Network Clustering

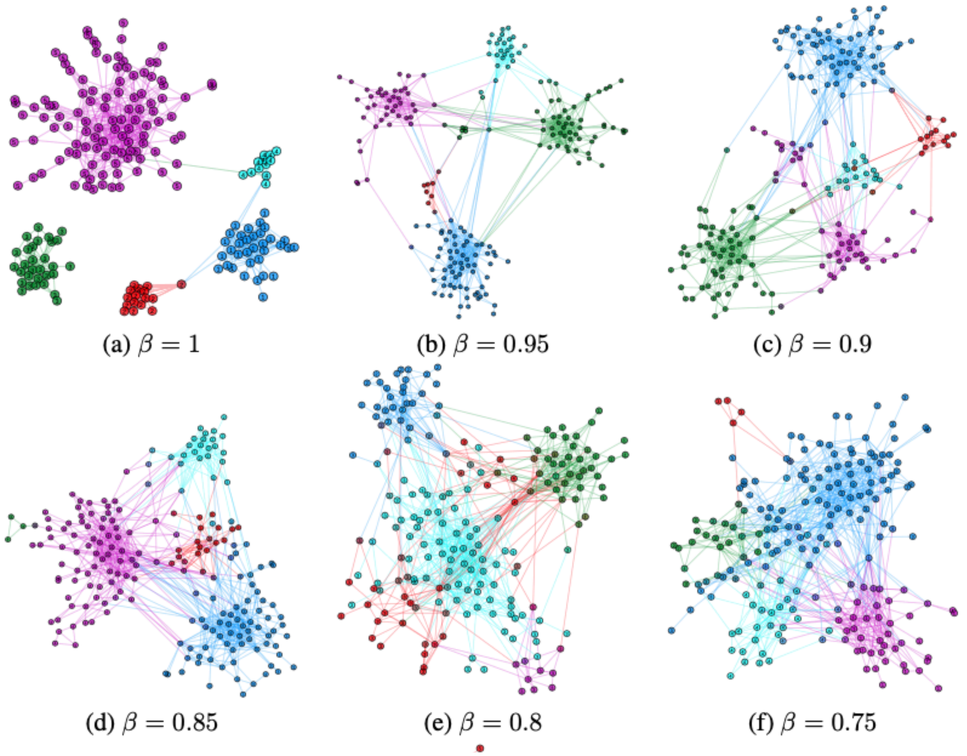
Part 1

Alexander Ponomarenko

What is cluster? What is community?

“A community is a subset of actors among whom there are relatively strong, direct, intense, frequent or positive ties”

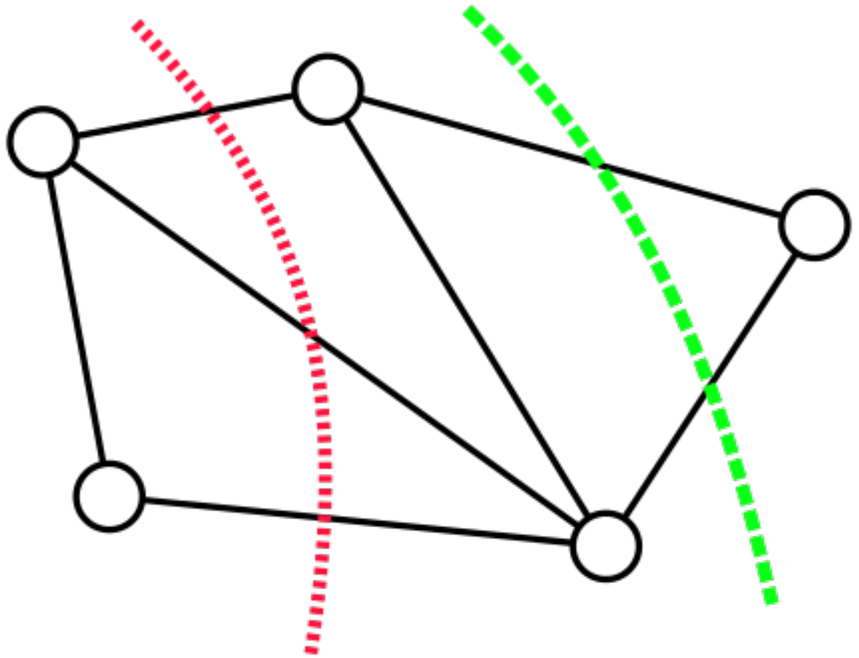
— Wasserman and Faust, Social Network Analysis, Method and Applications



