Fuzzy clustering and the concept of bridgeness in social networks

Tamás Nepusz[1,2,3] Andrea Petróczi[3] Fülöp Bazsó[2]

[1]Budapest University of Technology and Economics
Dept. of Measurement and Information Systems

[2]Research Institute for Particle and Nuclear Physics
of the Hungarian Academy of Sciences
Dept. of Biophysics,
Computational Neuroscience Group

[3]Kingston University, School of Life Sciences

Intl. Workshop and Conference on Network Science 2007

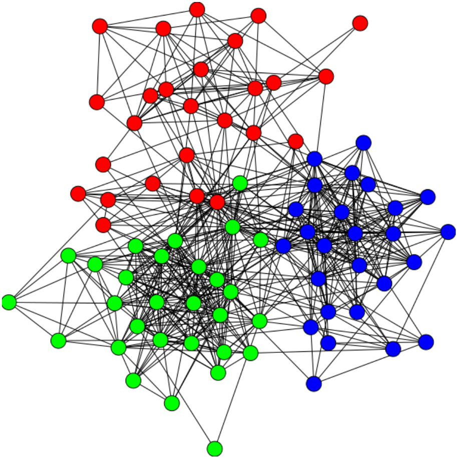
Outline

Why bother with fuzzy graph clustering?

- Overlapping community structure in complex networks

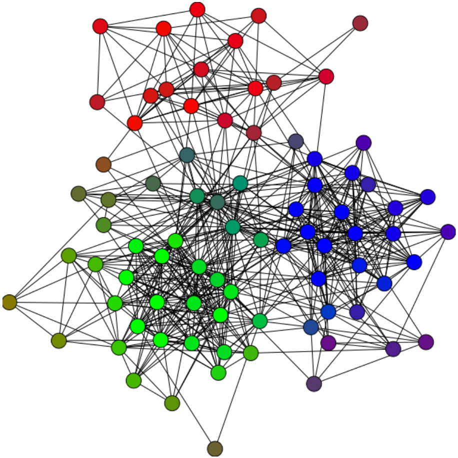
  G. Palla, I. Derényi, I. Farkas and T. Vicsek: *Uncovering the overlapping community structure of complex networks in nature and society*. Nature, 435(7043) 814–818, 2005.
- Vertices belonging to multiple clusters are particularly interesting:
  - Social networks: “social bridges”
  - Brain areas: high level information processing
  - . . .
- Fuzzy sets provide a great tool for studying these overlapping structures.

Graph clustering – traditional approach



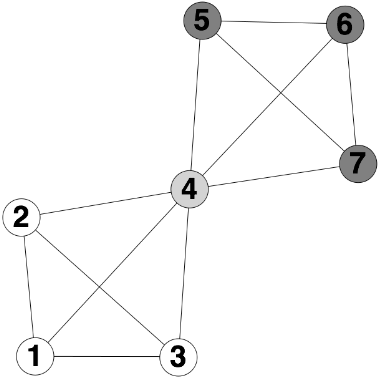
- There are a fixed number of clusters in the end
- The clusters are classical ("crisp") sets: a vertex is either included in a cluster or not
- Evey vertex belongs to exactly one of the clusters

Graph clustering – fuzzy approach



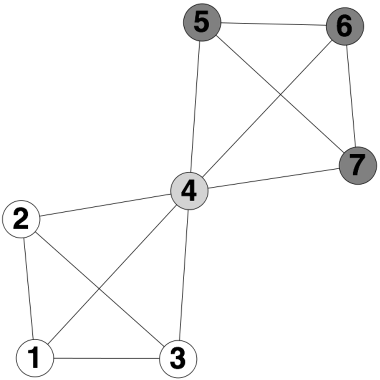
- There are a fixed number of clusters
- The clusters are fuzzy sets: a vertex is included in a cluster with a given grade of membership
  - 0 = not included
  - 1 = fully included
  - Between 0 and 1 = partially included
- The grades of memberships for all vertices add up to 1.

