detection*** *relates to finding a natural division of the network into groups of vertices such that there are many edges within the community, and several edges between communities ."*



Modularity

*"**Modularity** measures the actual ratio of edges within the community reduced by the expected value in the null–model, where the division into communities is the same, but the edges between the vertices are placed randomly. "*

$$Q_d = \frac{1}{m} \sum_{1 \leq i,j \leq n} \left[A_{ij} - \frac{d^{in}(j) d^{out}(i)}{m} \right] \delta(l_i, l_j)$$

Mathematical framework

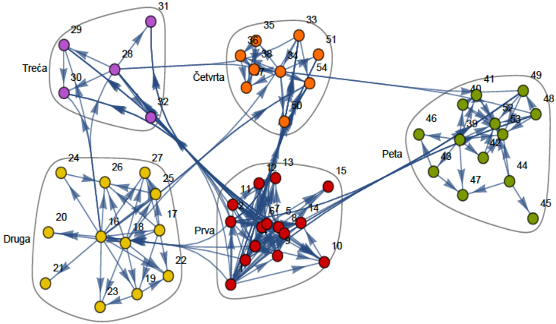
- G = directed acyclic network with n vertices and m directed edges
- it holds $x_1 \prec x_2 \prec \ldots \prec x_n$ (topological order)

We are interested in finding communities $C_1, C_2, \ldots, C_k$ such that

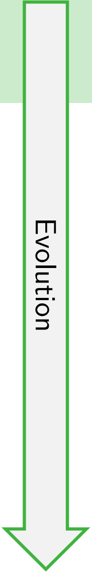
if $x_i \prec x_j$, $x_i \in C_p$ and $x_j \in C_q$ then $C_p \prec C_q$ or $C_p = C_q$.

Challenges

- formulation of the term "community"
- edge direction
- topological order of communities

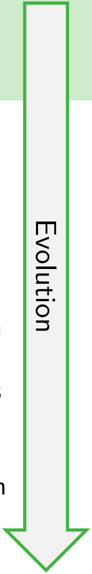


Label propagation algorithms



- **LPA**
  - every node is initialized with a unique community label
  - these labels propagate through the network
  - at every iteration of propagation, each node updates its label to the one that the maximum numbers of its neighbours belongs to. Ties are broken arbitrarily but deterministically
  - LPA reaches convergence when each node has the majority label of its neighbours
  - LPA stops if either convergence, or the user-defined maximum number of iterations is achieved

Label propagation algorithms



- **LPAm**
  - modularity-specialized label propagation algorithm
  - modified label update rule: choosing a label that will result in maximal modularity increse
  - LPAm brings a monotonous increase in modularity and avoids the possibility of forming a trivial solution
  - LPAm has the same effective speed as LPA
  - however, the tendency is to get stuck at a low local maximum of modularity

Label propagation algorithms

- **LPAm+**
  - advanced modularity-specialized label propagation algorithm
  - to escape local maxima, algorithm employs a multistep greedy agglomerative algorithm (MSG) that can merge multiple pairs of communities at a time
  - LPAm+ successfully detects communities with higher modularity values
  - LPAm+ offers a fair compromise between accuracy and speed

Evolution

Orientation Respecting LPAm

Algorithm 1 Orientation Respecting LPAm (**OLPAm)**

Require: Edge list

Ensure: Community division, modularity

to each vertex i assign a unique numerical label $l_i(0) = p(i)$
set $t = 1$
repeat
  put vertices in random order X
  for each vertex $i \in X$ **do**
    among in–neighbors $x_{i_1}, x_{i_2}, ..., , x_{i_k}$ of vertex i with labels $l_{i_1}, l_{i_2}, ..., l_{i_k}$ find the largest label l_{max}
    among out–neighbors $x_{i_{k+1}}, x_{i_{k+2}}, ..., , x_{i_n}$ of vertex i with labels $l_{i_{k+1}}, l_{i_{k+2}}, ..., l_{i_n}$ find the smallest label l_{min}
    calculate $\Delta Q_d(i, max)$ and $\Delta Q_d(i, min)$
    if $\Delta Q_d(i, max) > \Delta Q_d(i, min)$ and $\Delta Q_d(i, max) > 0$ **then**
      set $l_i(t) = l_{max}$
    else if $\Delta Q_d(i, min) > \Delta Q_d(i, max)$ and $\Delta Q_d(i, min) > 0$ **then**
      set $l_i(t) = l_{min}$
    else if $\Delta Q_d(i, min) = \Delta Q_d(i, max) > 0$ **then**
      uniformly at random pick l_{max} or l_{min} and set it for l_i
    end if
    set $t = t + 1$
  end for
  if neither of vertices $i \in X$ changes its label **then**
    end algorithm
  else
    set $t = t + 1$
  end if
until neither vertex in the iteration changes its label