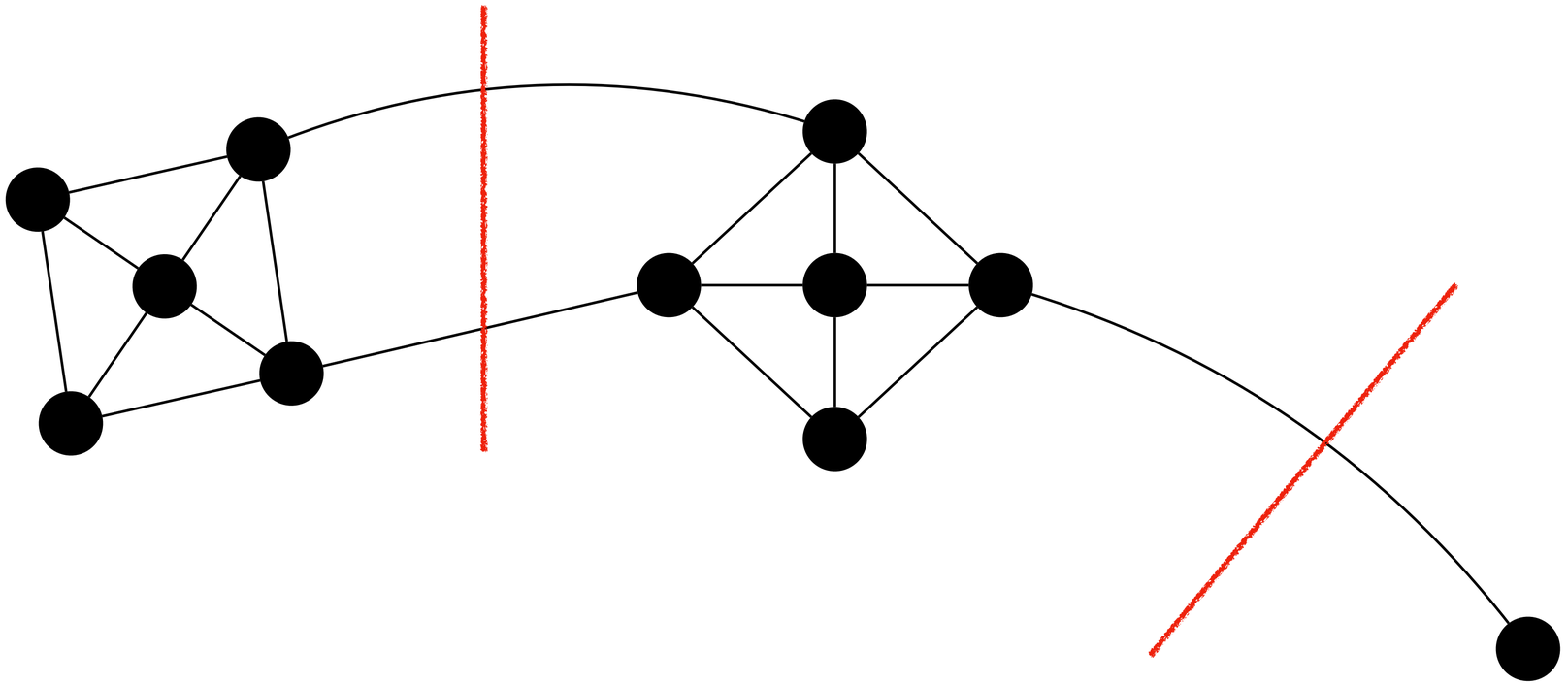
- Label propagation
- Utility function optimisation
  - Modularity
- Direct Optimisation Methods
- Distance Based
- Spectral clustering
- Greedy Clique Extension
- Min-cut
- Graph neural networks

Minimum Cut

- Can be solved in polynomial time by the Stoer-Wagner algorithm.
- A generalization of the minimum cut problem without terminals is the minimum *k*-cut
- For a fixed value of *k*, this problem can be solved in polynomial time, though the algorithm is not practical for large k



$$\min_{C_1,\ldots,C_k} \sum_{s=1}^{k} links(C_s, V \backslash C_s)$$



Ratio Cut

$$\min_{C_1,\ldots,C_k} \sum_{s=1}^{k} \frac{links(C_s, V \backslash C_s)}{|C_s|}$$

[Wei Y. C., Cheng C. K. Ratio cut partitioning for hierarchical designs //IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 1991. – Т. 10. – №. 7. – С. 911-921.]

Normalized Cut

$$\min_{C_1,\ldots,C_k} \sum_{s=1}^{k} \frac{links(C_s, V \backslash C_s)}{degree(C_s)}$$

[Shi J., Malik J. Normalized cuts and image segmentation //IEEE Transactions on pattern analysis and machine intelligence. – 2000. – Т. 22. – №. 8. – С. 888-905.]

