

VERTEX SIMILARITY AND ITS APPLICATION TO FUNCTIONAL PREDICTION

Petter Holme

University of Michigan, Ann Arbor, U.S.A.

with **Elizabeth Leicht** and **Mark Newman** (University of Michigan) and **Mikael Huss** (Royal Institute of Technology, Stockholm, Sweden)

<http://www-personal.umich.edu/~pholme/>



Vertex equivalence / similarity

- In complex networks the nodes have different functions. These functions are reflected in their position in the networks.



Vertex equivalence / similarity

- In complex networks the nodes have different functions. These functions are reflected in their position in the networks.
- Can we, from the network structure, guess if two vertices have similar function?



Vertex equivalence / similarity

- In complex networks the nodes have different functions. These functions are reflected in their position in the networks.
- Can we, from the network structure, guess if two vertices have similar function?
- How can we use this information to classify the vertices / predict their functions?



Precepts of similarity measures

a vertex is similar to itself

two vertices are similar if their neighborhoods are similar



Precepts of similarity measures

a vertex is similar to itself

two vertices are similar if their neighborhoods are similar



Precepts of similarity measures

a vertex is similar to itself

two vertices are similar if their neighborhoods are similar

structural equivalence
/
structural similarity

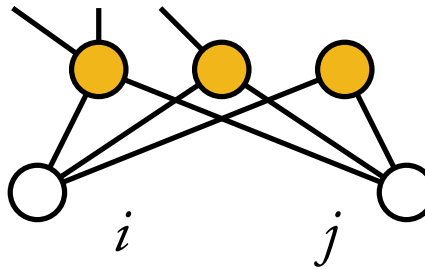


Precepts of similarity measures

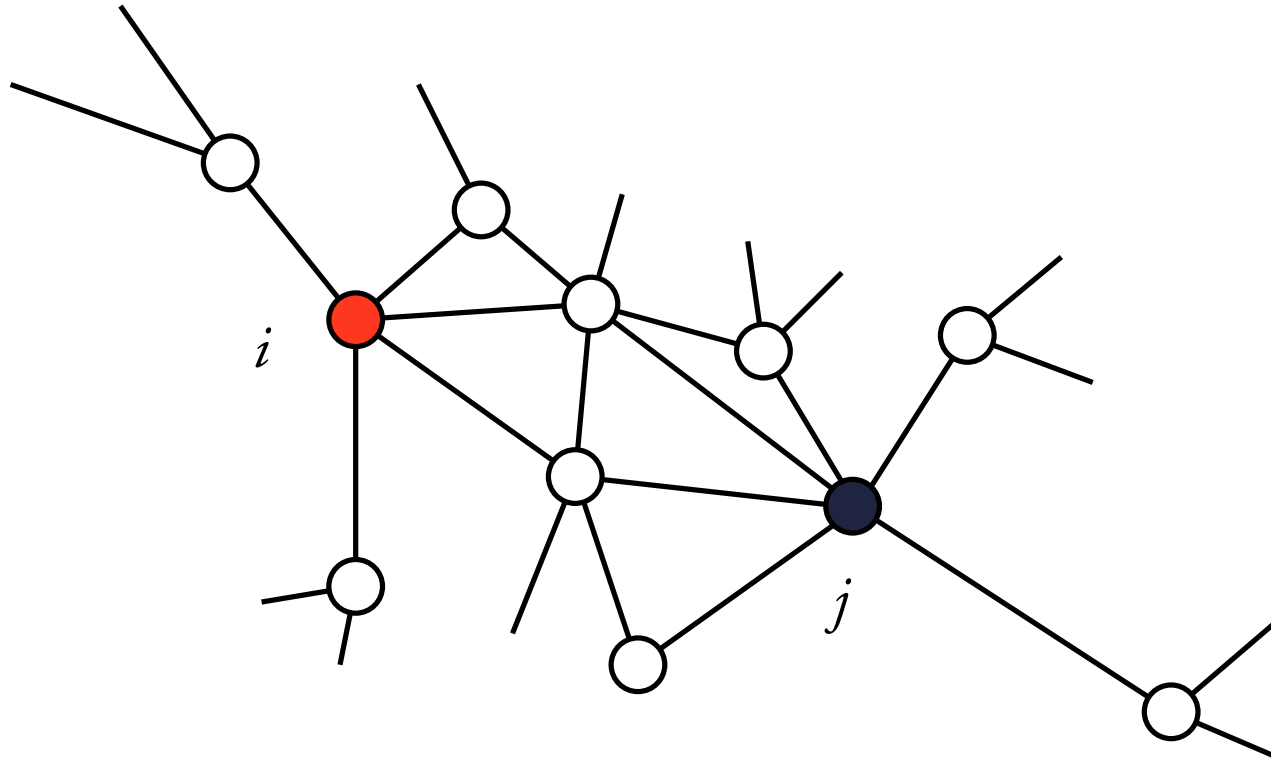
a vertex is similar to itself

two vertices are similar if their neighborhoods are similar

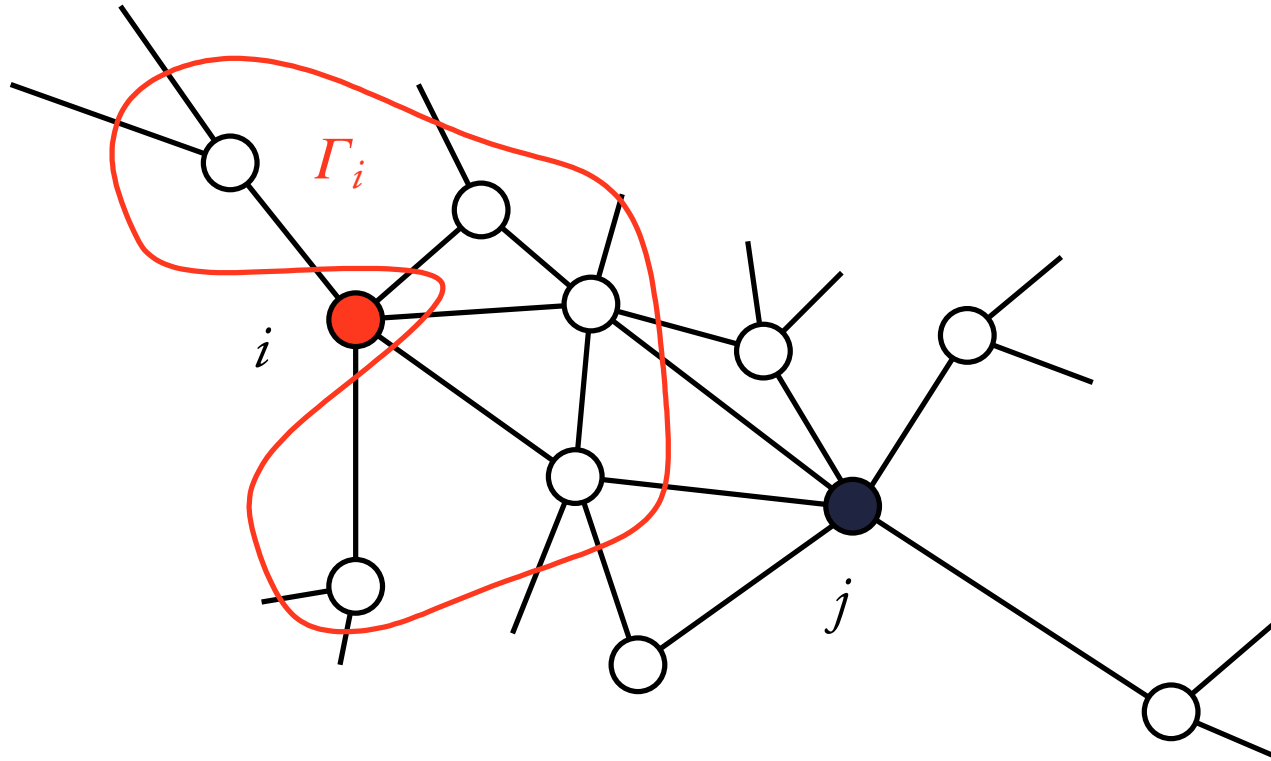
structural equivalence
/
structural similarity