Orientation Respecting LPAm+

Algorithm 2 Orientation Respecting LPAm+ (**OLPAm+**)

assign to each vertex a unique numeric label
using OLPAm algorithm maximize modularity Q_d
while there are communities A_i and A_j such that $\Delta Q_d(l_i l_j) > 0$ **do**
  for each community A_i **do**
    calculate $\Delta Q_d(l_i l_{max})$ and $\Delta Q_d(l_i l_{min})$
  end for
  find the maximal value of all $\Delta Q_d(l_i l_j) > 0$
  merge communities A_i and A_j such that $\Delta Q_d(l_i l_j) > 0$ is maximal
  maximize modularity Q_d using OLPAm algorithm
end while

Algorithm 3 Modified OLPAm+ with multiple merging of communities

assign to each vertex a unique numeric label
using OLPAm algorithm maximize modularity Q_d
while $\exists$ pair of communities (A_i, A_j) such that $\Delta Q(l_i, l_j) > 0$ **do**
  for each pair of connected communities (A_i, A_j) where $\Delta Q(l_i, l_j) > 0$ **do**
    if there is no community A labeled l such that $\Delta Q(l, l_i) > \Delta Q(l_i, l_j)$ and $\Delta Q(l, l_j) > \Delta Q(l_i, l_j)$ **then**
      merge communities A_i and A_j
    end if
  end for
  maximize modularity Q_d using OLPAm algorithm
end while

Data sets

Table: **Basic statistics for curriculum networks.**

Network	n	m	d_{in}	d_{out}	d_{avg}	**l**	**C**
Number set $\mathbb{Q}$	47	254	17	26	5.404	2.011	0.254
Elementary functions	84	502	27	51	5.976	2.132	0.255
Integral	223	655	15	28	2.941	3.899	0.084
Physics	31	49	4	8	1.581	1.575	0.049
Primary production	28	93	9	14	3.321	2.135	0.183
Data processing	54	197	12	22	3.648	1.744	0.338

Results

Table: **Comparison of the results obtained using the *OLPAm+* with the results suggested by the experts who compiled the curriculum networks.**

			Expert		*OLPAm+*	
	n	m	Q_d	N_c	Q_d	N_c
Number set $\mathbb{Q}$	47	254	0.311	5	0.377	4
Elementary functions	84	502	0.239	6	0.354	4
Integral	223	655	0.455	10	0.468	7
Data processing	54	197	0.389	6	0.426	5
Primary production	28	93	0.237	3	0.293	3
Physics	31	49	0.238	6	0.476	6

Results

Table: **Comparison of the results obtained using the $OLPAm+^{(m)}$ with the results suggested by the experts who compiled the curriculum networks.**

			Expert		$OLPAm+^{(m)}$	
	n	m	Q_d	N_c	Q_d	N_c
Number set $\mathbb{Q}$	47	254	0.311	5	0.377	4
Elementary functions	84	502	0.239	6	0.337	5
Integral	223	655	0.455	10	0.470	10
Data processing	54	197	0.389	6	0.426	5
Primary production	28	93	0.237	3	0.283	3
Physics	31	49	0.238	6	0.467	5

Results

Table: **Comparison of the results obtained using different algorithms for community detection under constrints.**

	Expert		*RA*		*OLPAm+*		$OLPAm+^{(m)}$	
	Q_d	N_c	Q_d	N_c	Q_d	N_c	Q_d	N_c
$\mathbb{Q}$ set	0.311	5	0.377	4	0.377	4	0.377	4
El. functions	0.239	6	0.286	8	0.354	4	0.337	5
Integral	0.455	10	0.484	10	0.468	7	0.470	10
Data proc.	0.389	6	0.430	6	0.426	5	0.426	5
Pr. prod.	0.237	3	0.259	3	0.293	3	0.293	3
Physics	0.238	6	0.375	4	0.476	6	0.467	5

Thank you for your attention!

"Mathematics reveals its secrets only to those who approach it with pure love, for its own beauty."

– Archimedes