Min-Max Cut

$$\min_{C_1,\ldots,C_k} \sum_{s=1}^{k} \frac{links(C_s, V \backslash C_s)}{links(C_s, C_s)}$$

[Ding C. H. Q. et al. A min-max cut algorithm for graph partitioning and data clustering //Proceedings 2001 IEEE international conference on data mining. – IEEE, 2001. – C. 107-114.]

Minimum cut:

$$\min_{C_1,\ldots,C_k} \sum_{s=1}^{k} links(C_s, V \backslash C_s)$$

Ratio cut (Cheng and Wei, 1991):

$$\min_{C_1,\ldots,C_k} \sum_{s=1}^{k} \frac{links(C_s, V \backslash C_s)}{|C_s|}$$

Normalized cut (Shi and Malik, 2000):

$$\min_{C_1,\ldots,C_k} \sum_{s=1}^{k} \frac{links(C_s, V \backslash C_s)}{degree(C_s)}$$

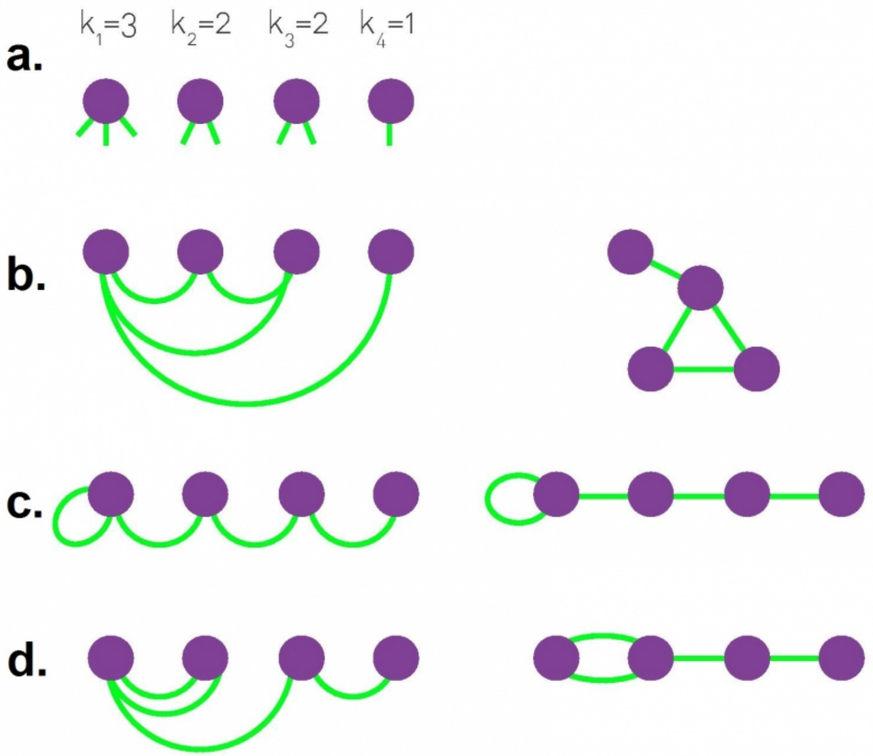
Min-max cut (Ding et al., 2001):

$$\min_{C_1,\ldots,C_k} \sum_{s=1}^{k} \frac{links(C_s, V \backslash C_s)}{links(C_s, C_s)}$$

Modularity & Configuration Model

Configuration Model

1. Take a degree sequence, i. e. assign a degree k_i to each vertex. The degrees of the vertices are represented as half-links or stubs. The sum of stubs must be even in order to be able to construct a graph ($\sum k_i = 2m$). The degree sequence can be drawn from a theoretical distribution or it can represent a real network (determined from the adjacency matrix of the network).
2. Choose two stubs uniformly at random and connect them to form an edge. Choose another pair from the remaining $2m - 2$ stubs and connect them. Continue until you run out of stubs. The result is a network with the pre-defined degree sequence. The realization of the network changes with the order in which the stubs are chosen, they might include cycles (b), self-loops (c) or multi-links (d) (Figure 1). Yet, the expected number of self-loops and multi-links goes to zero in the $N \to \infty$ limit.[1]



n — number of nodes, *m* — *number of edges*

Modularity *Q* is defined as the fraction of edges that fall within group 1 or 2, minus the expected number of edges within groups 1 and 2 for a random graph with the same node degree distribution as the given network

Modularity

ожидаемое значение той же величины в графе, в котором вершины имеют одинаковую степень, но ребра расположены случайным образом.