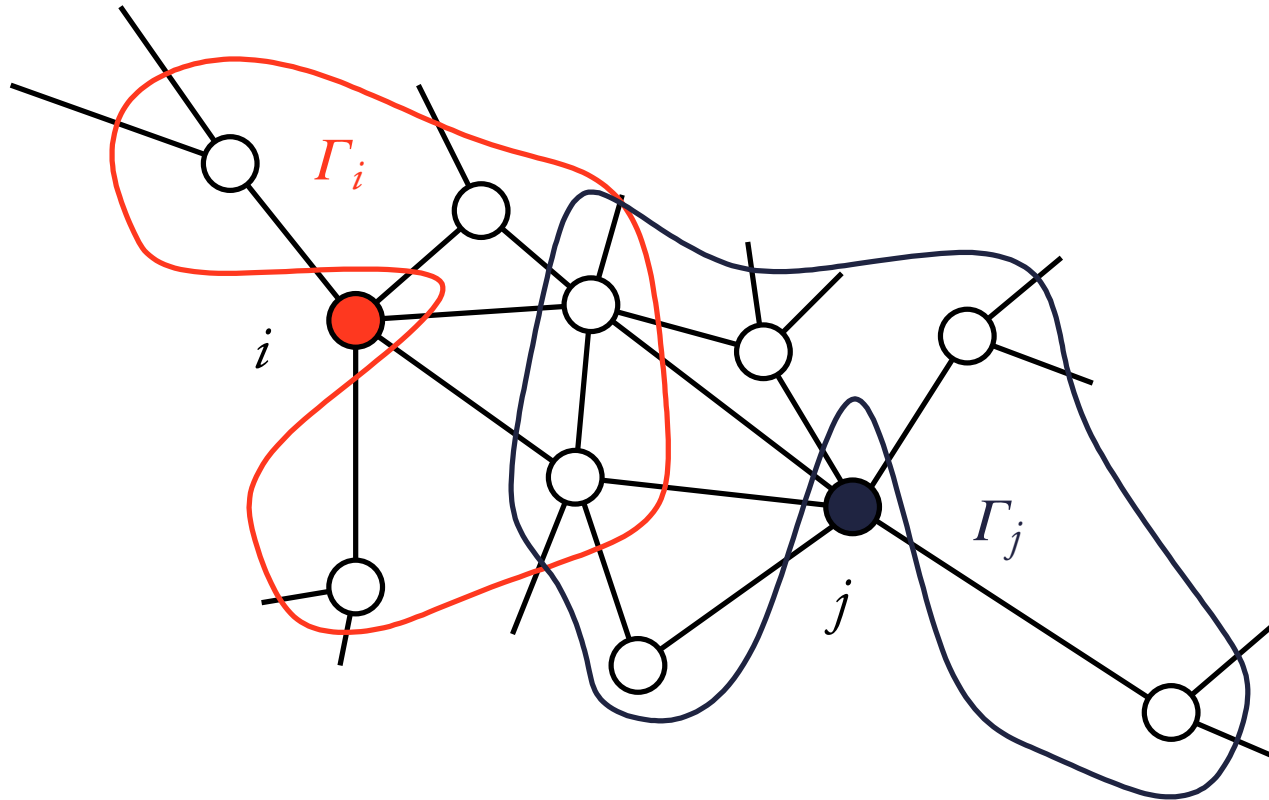
Structural similarity measures



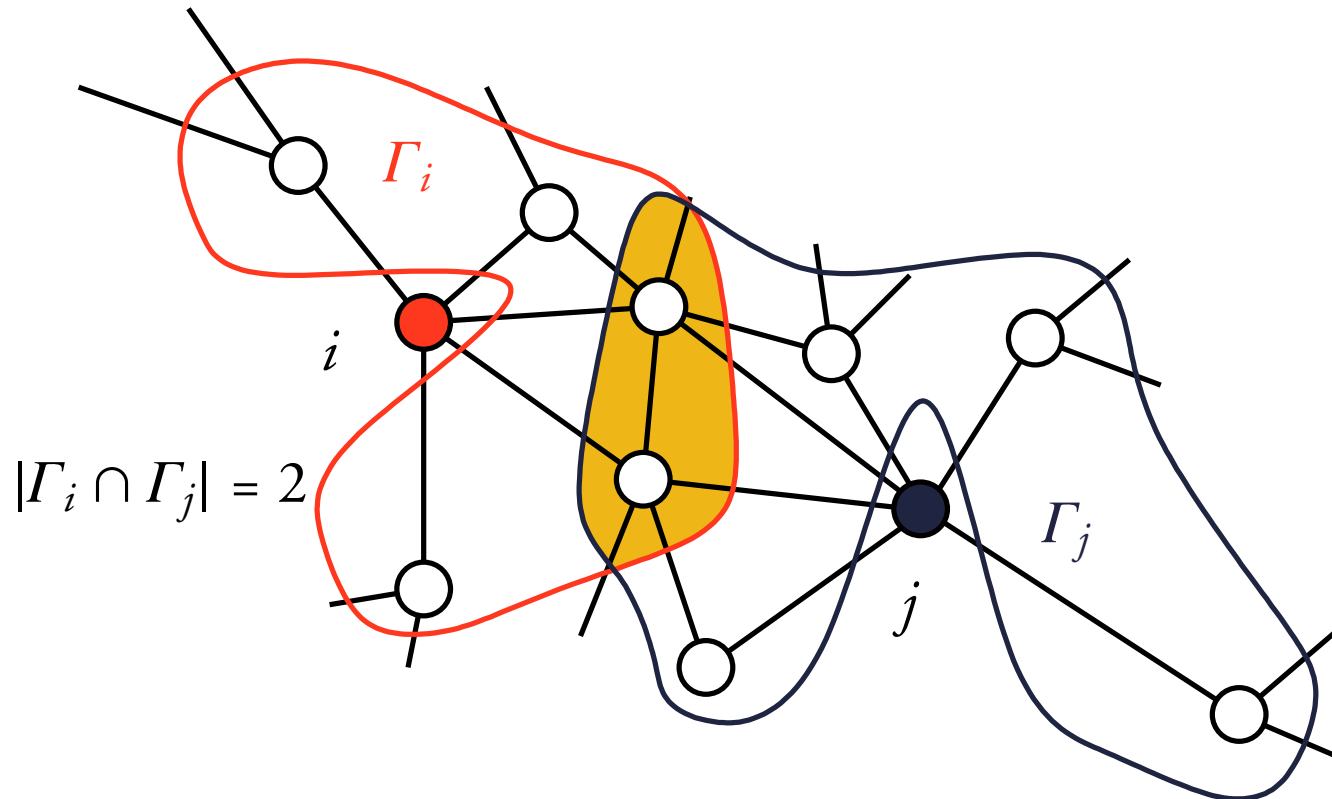
Structural similarity measures



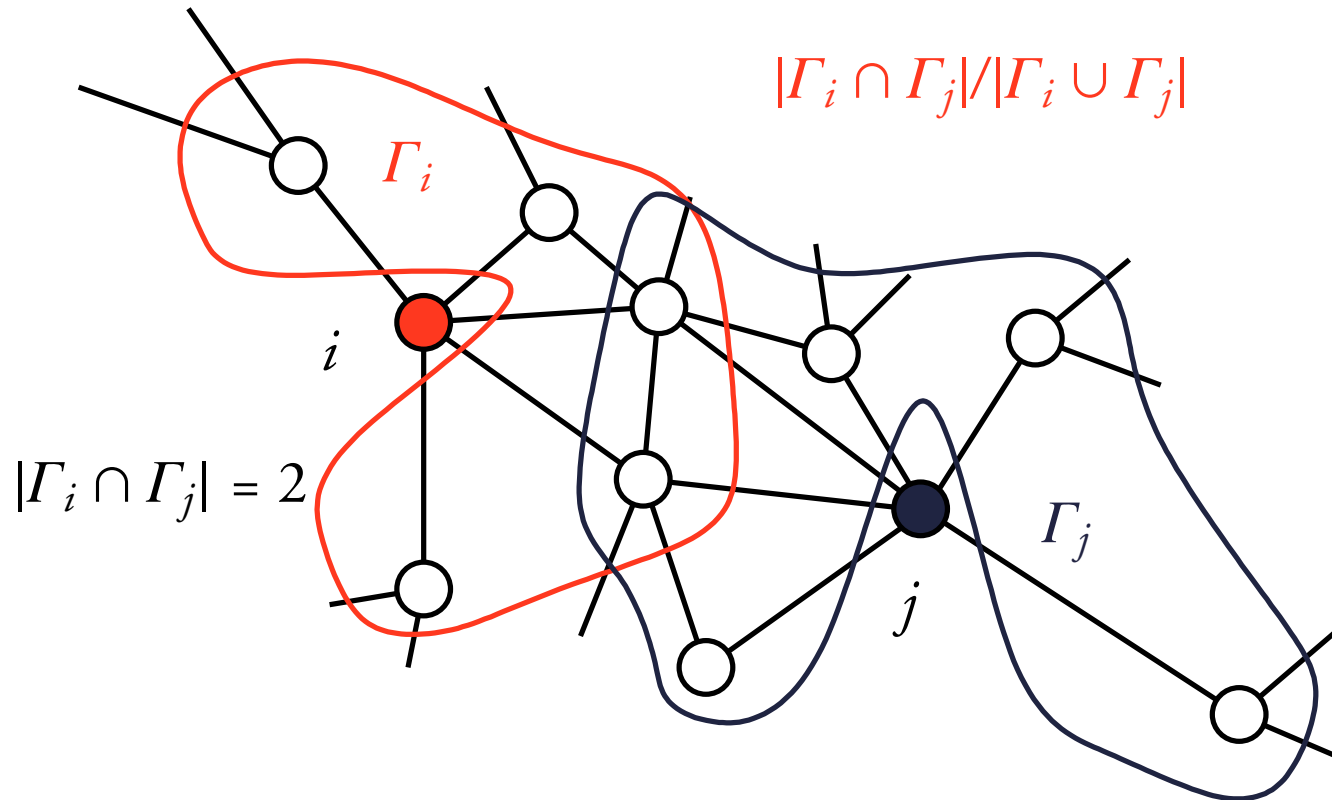
Structural similarity measures



Structural similarity measures



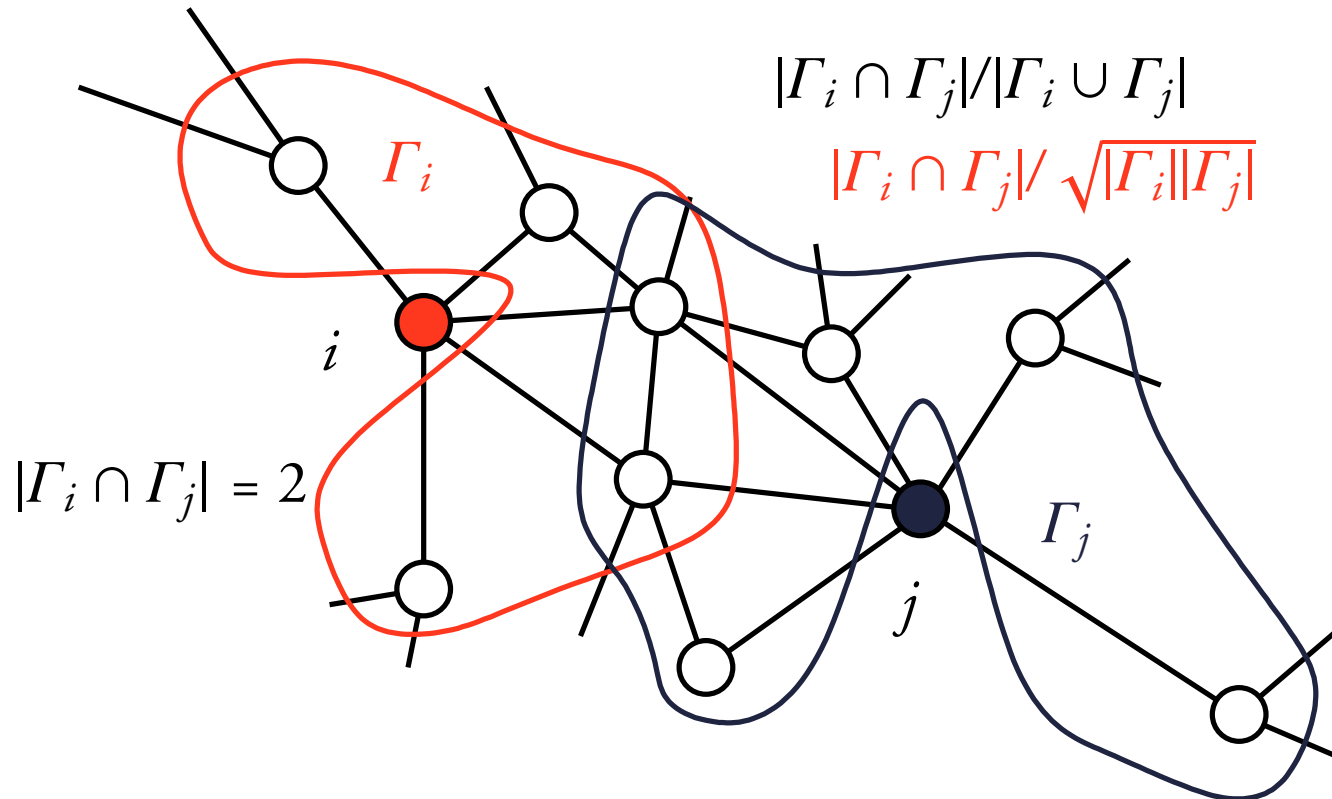
Structural similarity measures



Jaccard (1901)



