Network structure

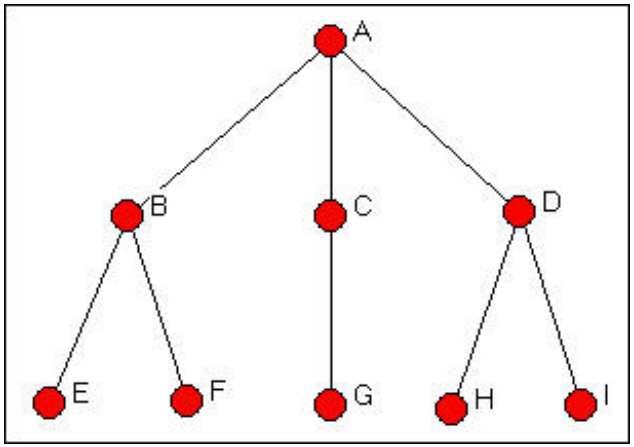
Leonid E. Zhukov

School of Applied Mathematics and Information Science
National Research University Higher School of Economics

27.11.2013

Patterns of relations

- Equivalence of positions and roles



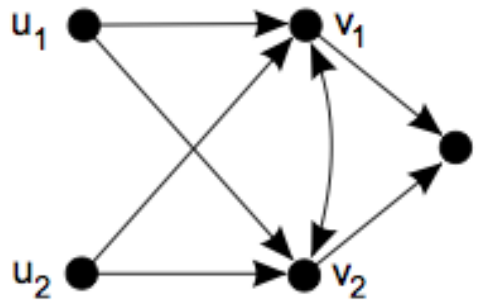
1. $\{A\}, \{B\}, \{C\}, \{D\}, \{E, F\}, \{H, I\}$
2. $\{A\}, \{C\}, \{G\}, \{B, D\}, \{E, F, H, I\}$
3. $\{A\}, \{E, F, G, H, I\}, \{B, C, D\}$

- Approximate equivalence, similarity between nodes

Structural equivalence

Definition

Structural equivalence: two vertices are structurally equivalent if their respective sets of in-neighbors and out-neighbors are the same



	u1	u2	v1	v2	w
u1	0	0	1	1	0
u2	0	0	1	1	0
v1	0	0	0	1	1
v2	0	0	1	0	1
w	0	0	0	0	0

rows and columns of adjacency matrix of structually equivalent nodes are identical, "connect to the same neighbors"

Approximate equvivalence

- Unweighted graph - binary matrix, only 0/1
- Eucledean distance between vectors

$$d(v_i, v_j) = \sqrt{\sum_k ((A_{ik} - A_{jk})^2 + (A_{ki} - A_{kj})^2)}$$

- Hamming distance - number of positions where vectors are different (Manhattan distance for binary matrix)

$$h(v_i, v_j) = \sum_k |A_{ik} - A_{jk}|$$

R: dist(data,method="euclidean')
R: dist(data,method="manhattan')

Similarity measures

- Cosine similarity (vectors in n-dim space)

$$\sigma(v_i, v_j) = cos(\theta_{ij}) = \frac{v_i v_j}{||v_i||||v_j||} = \frac{\sum_k A_{ik} A_{kj}}{\sqrt{\sum A_{ik} A_{ki}} \sqrt{\sum A_{jk} A_{kj}}} = \frac{n_{ij}}{\sqrt{k_i k_j}}$$

- Jaccard similarity

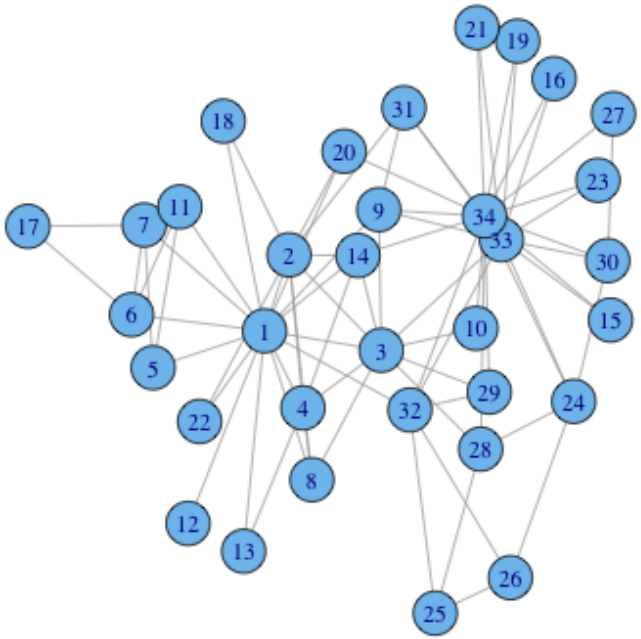
$$J(v_i, v_j) = \frac{|\mathcal{N}(v_i) \cap \mathcal{N}(v_j)|}{|\mathcal{N}(v_i) \cup \mathcal{N}(v_j)|}$$