$$Q = \sum_s [a_s - e_s]$$

доля всех ребер, лежащих внутри кластера *s*

нормализация

вероятность ребра между *i* и *j*

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) \delta(C^{(i)} C^{(j)})$$

A – это матрица смежности графа $G(V, E)$

$m = |E|$ – количество рёбер в графе

k_i – степени вершины i

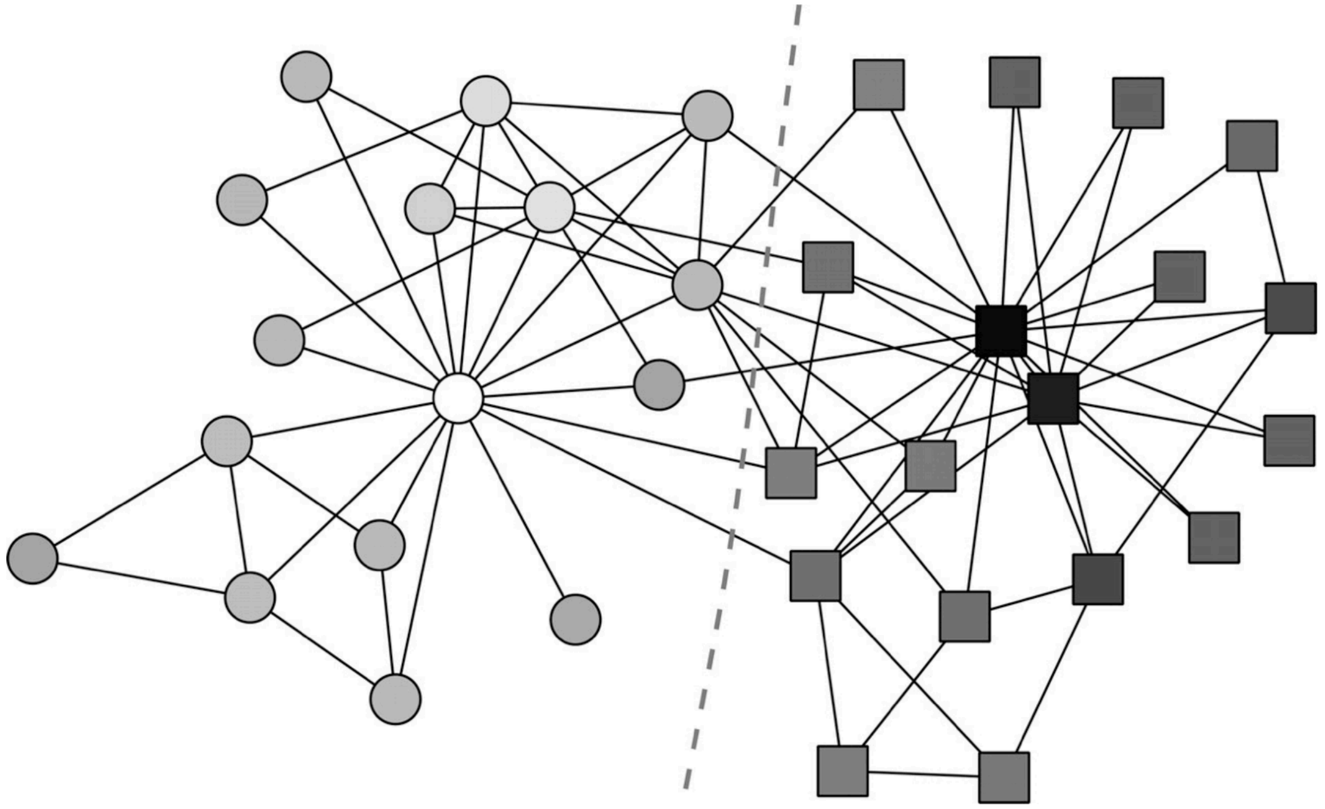
$C^{(i)}$ – номер кластера, к которому принадлежит вершина i