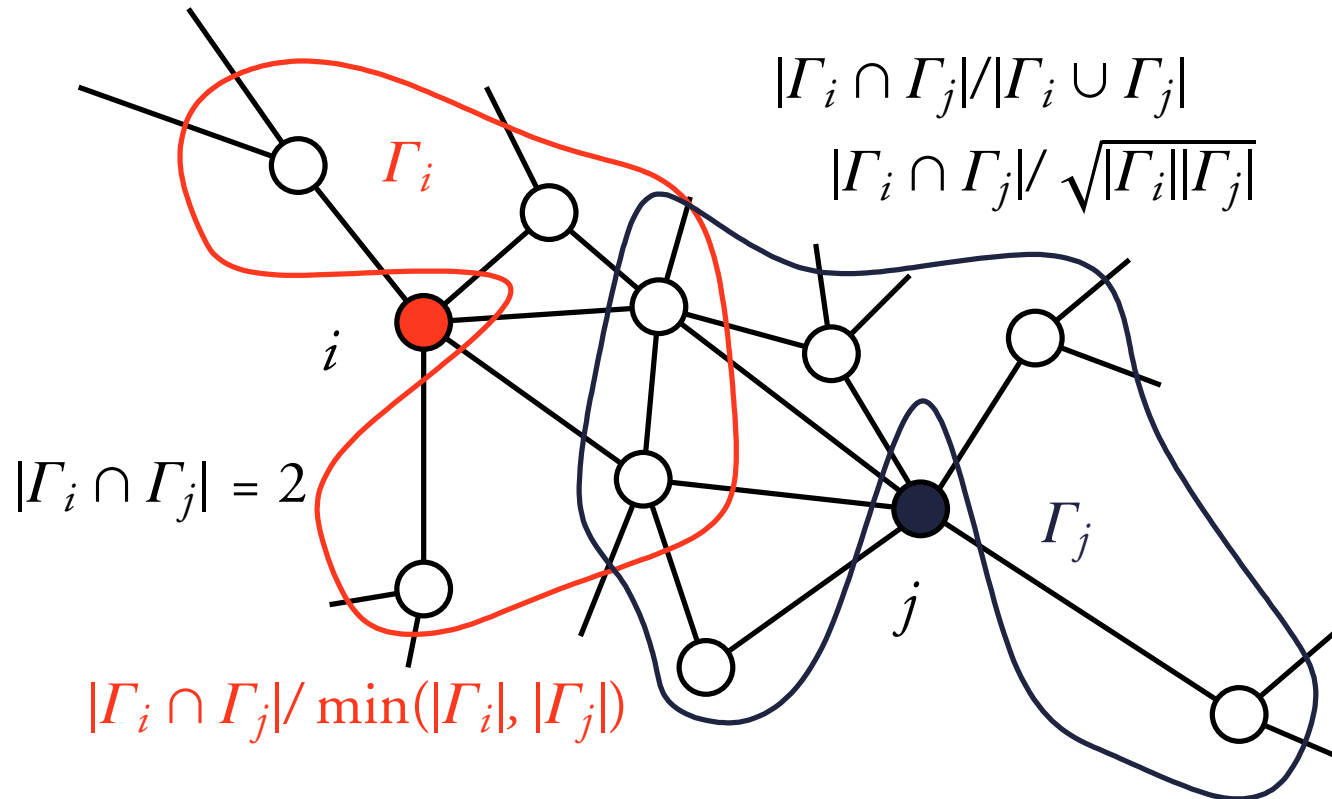
Structural similarity measures



Salton (1989)



Structural similarity measures



Ravasz *et al.* (2002)



Precepts of similarity measures

a vertex is similar to itself

two vertices are similar if their neighborhoods are similar

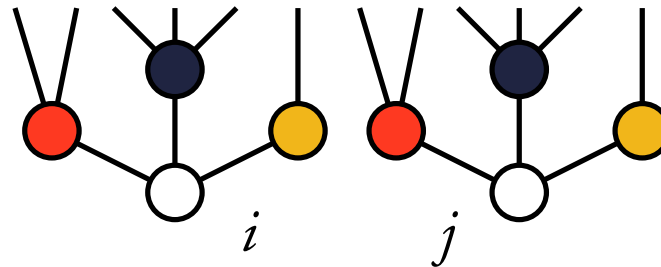


Precepts of similarity measures

a vertex is similar to itself

two vertices are similar if their neighborhoods are similar

regular equivalence
/
regular similarity



Precepts of similarity measures

a vertex is similar to itself

a vertex is similar to another
if the its neighborhood is
similar to the other vertex

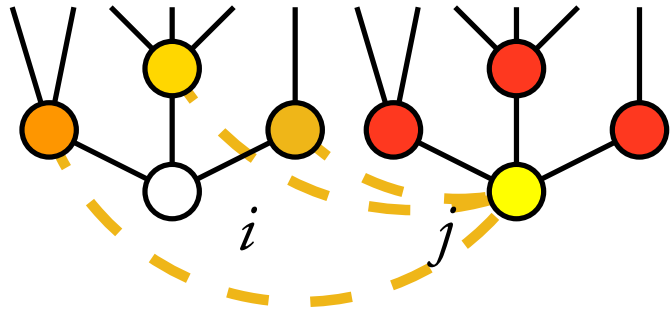


Precepts of similarity measures

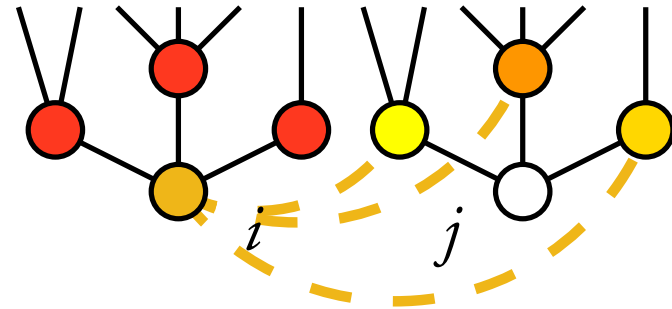
a vertex is similar to itself

a vertex is similar to another
if its neighborhood is
similar to the other vertex

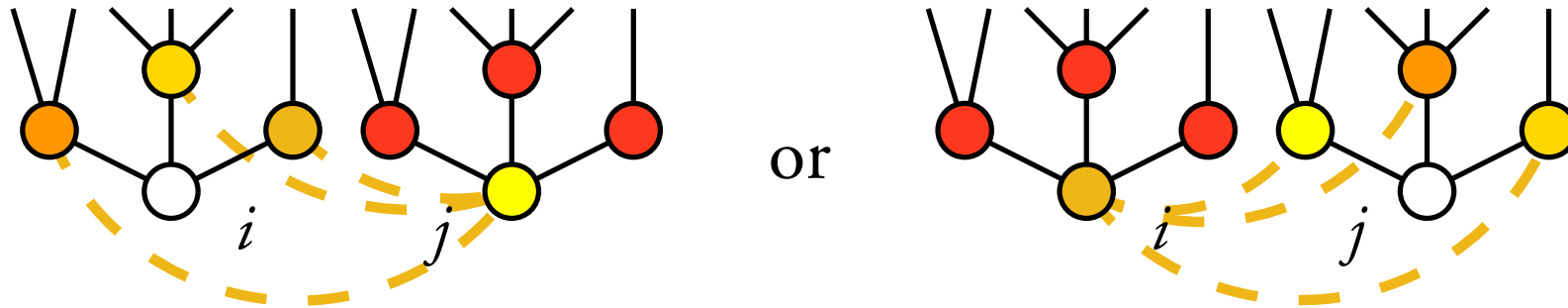
our similarity



or



Our similarity measure



A starting point...

$$S_{ij} = \phi \sum_v A_{iv} S_{vj} + \psi \delta_{ij} \Rightarrow (\text{setting } \psi = 1)$$

$$\mathbf{S} = (\mathbf{I} - \phi \mathbf{A})^{-1} = \mathbf{I} + \phi \mathbf{A} + \phi^2 \mathbf{A}^2 + \dots$$



Our similarity measure

We replace ϕ^l by individual factors C_l^{ij} representing
1 / expected # of paths of length l between i and j ...

$$S_{ij} = \sum_{l=0}^{\infty} C_l^{ij} (\mathbf{A}^l)_{ij}$$

We obtain

$$C_l^{ij} \approx \begin{cases} (2m/k_i k_j) \lambda_1^{1-l} & l \geq 1 \\ \delta_{ij} & l = 0 \end{cases}$$

Unfortunately $C_l^{ij} (\mathbf{A}^l)_{ij} \in O(1)$, so...