Cluster profile



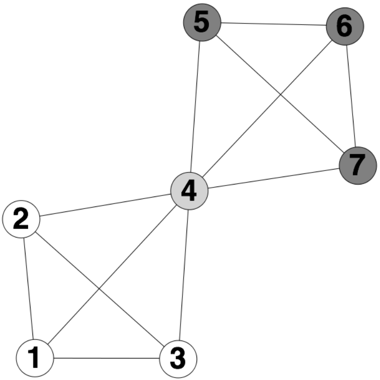
- The cluster profile of a vertex is the grade of membership of the vertex in each cluster, formulated as a vector
- Vertex 1: $\mathbf{c}_1^T = [1, 0]$
- Vertex 7: $\mathbf{c}_7^T = [0, 1]$
- Vertex 4: $\mathbf{c}_4^T = [0.5, 0.5]$

Cluster profile



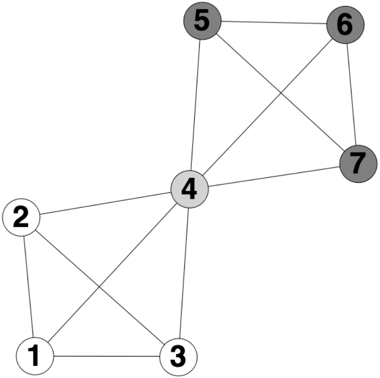
- The cluster profile of a vertex is the grade of membership of the vertex in each cluster, formulated as a vector
- Vertex 1: $\mathbf{c}_1^T = [1, 0]$
- Vertex 7: $\mathbf{c}_7^T = [0, 1]$
- Vertex 4: $\mathbf{c}_4^T = [0.5, 0.5]$

Cluster profile



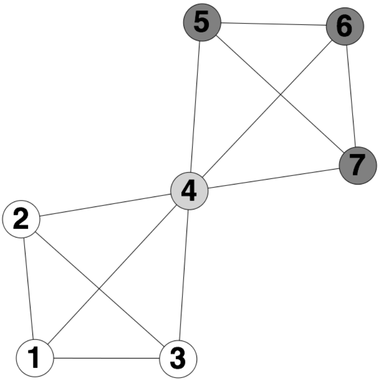
- The cluster profile of a vertex is the grade of membership of the vertex in each cluster, formulated as a vector
- Vertex 1: $\mathbf{c}_1^T = [1, 0]$
- Vertex 7: $\mathbf{c}_7^T = [0, 1]$
- Vertex 4: $\mathbf{c}_4^T = [0.5, 0.5]$

Cluster profile



- The cluster profile of a vertex is the grade of membership of the vertex in each cluster, formulated as a vector
- Vertex 1: $\mathbf{c}_1^T = [1, 0]$
- Vertex 7: $\mathbf{c}_7^T = [0, 1]$
- Vertex 4: $\mathbf{c}_4^T = [0.5, 0.5]$

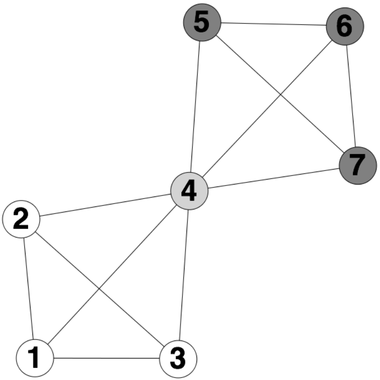
Cluster profile matrix



- The cluster profiles of all vertices can be written in a matrix form:

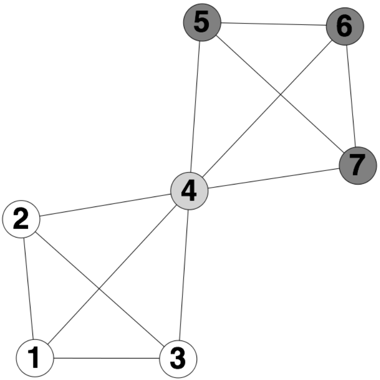
$$\mathbf{C} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0.5 & 0.5 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Similarity



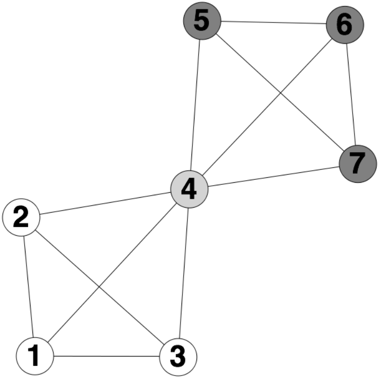
- The similarity of two vertices is the dot product of their cluster profiles
- $s_{1,2} = 1 \times 1 + 0 \times 0 = 1$
- $s_{1,7} = 1 \times 0 + 0 \times 1 = 0$
- $s_{1,4} = 1 \times 0.5 + 0 \times 0.5 = 0.5$