$\delta(x, y)$ – Кронекер дельта функция

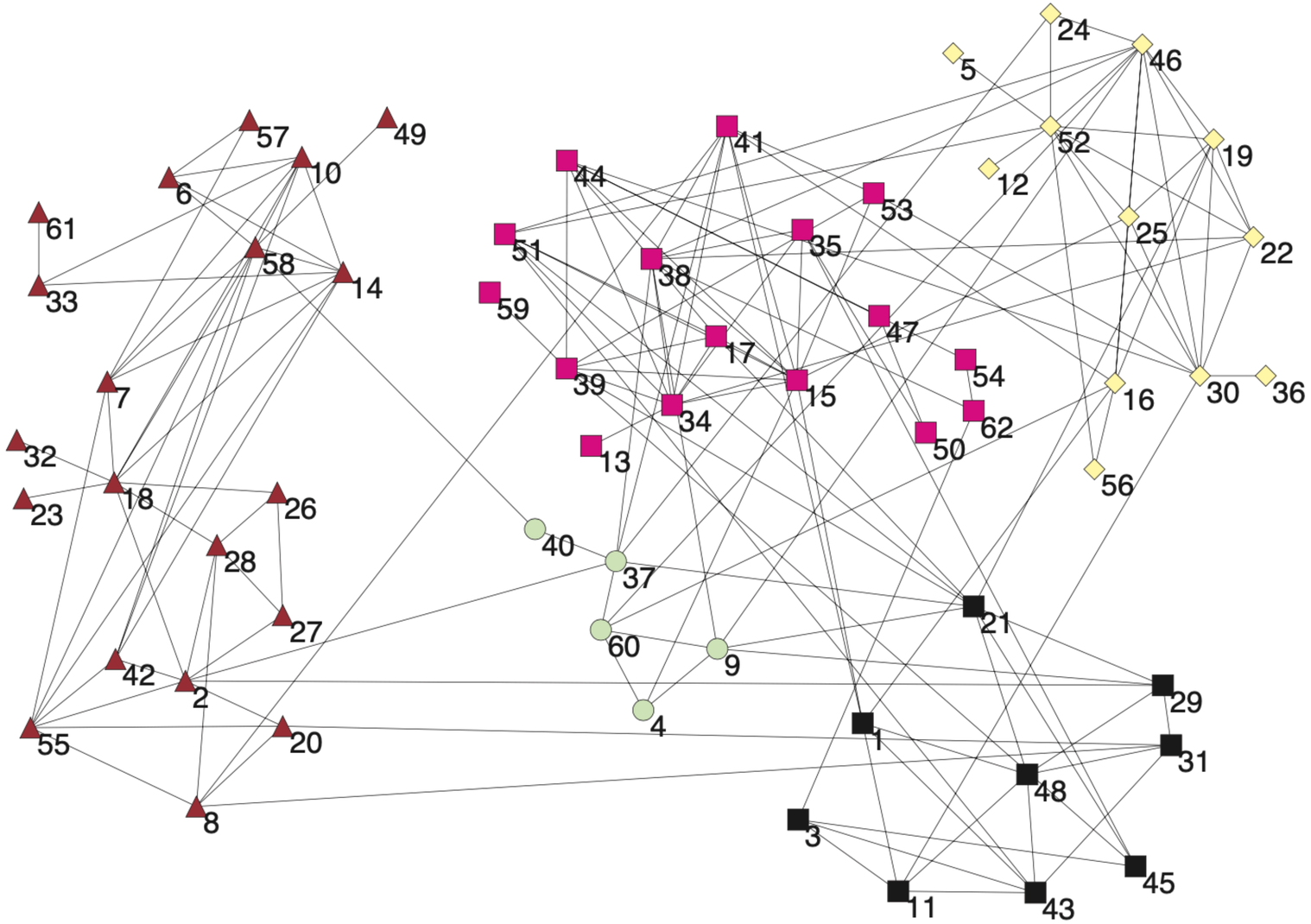
$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

[Newman M. E. J., Girvan M. Finding and evaluating community structure in networks //Physical review E. – 2004. – Т. 69. – №. 2. – С. 026113]