Our similarity measure

...we scale down each term by a factor α^l , $0 < \alpha < 1$:

$$\begin{aligned} S_{ij} &= \delta_{ij} + \frac{2m}{k_i k_j} \sum_{l=1}^{\infty} \alpha^l \lambda_1^{-l+1} [\mathbf{A}^l]_{ij} \\ &= \left[1 - \frac{2m\lambda_1}{k_i k_j} \right] \delta_{ij} + \frac{2m\lambda_1}{k_i k_j} \left[\left(\mathbf{I} - \frac{\alpha}{\lambda_1} \mathbf{A} \right)^{-1} \right]_{ij} \end{aligned}$$

...and omit the first term only contributing to the diagonal ...



Our similarity measure

$$S_{ij} = \frac{2m\lambda_1}{k_i k_j} \left[\left(\mathbf{I} - \frac{\alpha}{\lambda_1} \mathbf{A} \right)^{-1} \right]_{ij}$$

E. A. Leicht, P. Holme & M. E. J. Newman, Vertex similarity in networks, e-print physics/0510143



Evaluation: Model

Stratified network model:



Evaluation: Model

Stratified network model:

- Assign an “age” $t = 1, \dots, 10$ to N vertices with uniform randomness.



Evaluation: Model

Stratified network model:

- Assign an “age” $t = 1, \dots, 10$ to N vertices with uniform randomness.
- Let there be a link between i and j with probability $P(\Delta t) = p_0 \exp(-a\Delta t)$. (We choose $p_0 = 0.12$ and $a = 2.0$.)



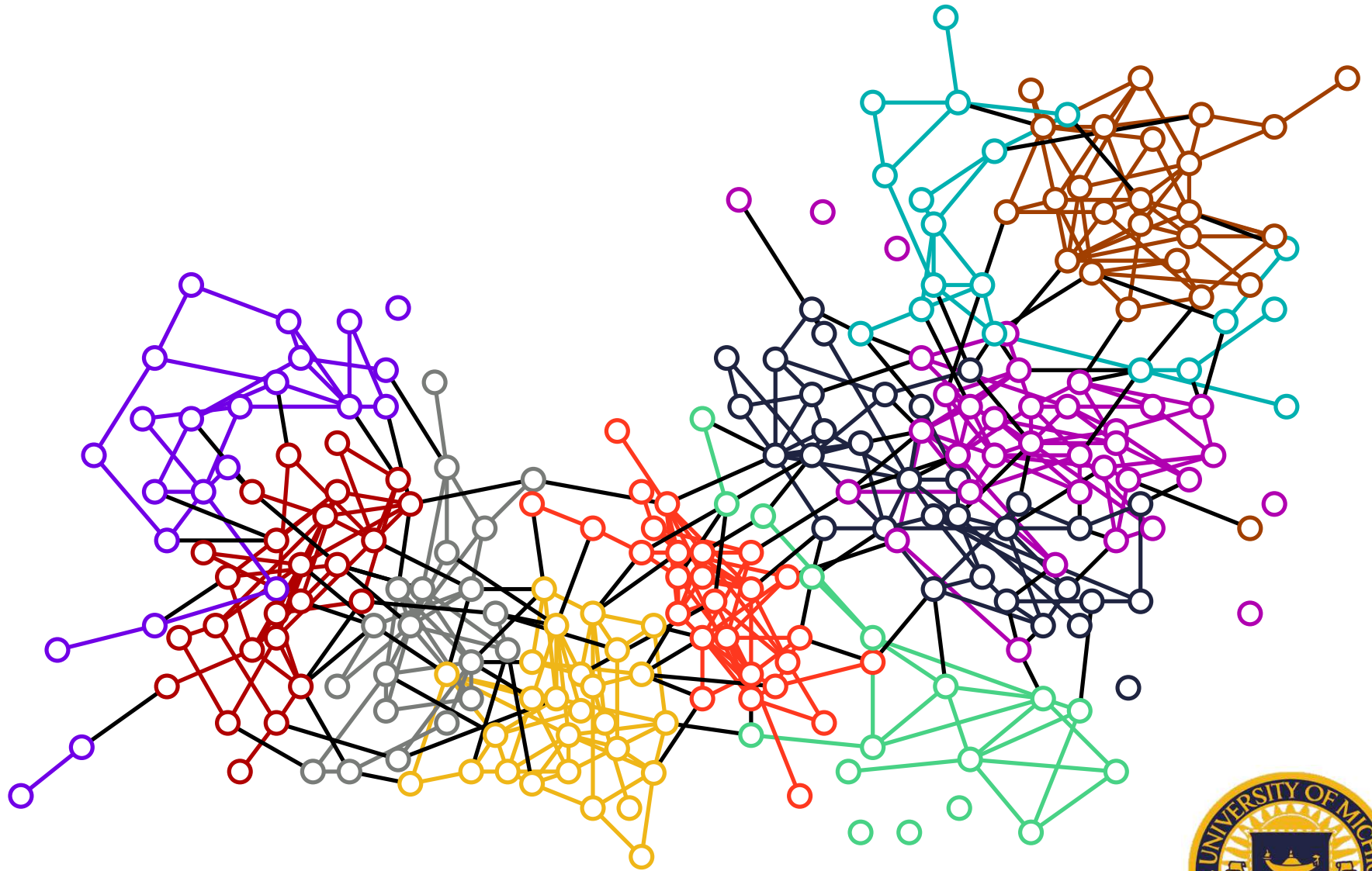
Evaluation: Model

Stratified network model:

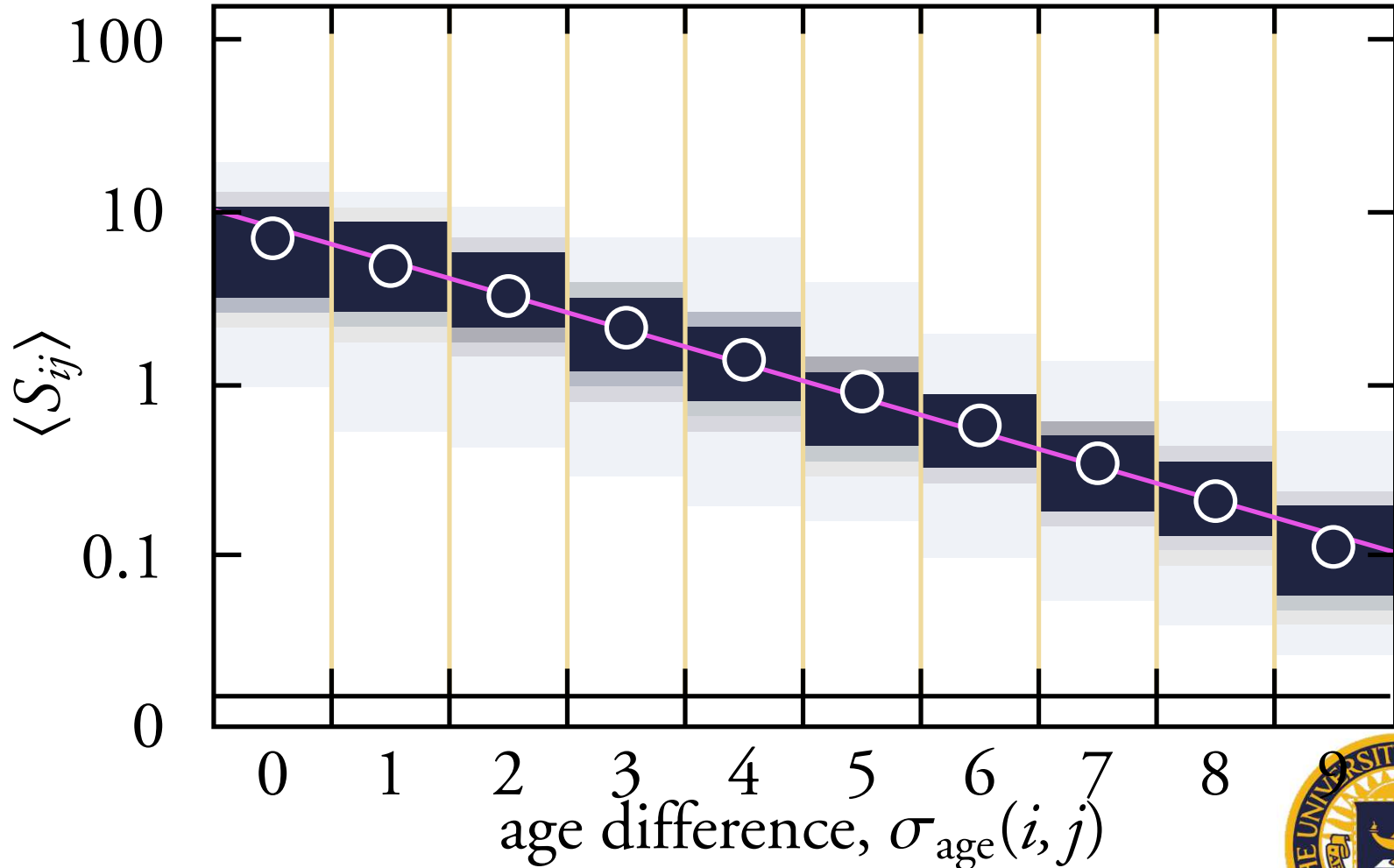
- Assign an “age” $t = 1, \dots, 10$ to N vertices with uniform randomness.
- Let there be a link between i and j with probability $P(\Delta t) = p_0 \exp(-a\Delta t)$. (We choose $p_0 = 0.12$ and $a = 2.0$.)
- The probability of a link drops by a factor of e^a for every additional year separating their ages.