Similarity



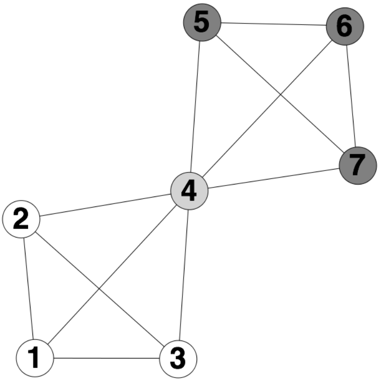
- The similarity of two vertices is the dot product of their cluster profiles
- $s_{1,2} = 1 \times 1 + 0 \times 0 = 1$
- $s_{1,7} = 1 \times 0 + 0 \times 1 = 0$
- $s_{1,4} = 1 \times 0.5 + 0 \times 0.5 = 0.5$

Similarity



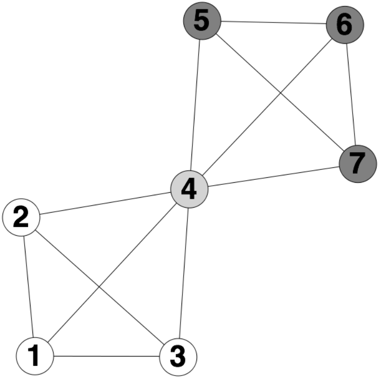
- The similarity of two vertices is the dot product of their cluster profiles
- $s_{1,2} = 1 \times 1 + 0 \times 0 = 1$
- $s_{1,7} = 1 \times 0 + 0 \times 1 = 0$
- $s_{1,4} = 1 \times 0.5 + 0 \times 0.5 = 0.5$

Similarity



- The similarity of two vertices is the dot product of their cluster profiles
- $s_{1,2} = 1 \times 1 + 0 \times 0 = 1$
- $s_{1,7} = 1 \times 0 + 0 \times 1 = 0$
- $s_{1,4} = 1 \times 0.5 + 0 \times 0.5 = 0.5$

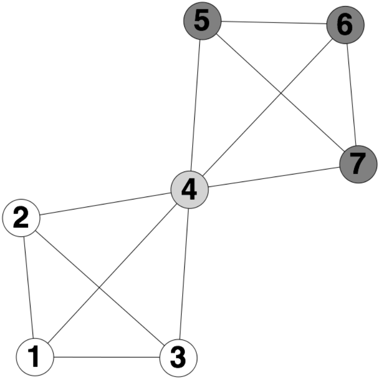
Similarity matrix



- Pairwise similarities are calculated in the similarity matrix $\mathbf{S}$
- Note that $\mathbf{S} = \mathbf{C}^T\mathbf{C}$

$$\begin{bmatrix} 1 & 1 & 1 & 0.5 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0.5 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0.5 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0.5 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0.5 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0.5 & 1 & 1 & 1 \end{bmatrix}$$

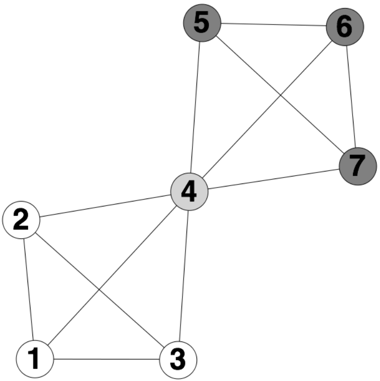
Bridgeness



- Bridgeness measures how “shared” a vertex is among the clusters
- $b_1 = b_2 = b_3 = 0$
- $b_5 = b_6 = b_7 = 0$
- $b_4 = 1$
- $\mathbf{b}^T = [0, 0, 0, 1, 0, 0, 0]$
- A possible way to calculate it:

$$b_i = \delta_i \left(1 - \frac{k}{k-1} \sum_{i=1}^{k} \left(c_{ik} - \frac{1}{k} \right)^2 \right)$$