Karate Club Results: Exactly Right



Dolphins network



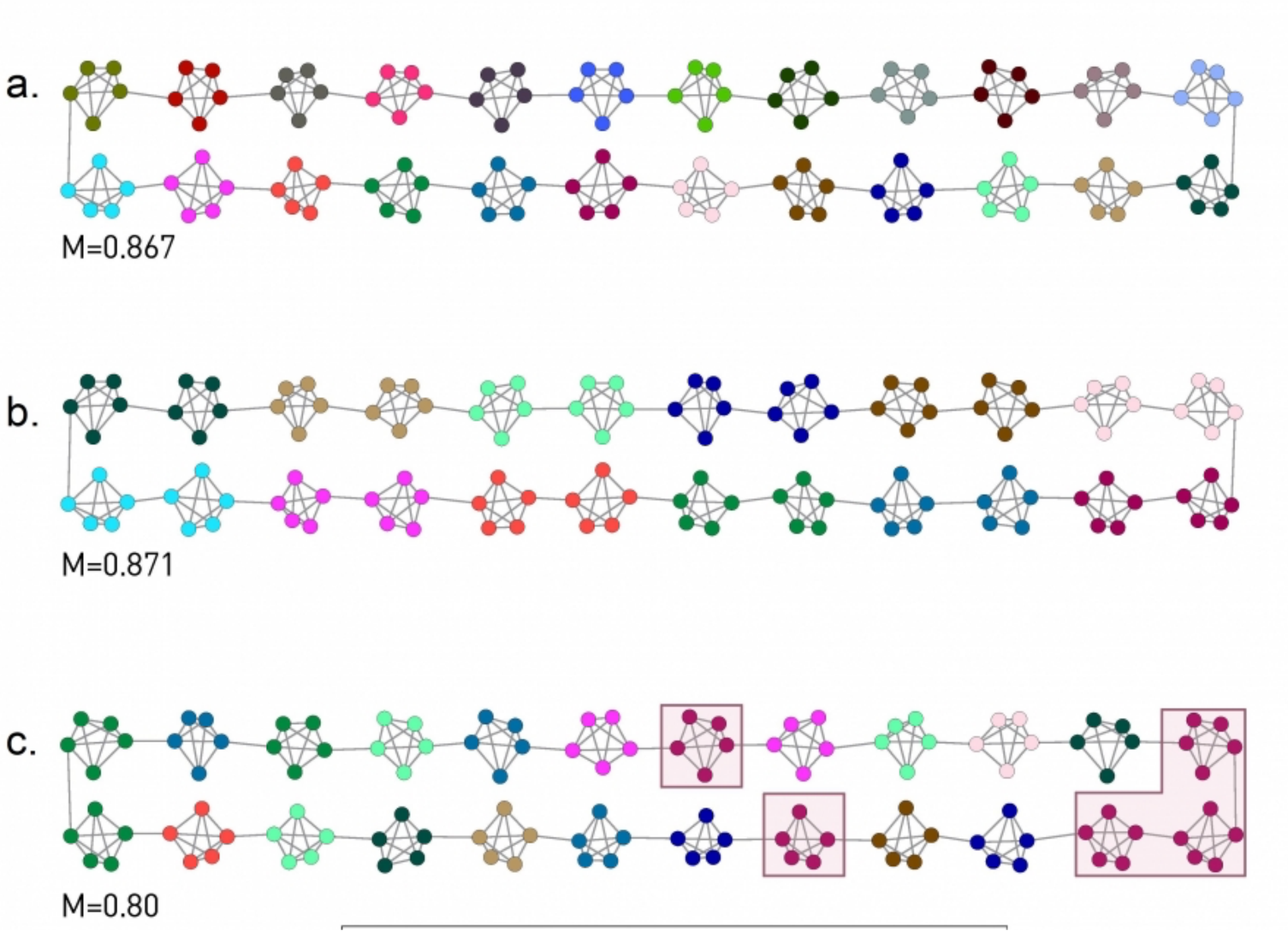
дельфины-афалины, которых изучал Люссо в Даутфул-Саунд, Новая Зеландия.
Сеть с 62 вершинами, соответствующими дельфинам, и 159 ребрами, соединяющими вершины, связанные с парами дельфинов, с частым общением между ними.

List of modularity maximisation based methods

- Louvain Algorithm https://arxiv.org/abs/0803.0476
- https://arxiv.org/abs/0803.0476
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5003342/
- Newman M. E. J. Fast algorithm for detecting community structure in networks //Physical review E. – 2004. – Т. 69. – №. 6. – С. 066133.
- Clauset A., Newman M. E. J., Moore C. Finding community structure in very large networks //Physical review E. – 2004. – Т. 70. – №. 6. – С. 066111.
- Wakita K., Tsurumi T. Finding community structure in mega-scale social networks //Proceedings of the 16th international conference on World Wide Web
C. 1275-1276.
- Guillaume L. Fast unfolding of communities in large networks //Journal Statistical Mechanics: Theory and Experiment. – 2008. – Т. 10. – С. P1008.

Is modularity a silver bullet?