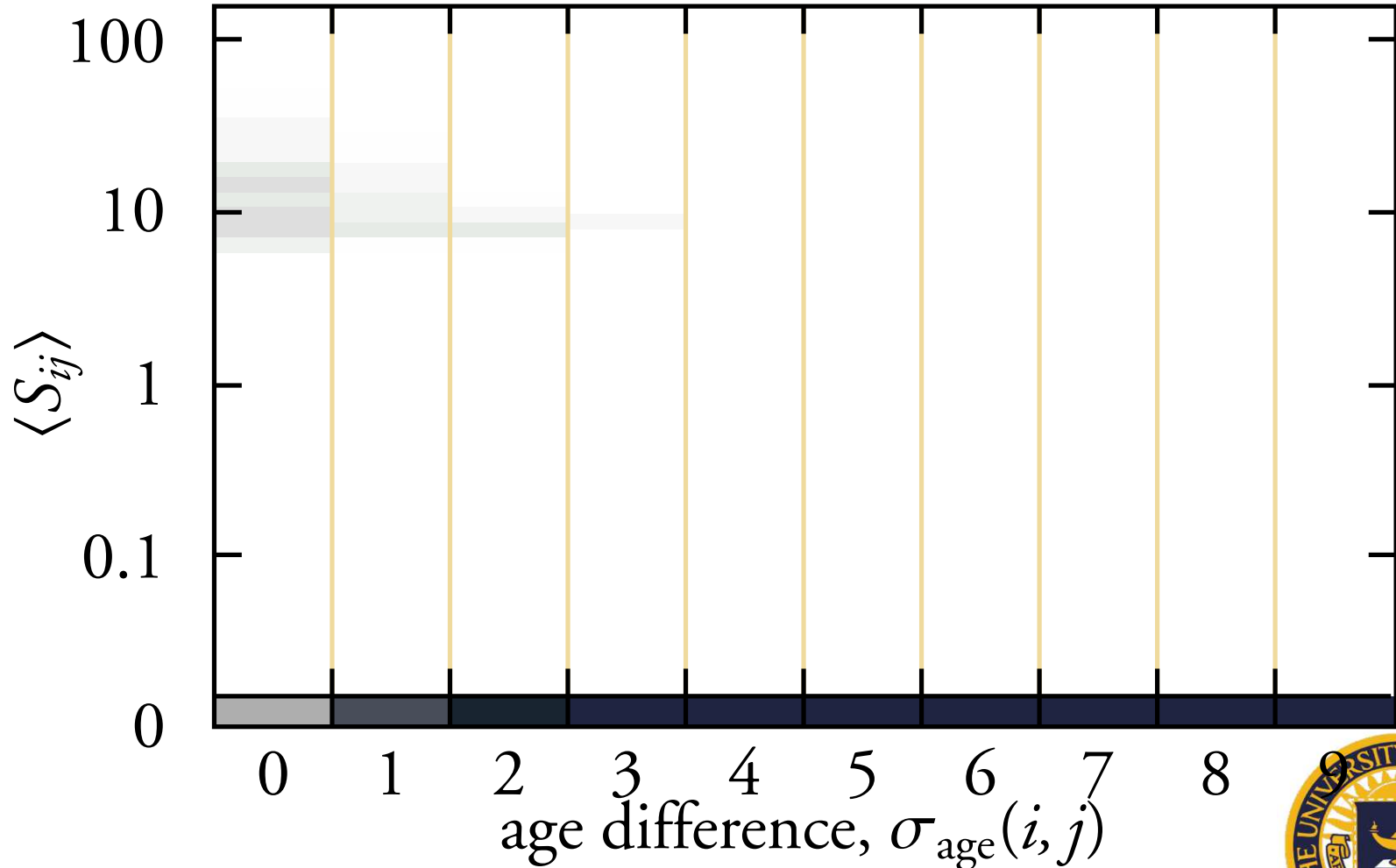
Evaluation: Model



Evaluation: Model



Evaluation: Model



Evaluation: Roget's Thesaurus

Classes

(Divisions)

Sections

Subsections

Words



Evaluation: Roget's Thesaurus

Words Expressing Abstract Relations

Words Relating to the Sentient and Moral Powers

Words Relating to the Intellectual Faculties

Words Relating to Space

(Divisions)

Sections

Subsections

Words



Evaluation: Roget's Thesaurus

Words Expressing Abstract Relations

Words Relating to the Sentient and Moral Powers

Words Relating to the Intellectual Faculties

Affections in General

Words Relating to Space

Personal Affections

Sympathetic Affections

Religious Affections

Subsections

Words



Evaluation: Roget's Thesaurus

Words Expressing Abstract Relations

Words Relating to the Sentient and Moral Powers

Words Relating to the Intellectual Faculties

Affections in General

Words Relating to Space

Personal Affections

Sympathetic Affections

Religious Affections

Religious doctrines

Superhuman beings and regions

Religious Sentiments

Words



Evaluation: Roget's Thesaurus

Words Expressing Abstract Relations

Words Relating to the Sentient and Moral Powers

Words Relating to the Intellectual Faculties

Affections in General

Words Relating to Space

Personal Affections

Sympathetic Affections

Religious Affections

Religious doctrines

Superhuman beings and regions

Deity

Angel

Satan

Heaven

Hell

Religious Sentiments



Evaluation: Roget's Thesaurus

word	our measure		cosine similarity	
alarm	warning	32.0	omen	0.516
	danger	25.8	threat	0.471
	omen	18.8	prediction	0.348
hell	heaven	63.4	pleasure	0.408
	pain	28.9	inferiority	0.222
	discontent	7.0	weariness	0.267
water	plunge	33.6	dryness	0.447
	air	25.3	wind	0.316
	moisture	25.3	ocean	0.316



Evaluation: AddHealth

- Friendship network of school children.



Evaluation: AddHealth

- Friendship network of school children.
- 90 118 students at 168 schools.

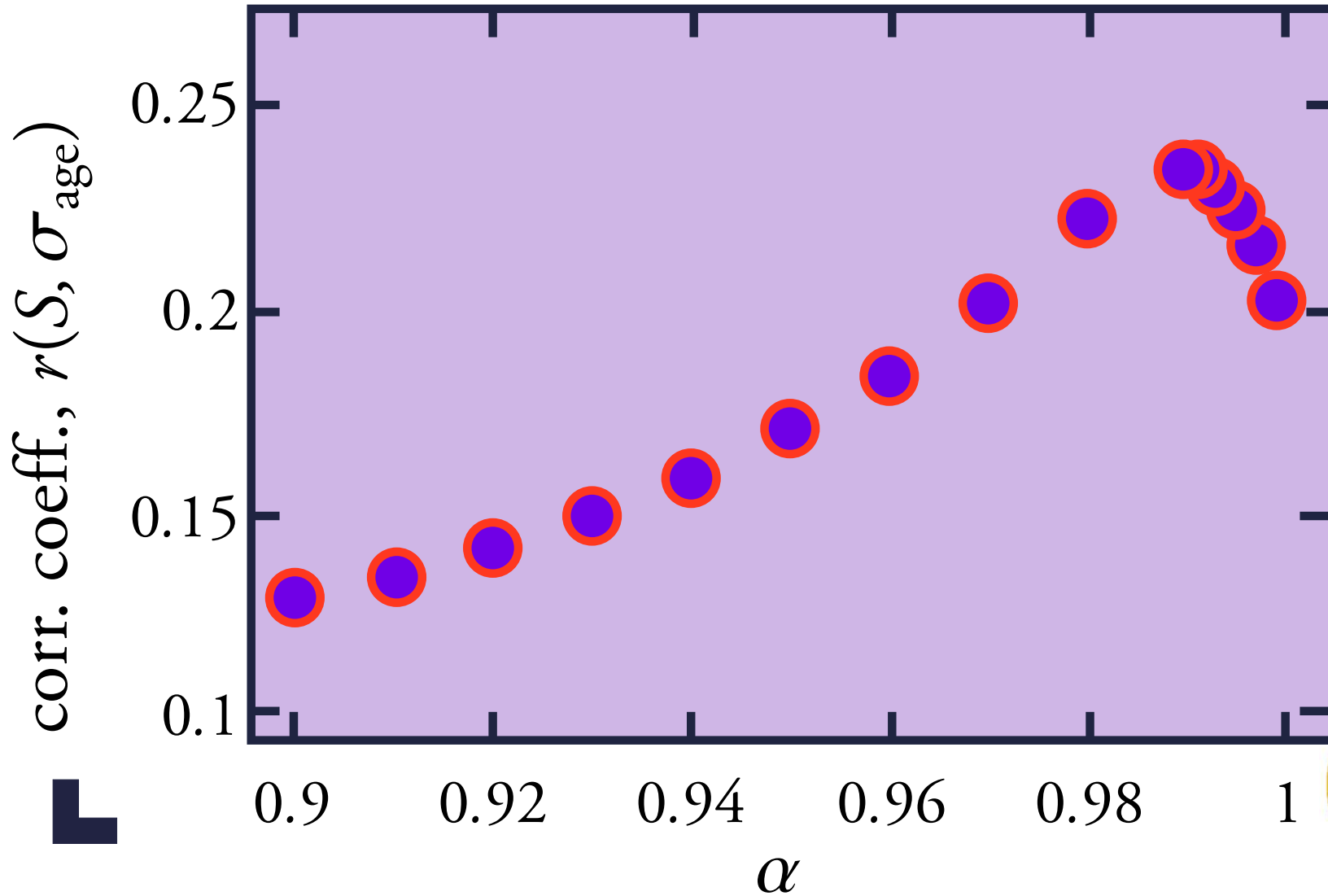


Evaluation: AddHealth

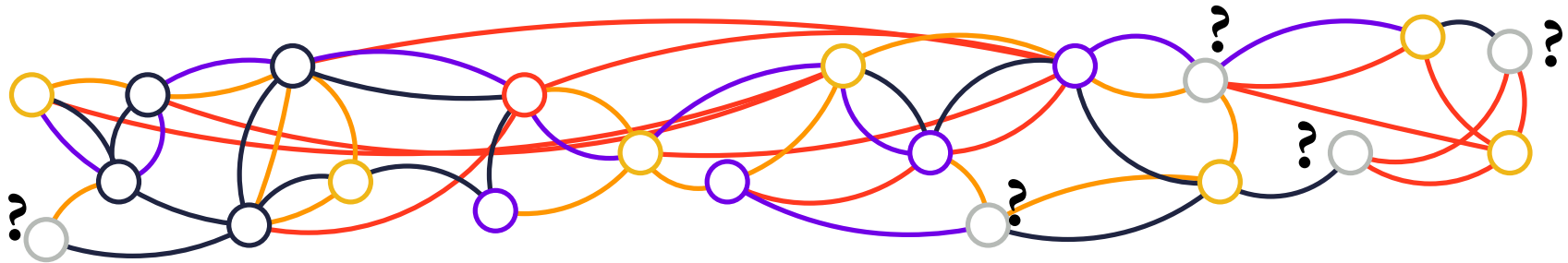
- Friendship network of school children.
- 90 118 students at 168 schools.
- Information about grade, race and gender