- Pearson correlation coefficient:

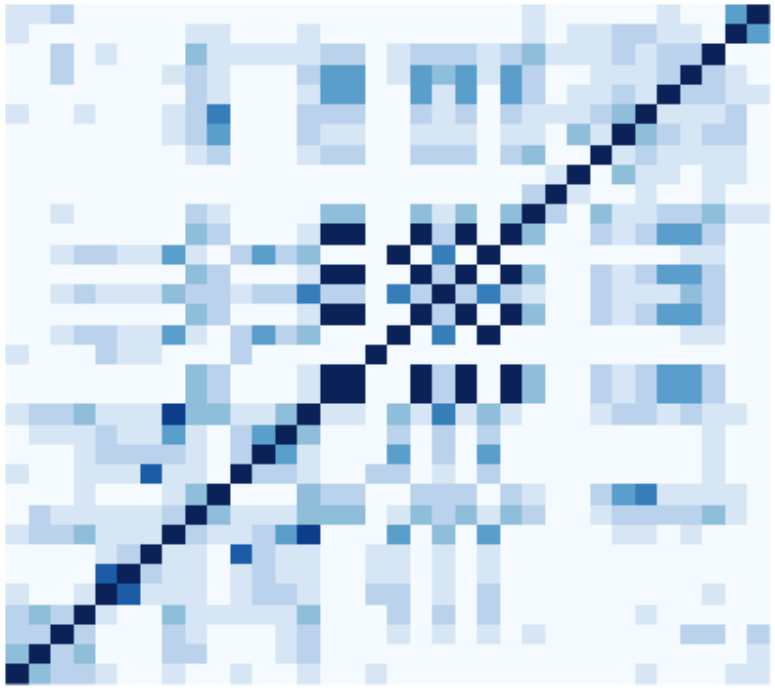
$$r(v_i, v_j) = \frac{\sum_k (A_{ik} - \langle A_i \rangle)(A_{jk} - \langle A_j \rangle)}{\sqrt{\sum_k (A_{ik} - \langle A_i \rangle)^2} \sqrt{\sum_k (A_{jk} - \langle A_j \rangle)^2}} = \frac{n_{ij} - \frac{k_i k_j}{n}}{\sqrt{k_i - \frac{k_i^2}{n}} \sqrt{k_j - \frac{k_j^2}{n}}}$$

R: similarity.jaccard {igraph}

Similarity matrix



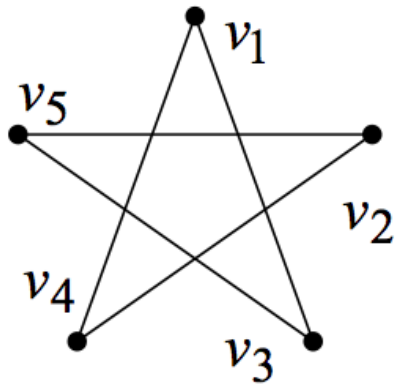
Graph



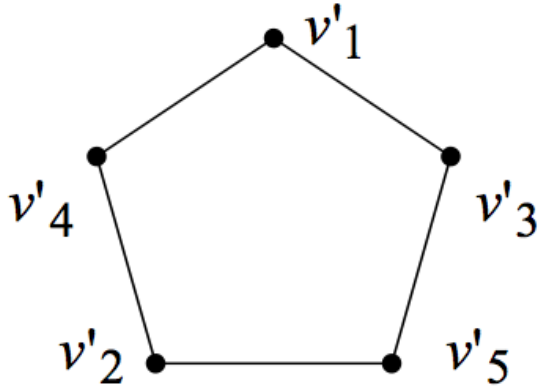
Similarity matrix

Graph isomorphism

Two graphs are isomorphic if there exists a one-to-one mapping of nodes, such that for every edge in one graph, there is a uniqe edge in another graph between the corresponding mapped vertices ("the same structure")



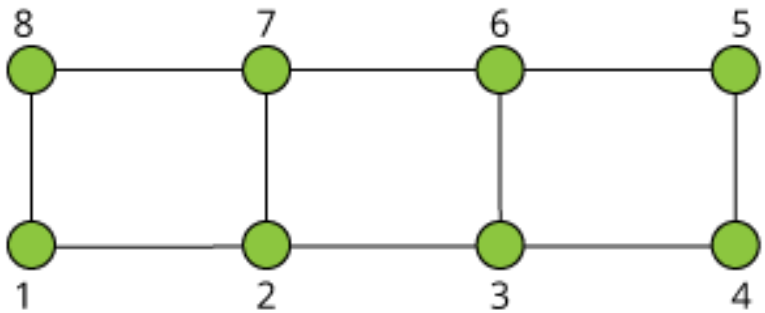
G



G'

Graph automorphism

Automorhism is a one-to-one mapping of nodes, such that for every edge in the graph, there is unique edge between the corresponding mapped vertices. This is a form of graph symmetry, isomorphism to istelf.