Modularity make no sense — Example



Label Propagation

Label Propagation

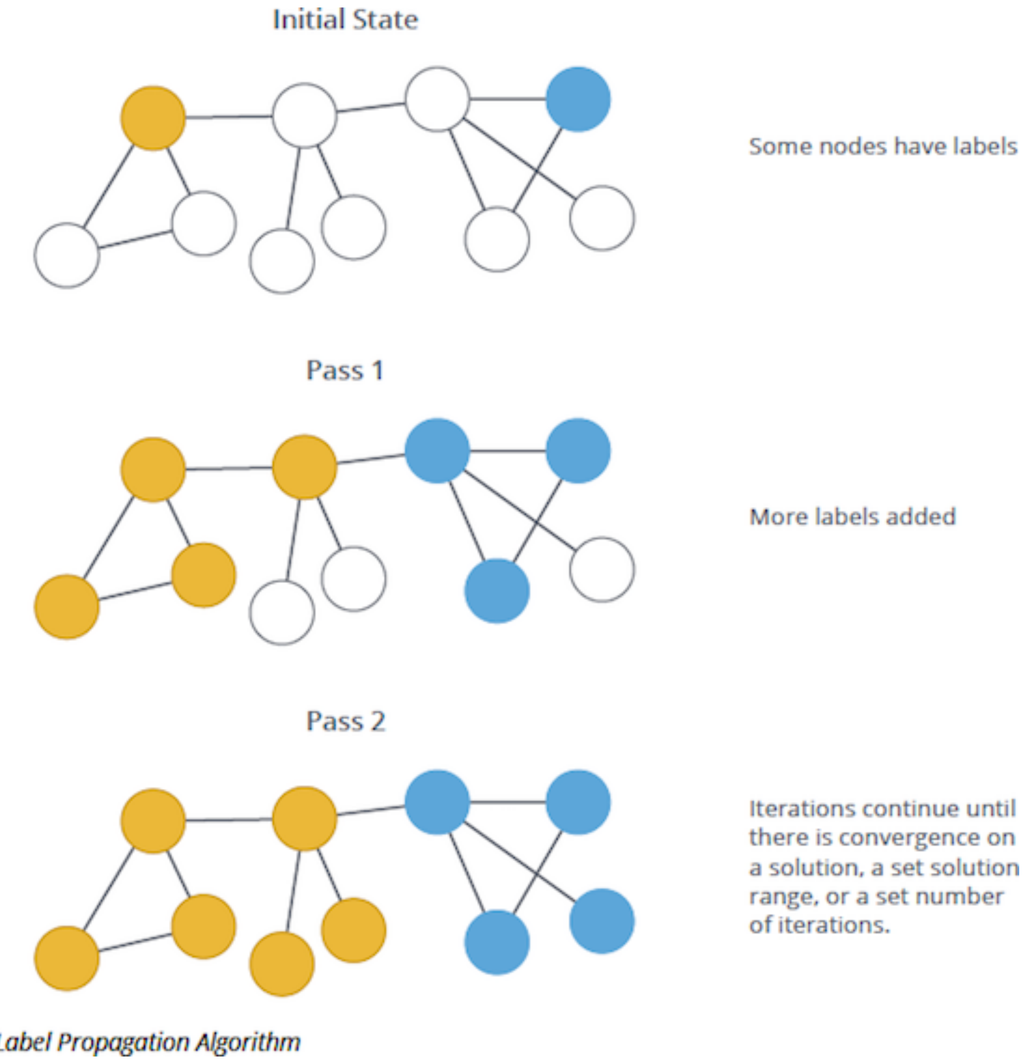
1. Initialize the labels at all nodes in the network. For a given node x, C_x (0) = x.

2. Set t = 1.

3. Arrange the nodes in the network in a random order and set it to X.

4. For each x ∈ X chosen in that specific order, let $C_x(t) = f(C_{x_{i1}}(t), ...,C_{x_{im}}(t),C_{x_{i(m+1)}}(t-1), ...,C_{x_{ik}}(t-1))$. Here returns the label occurring with the highest frequency among neighbours. Select a label at random if there are multiple highest frequency labels.

5. If every node has a label that the maximum number of their neighbours have, then stop the algorithm. Else, set t = t + 1 and go to (3).



Label Propagation Algorithm

Mixed Integer Programming (MIP) Approach

Example of the method based on MIP

Minimize $\quad \mathrm{D}_{max} + k_{max}^{out}$

subject to

$$D_{max} \geq d_{ij}(x_{ic} + x_{jc} - 1) \qquad \forall i, j, c \;\; i < j \tag{1}$$

$$\sum_{c=1}^{c_0} x_{ic} = 1 \qquad \forall i \tag{2}$$

$$\sum_{i=1}^{n} x_{ic} \geq 1 \qquad \forall c \tag{3}$$

$$\sum_{j=1}^{n} A_{ij} x_{jc} \geq x_{ic}\left(\frac{\sum_{j=1}^{n} A_{ij}}{2}\right) \qquad \forall i, c \tag{4}$$