Evaluation: AddHealth



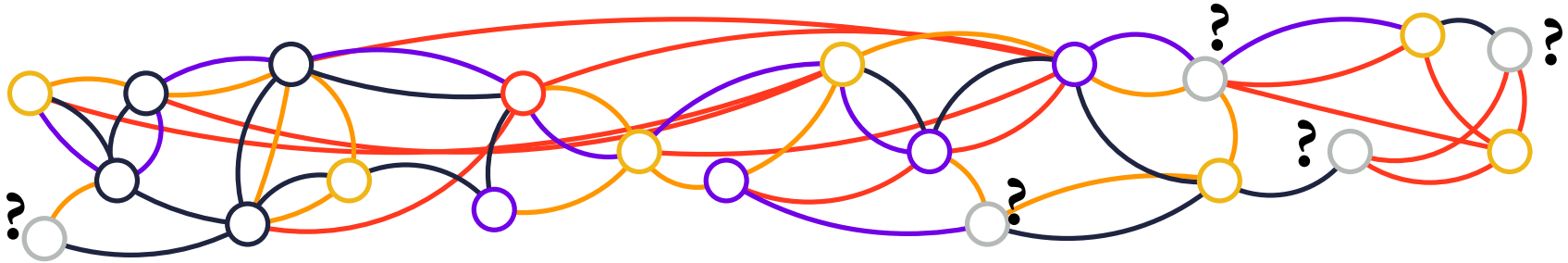
Similarity > > > prediction



Imagine a system with:



Similarity >>> prediction

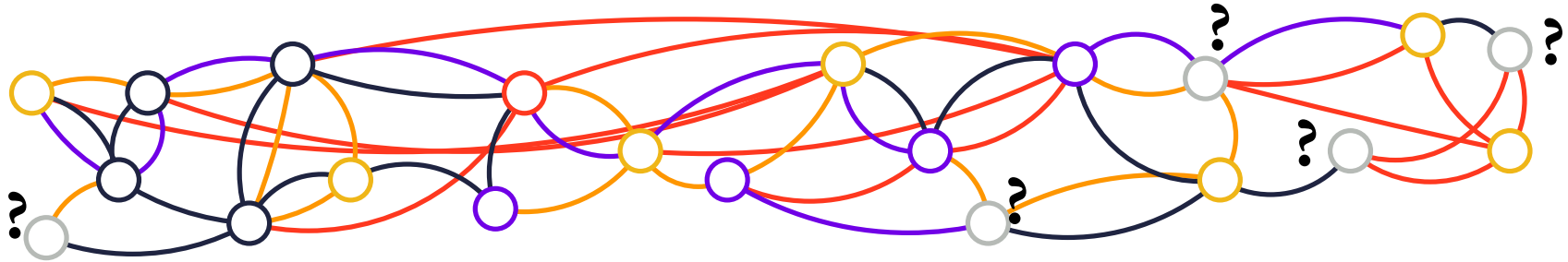


Imagine a system with:

- Its network structure known.
- The function of a fraction $r \in (0, 1)$ of the vertices classified.



Similarity >>> prediction



Imagine a system with:

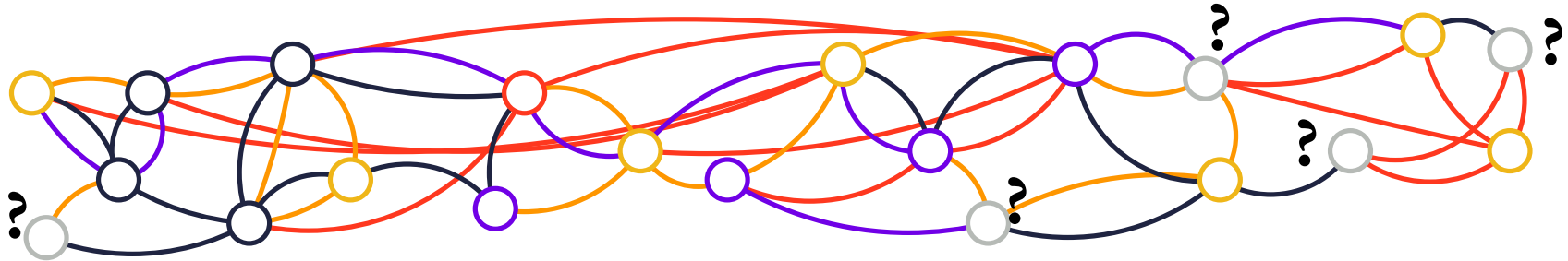
- Its network structure known.
- The function of a fraction $r \in (0, 1)$ of the vertices classified.

How can we assess the function of unclassified vertices?

P. Holme & M. Huss, Role-similarity based functional prediction in networked systems: application to the yeast proteome, *J. Roy. Soc. Interface* **2**, (2005) pp. 327-333.



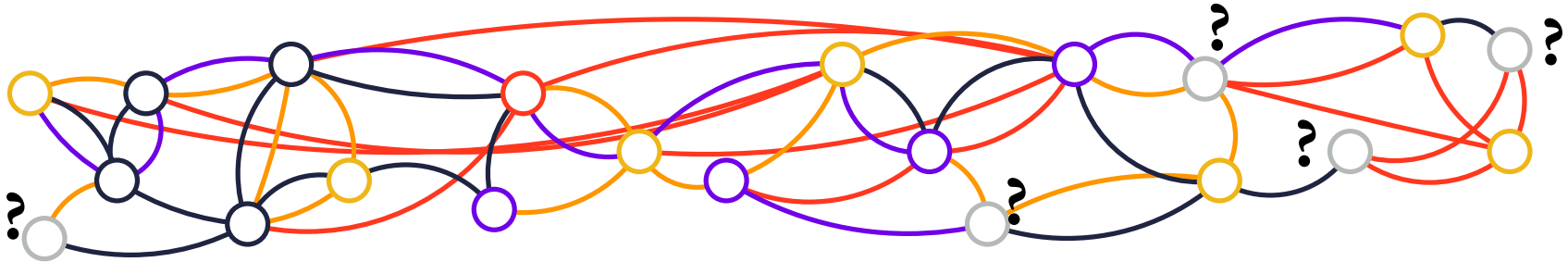
Prediction algorithm



- Assign a functional similarity between classified vertices. (If vertices can have more than one function, we can use the Jaccard index $|F_i \cap F_j|/|F_i \cup F_j|$.)



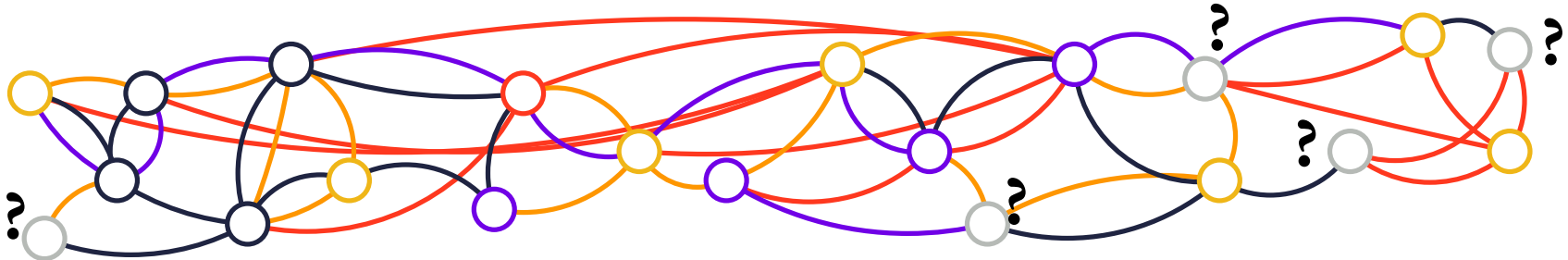
Prediction algorithm



- Assign a functional similarity between classified vertices. (If vertices can have more than one function, we can use the Jaccard index $|F_i \cap F_j|/|F_i \cup F_j|$.)
- Set $S_{ij} = \delta_{ij}$ if i or j is unclassified.



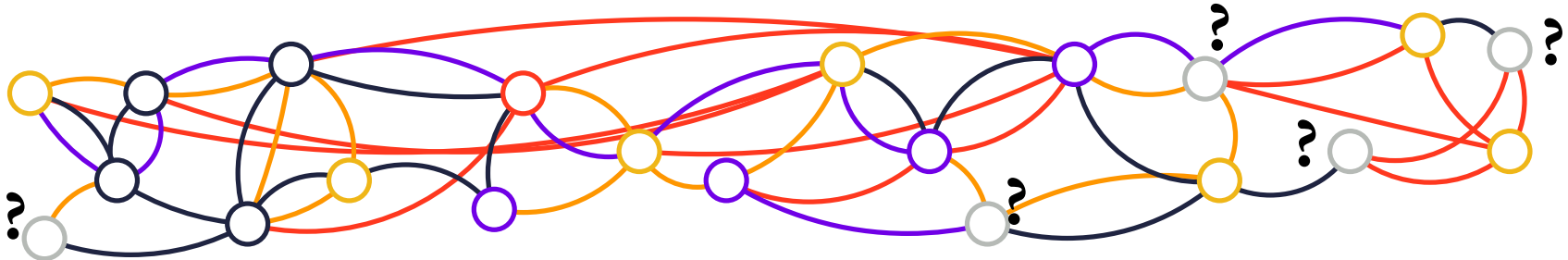
Prediction algorithm



- Assign a functional similarity between classified vertices. (If vertices can have more than one function, we can use the Jaccard index $|F_i \cap F_j|/|F_i \cup F_j|$.)
- Set $S_{ij} = \delta_{ij}$ if i or j is unclassified.
- Update S_{ij} (i or j is unclassified) iteratively.