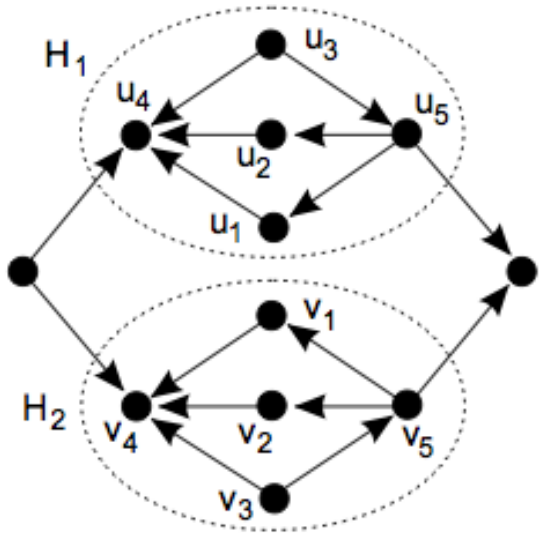


Automorphic equivalence

Definition

Two vertices are automorphically equivalent if there exist an automorphic mapping interchanging these nodes.

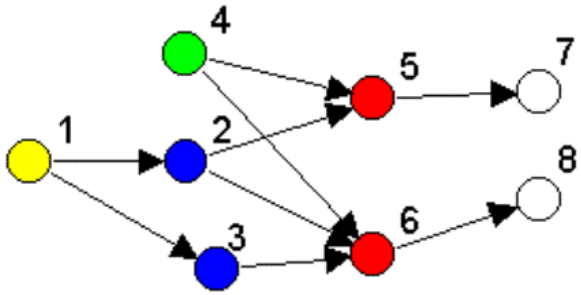
All vertcies relabeled forming isomorphic graph with two interchanged. All distance between nodes are preserved



Regular equivalence

Definition

Regular equivalence: Two vertices are regularly equivalent if they are equally related to equivalent others.



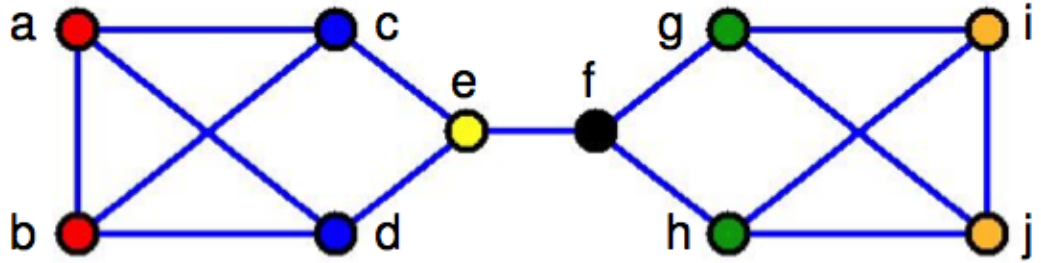
- when coloring, connected to the nodes of the same color
- σ_{ij} - similarity score

$$\sigma_{ij} = \alpha \sum_{k,l} A_{ik} A_{jl} \sigma_{kl}$$

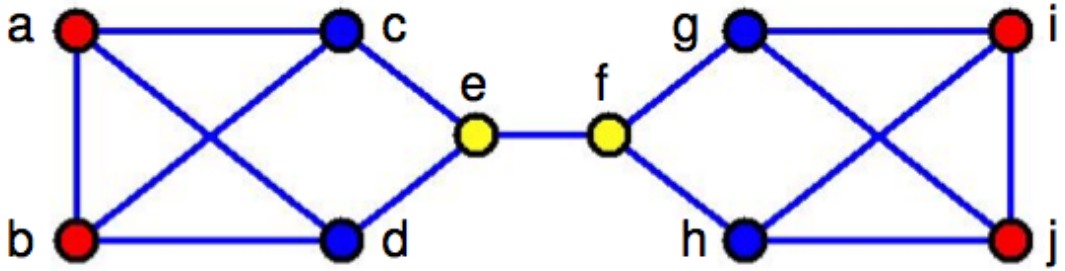
"connect to the same colors"