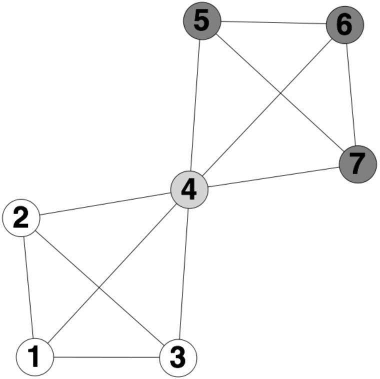
Bridgeness



- Bridgeness measures how "shared" a vertex is among the clusters
- $b_1 = b_2 = b_3 = 0$
- $b_5 = b_6 = b_7 = 0$
- $b_4 = 1$
- $\mathbf{b}^T = [0, 0, 0, 1, 0, 0, 0]$
- A possible way to calculate it:

$$b_i = \delta_i \left(1 - \frac{k}{k-1} \sum_{i=1}^{k} \left(c_{ik} - \frac{1}{k} \right)^2 \right)$$

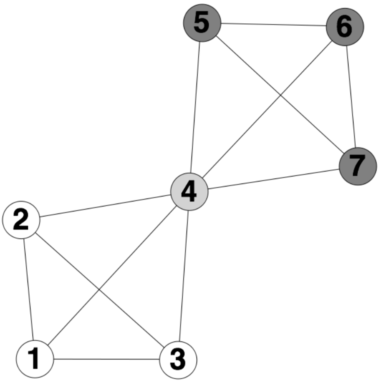
Bridgeness



- Bridgeness measures how “shared” a vertex is among the clusters
- $b_1 = b_2 = b_3 = 0$
- $b_5 = b_6 = b_7 = 0$
- $b_4 = 1$
- $\mathbf{b}^T = [0, 0, 0, 1, 0, 0, 0]$
- A possible way to calculate it:

$$b_i = \delta_i \left(1 - \frac{k}{k-1} \sum_{i=1}^{k} \left(c_{ik} - \frac{1}{k} \right)^2 \right)$$

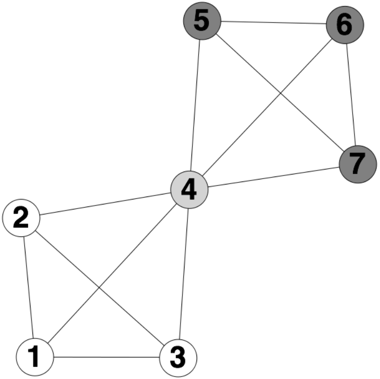
Bridgeness



- Bridgeness measures how “shared” a vertex is among the clusters
- $b_1 = b_2 = b_3 = 0$
- $b_5 = b_6 = b_7 = 0$
- $b_4 = 1$
- $\mathbf{b}^T = [0, 0, 0, 1, 0, 0, 0]$
- A possible way to calculate it:

$$b_i = \delta_i \left(1 - \frac{k}{k-1} \sum_{i=1}^{k} \left(c_{ik} - \frac{1}{k} \right)^2 \right)$$

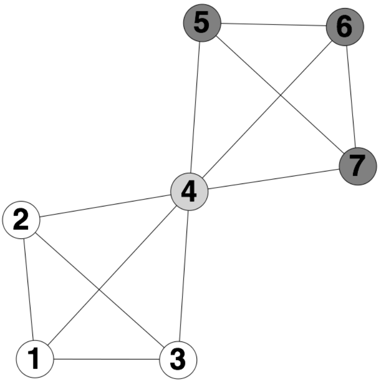
Bridgeness



- Bridgeness measures how "shared" a vertex is among the clusters
- $b_1 = b_2 = b_3 = 0$
- $b_5 = b_6 = b_7 = 0$
- $b_4 = 1$
- $\mathbf{b}^T = [0, 0, 0, 1, 0, 0, 0]$
- A possible way to calculate it:

$$b_i = \delta_i \left(1 - \frac{k}{k-1} \sum_{i=1}^{k} \left(c_{ik} - \frac{1}{k} \right)^2 \right)$$

Bridgeness



- Bridgeness measures how "shared" a vertex is among the clusters
- $b_1 = b_2 = b_3 = 0$
- $b_5 = b_6 = b_7 = 0$
- $b_4 = 1$
- $\mathbf{b}^T = [0, 0, 0, 1, 0, 0, 0]$
- A possible way to calculate it:

$$b_i = \delta_i \left(1 - \frac{k}{k-1} \sum_{i=1}^{k} \left(c_{ik} - \frac{1}{k} \right)^2 \right)$$

What is our goal with the clustering?