Prediction algorithm

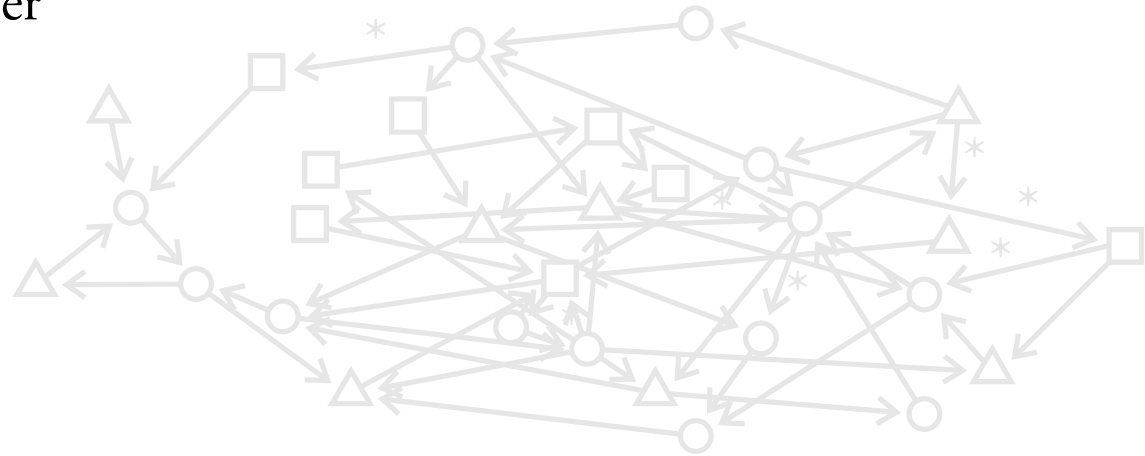


- Assign a functional similarity between classified vertices. (If vertices can have more than one function, we can use the Jaccard index $|F_i \cap F_j|/|F_i \cup F_j|$.)
- Set $S_{ij} = \delta_{ij}$ if i or j is unclassified.
- Update S_{ij} (i or j is unclassified) iteratively.
- For an unclassified vertex i let the functions of the most similar classified vertex be your guess.



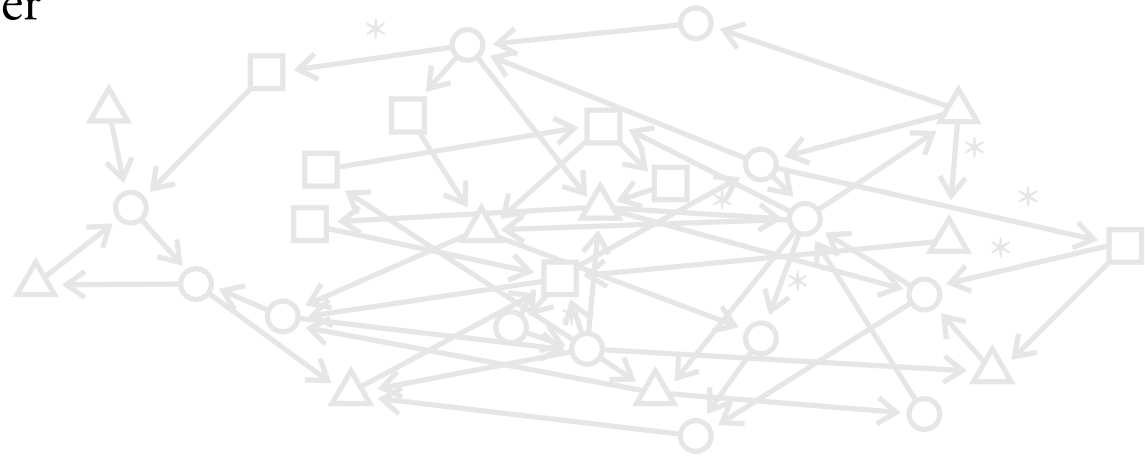
Testing prediction: Model

- supply
- delivery
- A-distributor
- B-distributor
- △ assembler
- A-edge
- B-edge
- C-edge



Testing prediction: Model

- supply
- delivery
- A-distributor
- B-distributor
- △ assembler
- A-edge
- B-edge
- C-edge



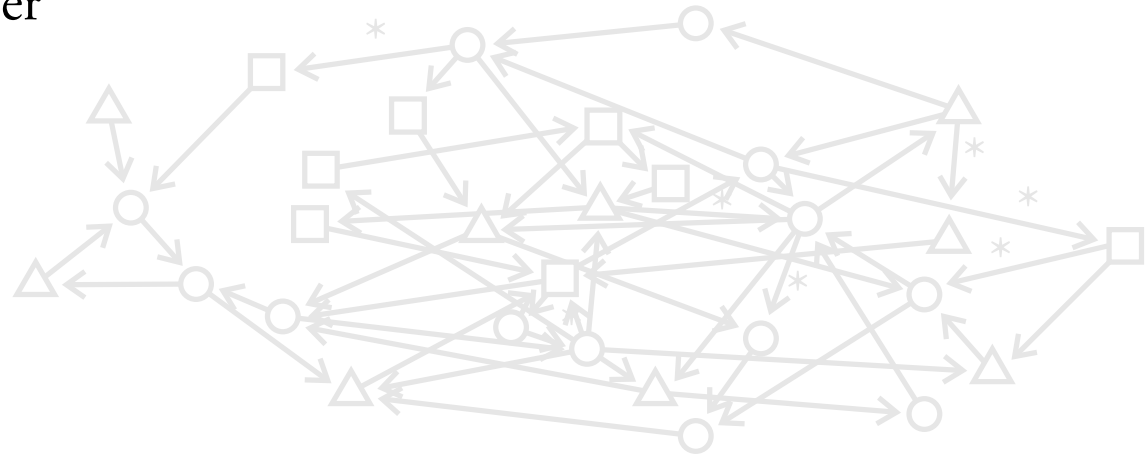
add supply vertex

(● → ■ or ● → △) and ○ → ●



Testing prediction: Model

- supply
- delivery
- A-distributor
- B-distributor
- △ assembler
- A-edge
- B-edge
- C-edge



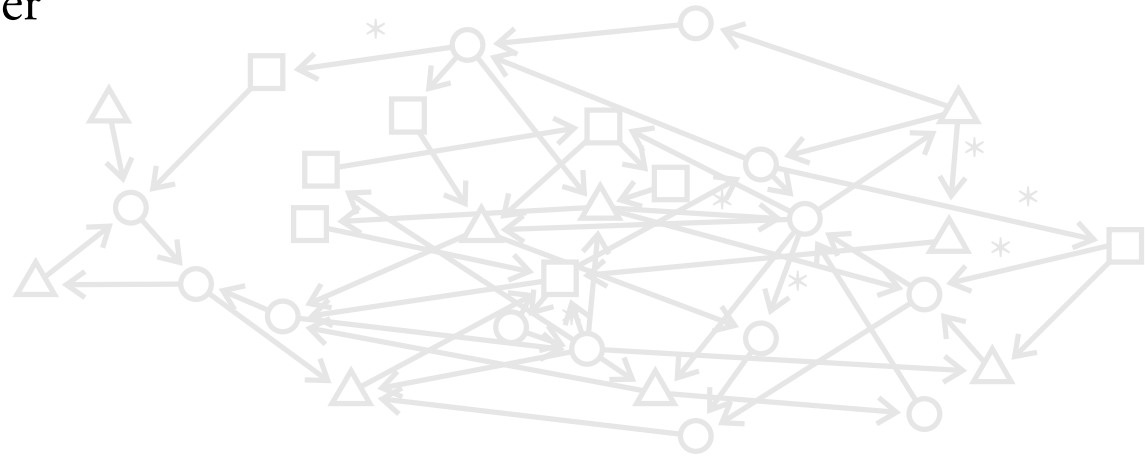
add assembler vertex

(△ → □ or △ → ○) and (■ → △ or ● → △)



Testing prediction: Model

- supply ▲ assembler
- delivery → A-edge
- A-distributor → B-edge
- B-distributor → C-edge



add delivery vertex

○ → ● and (▲ → ○ or □ → ○)



Testing prediction: Model

- supply
- delivery
- A-distributor
- B-distributor
- △ assembler
- A-edge
- B-edge
- C-edge



add A-distributor vertex

(■ → ■ or ■ → ○) and (■ → ■ or ● → ■)



Testing prediction: Model

- supply
- delivery
- A-distributor
- B-distributor
- △ assembler
- A-edge
- B-edge
- C-edge



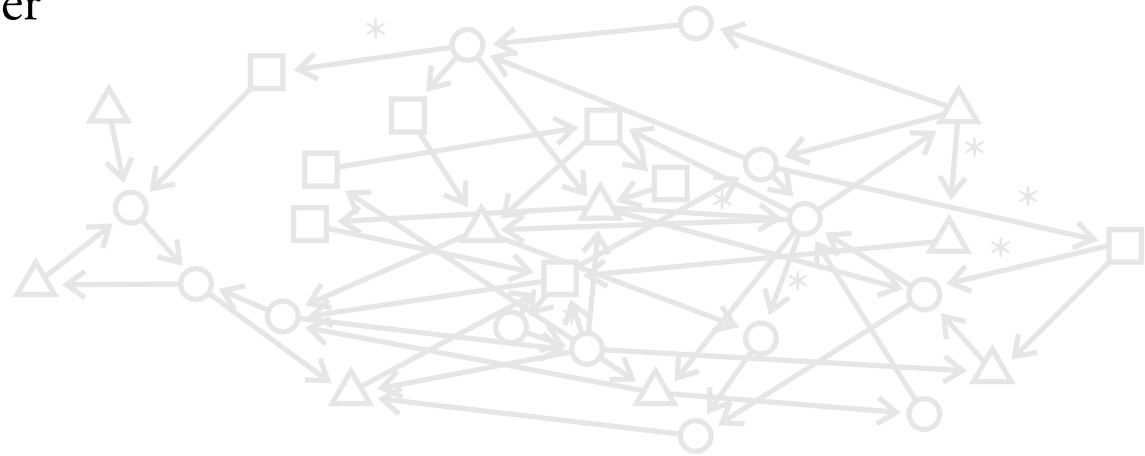
add B-distributor vertex

(□ → □ or □ → ○) and (□ → □ or ● → □)