Equivalence

- structural equivalence



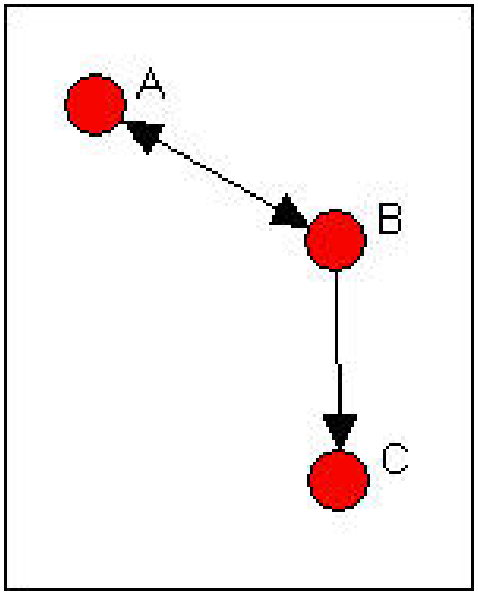
- regular equivalence



structural equivalence > automorphic equivalence > regular equivalence

Dyads

Dyad is a pair of verteces and possible relational ties between them: mutual, assymetric, null (non-existent)

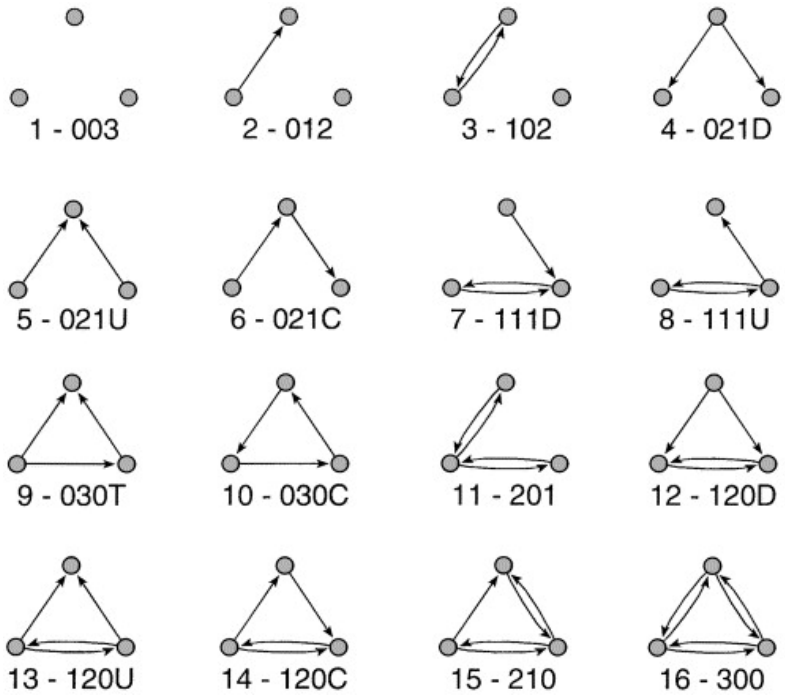


Dyad census computes each category (reciprocity)

R: dyad.census {igraph}

Triads

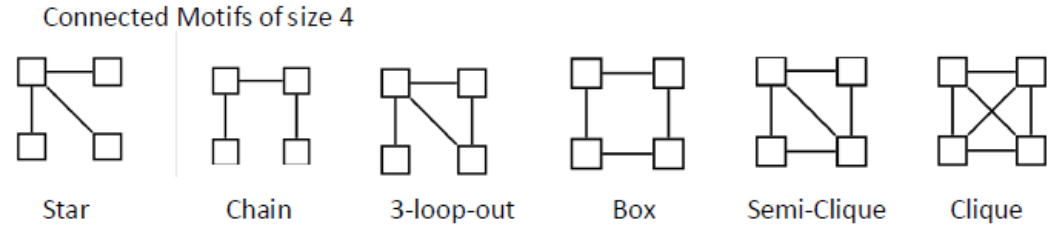
Triad is a subgraph of three vertices and possible ties between them

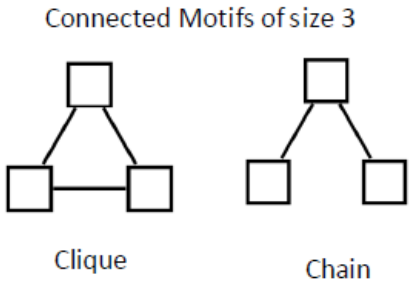


Triad census - classifying traids into 16 isomorphism classes
D - down, U - up, T - transitive, C - cyclic.
/mutual diads /assymetric dyads/ null dyads/
(transitivity)
R: triad.census {igraph}

Network motifs

Recurrent, statistically significant subgraphs or patterns in graphs (motifs are not induced subgraphs, i.e. they do not contatin all the graph edges between selected vertices)





$$Z_{score}(G) = \frac{F_M(G) - \mu_M(R)}{\sigma_M(R)}$$