We want...

1. ...the endpoints of the edges to be similar ($s_{ij} \approx 1$) (with only this constraint, we would end up with all vertices having the same cluster profile)
2. ...pairs of vertices without an edge between them to be dissimilar ($s_{ij} \approx 0$)

What can we do to achieve it?

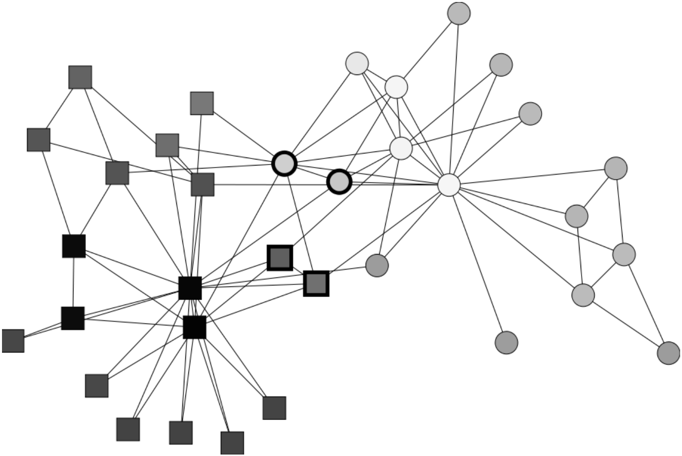
Fuzzy clustering as an optimization problem

- Define a goal function, e.g.:

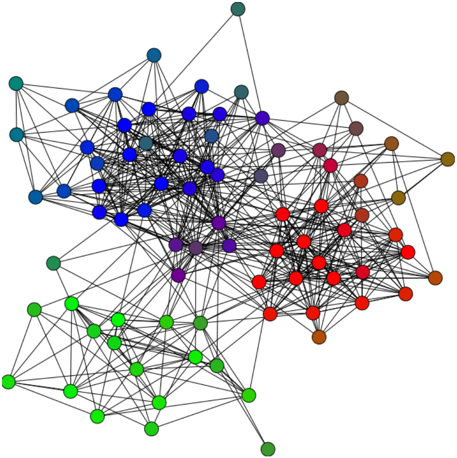
$$f(\mathbf{C}, G(V, E)) = \prod_{i \in V} \prod_{j \in V} \begin{cases} s_{ij} & \text{if } i \to j \in E \\ 1 - s_{ij} & \text{otherwise} \end{cases}$$

- Find **C** which maximizes the goal function while satisfying:
  1. $\mathbf{c}_i \geq [0, 0, \ldots, 0]$ for all i
  2. $\sum_{j=1}^{k} c_{jk} = 1$

Zachary karate club study



UK university dataset



- Personal ties of the academic staff of a Faculty of a UK university
- 3 schools
- 75 out of 81 vertices classified correctly
- Only 4 were misclassified
- No school affiliation information for the remaining two vertices

Conclusion

- A possible fuzzy extension of classical clustering was presented
- Identifies meaningful communities and bridges
- Advantage: more precise than graph partitioning
- Disadvantage: computationally complex in its present form (can be reduced)
- Possible extensions: directedness, edge weights

Contact information

- Tamás Nepusz
- E-mail: nepusz@mit.bme.hu
- Web: http://cneuro.rmki.kfki.hu/people/nepusz

How to maximize the goal function?

- By iterative methods (e.g. conjugate gradient)
- Using the Karush-Kuhn-Tucker conditions: if:
  1. all the inequality constraints are concave functions
  2. all the equality constraints are affine functions

  there exist a set of equations where the solution is the global maximum of the original goal function. This can be solved directly.

H. W. Kuhn and A. W. Tucker: *Nonlinear programming*. In: Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability, 481-492. University of California Press, 1951.

The Karush-Kuhn-Tucker conditions

- f: goal function
- $g_i \geq 0$: inequality constraints
- $h_j = 0$: equality constraints
  1. $\nabla f + \sum_{i=1}^{n} \mu_i \nabla g_i + \sum_{j=1}^{n} \nu_j \nabla h_j = 0$
  2. $\mu_i g_i \geq 0$ for all i
  3. $g_i \geq 0$ for all i
  4. $h_j = 0$ for all j

H. W. Kuhn and A. W. Tucker: *Nonlinear programming*. In: Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability, 481-492. University of California Press, 1951.