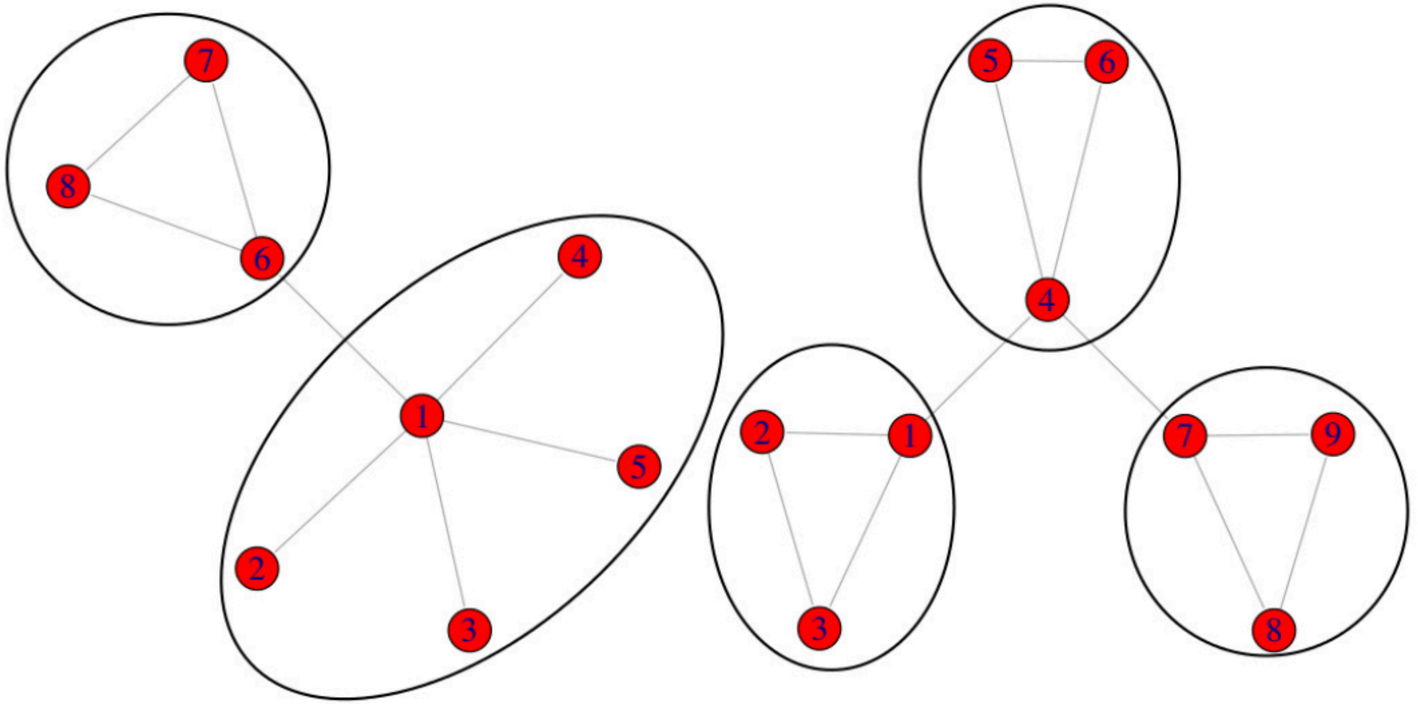
$$\sum_{j=1}^{n} A_{ij} x_{jc} \geq x_{ic} \sum_{j=1}^{n} A_{ij} - k_{max}^{out} \quad \forall i, c \tag{5}$$

$$x_{ic} \in \{0, 1\} \qquad \forall i, c \tag{6}$$

$$D_{max}, k_{max}^{out} \geq 0 \tag{7}$$

Model parameters are the number of objects (n), the number of clusters
), the shortest path distances between objects i and j (d_{ij}), and the ad-
ency of objects i and j (A_{ij}). A_{ij} is 1 if objects i and j are connected, 0
herwise. The decision variables are x_{ic}, D_{max}, and k_{max}^{out} which are defines
follows: x_{ic} is 1 if object i is assigned to cluster c, 0 otherwise; D_{max} is
length of the longest diameter among all cluster diameters; k_{max}^{out} is the

connection number of the object which has the maximum number of con-
tions with the objects outside its cluster. Constraint set (1) ensures that
$_{ax}$ is the maximum diameter. Constraint set (2) ensures that each object
assigned to exactly one cluster. Constraint set (3) ensures that a cluster
at least one object. Constraint set (4) ensures that an object has at least
many connections with objects inside its cluster as the number of con-
tions with objects outside its cluster. Constraint set (5) establishes the
ation with objective function. Constraint sets (6) and (7) ensure that x_{ic}
binary and D_{max} and k_{max}^{out} are non-negative. The model has nc_0 binary
iables, *two* continuous variables, and $\frac{c_0 n^2}{2} + \frac{3c_0 n}{2} + n + k$ constraints.


7
8
6
4
1
5
2
3
5
6
4
2
1
3
7
9
8

Продолжение следует...

Будь здоров