Testing prediction: Model

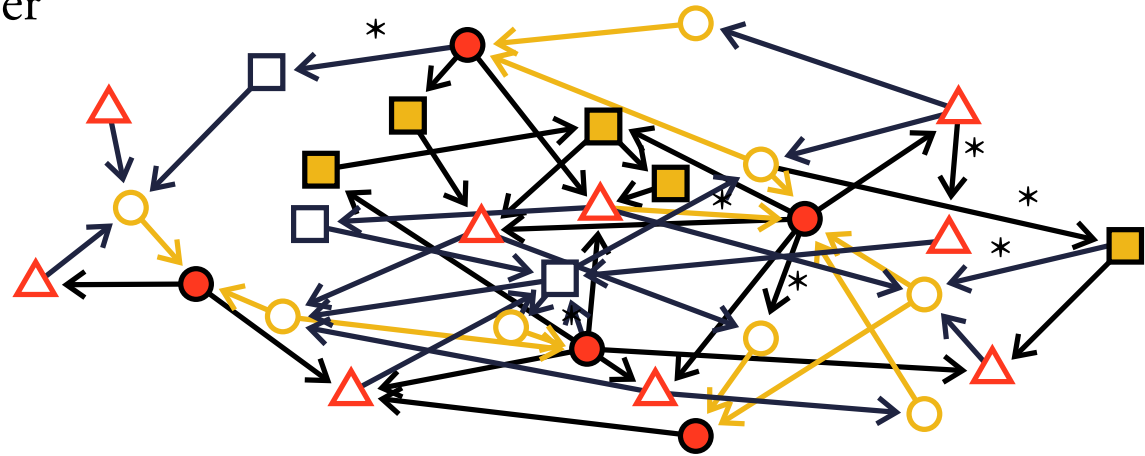
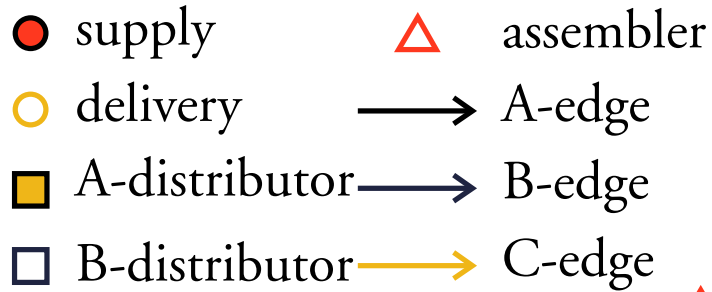
- supply
- delivery
- A-distributor
- B-distributor
- △ assembler
- A-edge
- B-edge
- C-edge



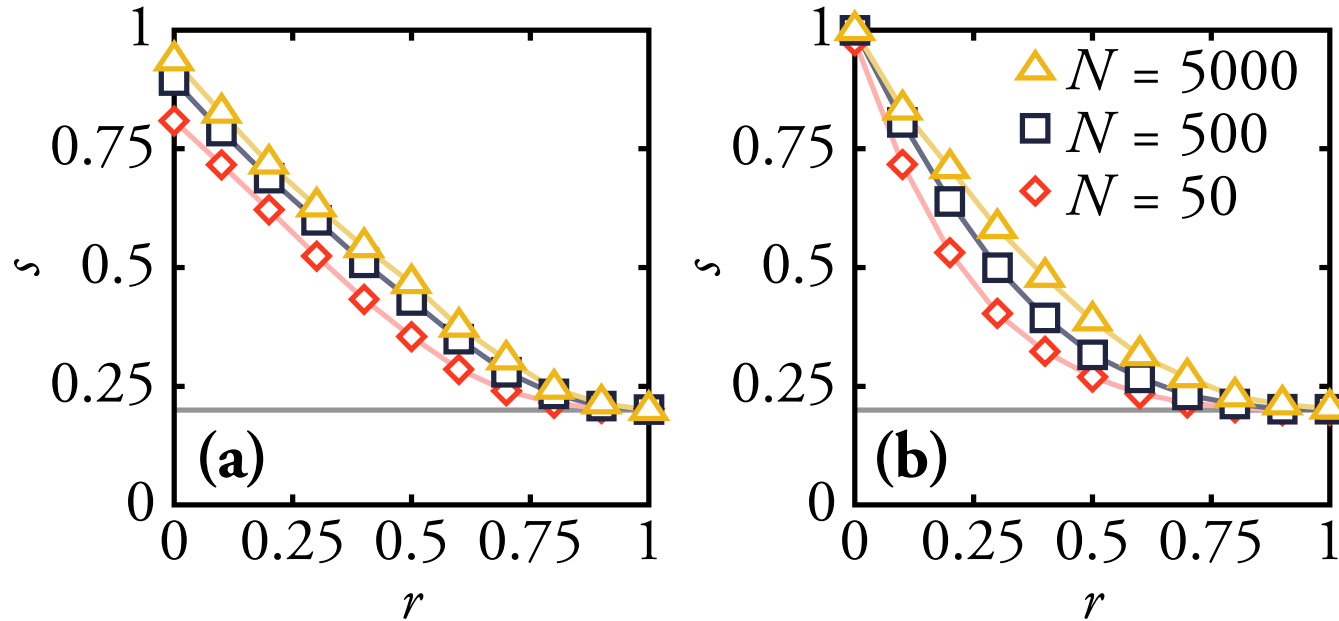
rewire the edges with probability r



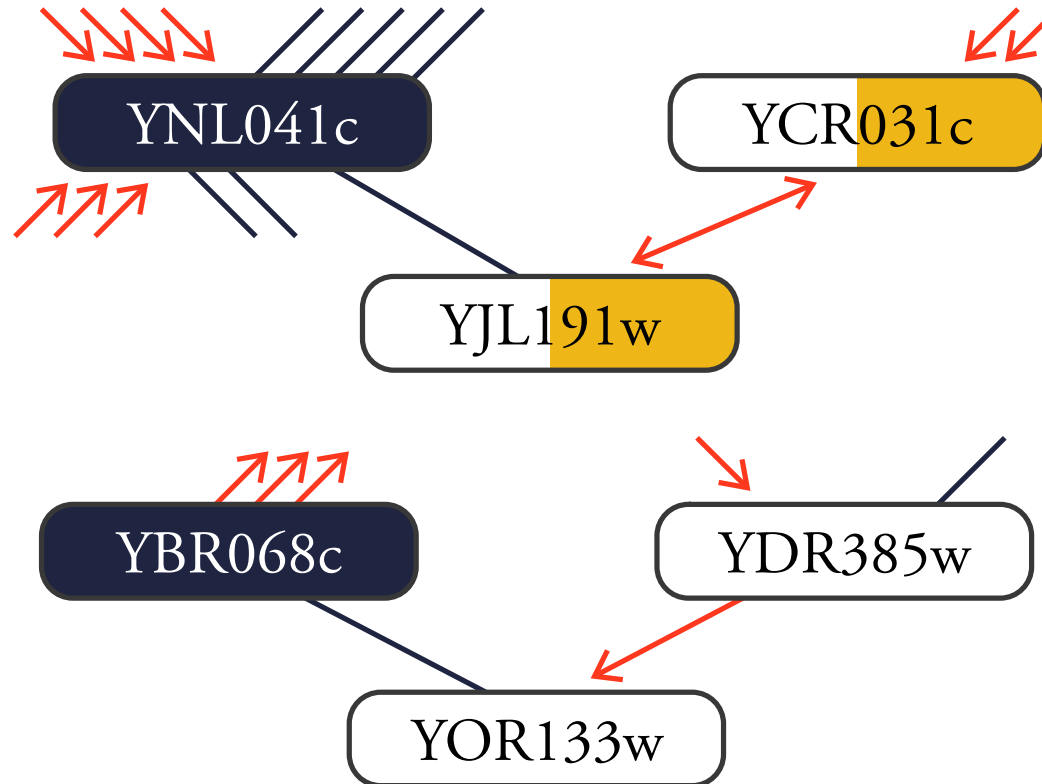
Testing prediction: Model






Testing prediction: Model



Testing prediction: Yeast



-  protein synthesis
-  protein with binding function or cofactor requirement (structural or catalytic)
-  cellular transport, transport facilitation and transport routes



Testing prediction: Yeast

$$s_+ = \left\langle \frac{n_c}{f_*} \right\rangle \text{ (precision) and } s_- = \left\langle \frac{n_c}{f} \right\rangle \text{ (recall)}$$

where n_c is the number of correctly predicted functions, f is the real number of functions and f_* is the number of predicted functions.



Testing prediction: Yeast

$$s_+ = \left\langle \frac{n_c}{f_*} \right\rangle \text{ (precision) and } s_- = \left\langle \frac{n_c}{f} \right\rangle \text{ (recall)}$$

where n_c is the number of correctly predicted functions, f is the real number of functions and f_* is the number of predicted functions.

	level 1			level 2		
	NCM	our I	our II	NCM	our I	our II
s_+	0.269	0.392	0.337	0.199	0.238	0.220
s_-	0.354	0.291	0.346	0.252	0.199	0.231



Conclusions

- Vertex similarity measures serve to show which vertex-pairs that have the same function in the network.



Conclusions

- Vertex similarity measures serve to show which vertex-pairs that have the same function in the network.
- If it is formulated as a linear algebra problem it turns into a measure based on path counts.



Conclusions

- Vertex similarity measures serve to show which vertex-pairs that have the same function in the network.
- If it is formulated as a linear algebra problem it turns into a measure based on path counts.

- $$S_{ij} = \frac{2m\lambda_1}{k_i k_j} \left[\left(\mathbf{I} - \frac{\alpha}{\lambda_1} \mathbf{A} \right)^{-1} \right]_{ij}$$



Conclusions

- Vertex similarity measures serve to show which vertex-pairs that have the same function in the network.
- If it is formulated as a linear algebra problem it turns into a measure based on path counts.
- $$S_{ij} = \frac{2m\lambda_1}{k_i k_j} \left[\left(\mathbf{I} - \frac{\alpha}{\lambda_1} \mathbf{A} \right)^{-1} \right]_{ij}$$
- Similarity measures can be turned into classification and prediction algorithms.



Conclusions

- Vertex similarity measures serve to show which vertex-pairs that have the same function in the network.
- If it is formulated as a linear algebra problem it turns into a measure based on path counts.
- $$S_{ij} = \frac{2m\lambda_1}{k_i k_j} \left[\left(\mathbf{I} - \frac{\alpha}{\lambda_1} \mathbf{A} \right)^{-1} \right]_{ij}$$
- Similarity measures can be turned into classification and prediction algorithms.
- Good in general, but for specific systems there can be smarter prediction / classification algorithms.

