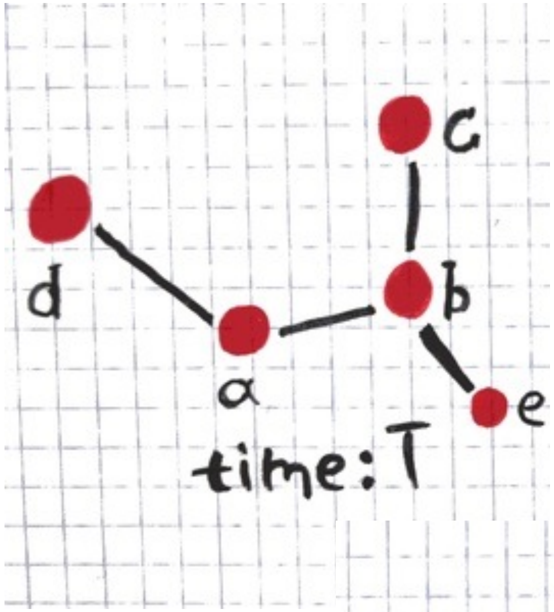
Network Science

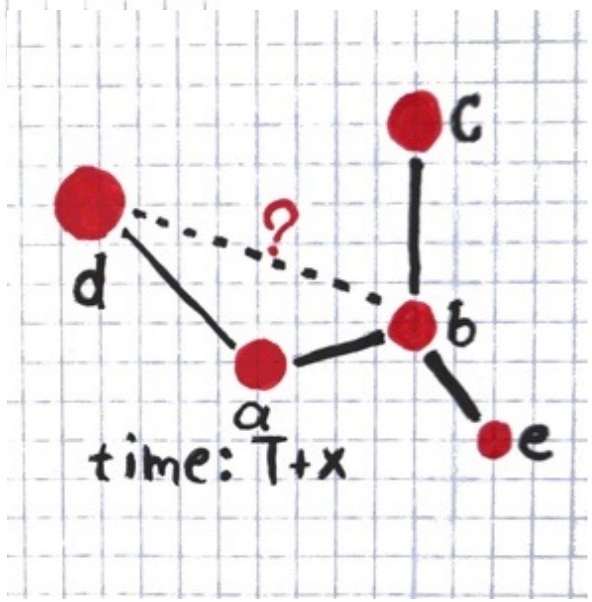
#20 Link prediction

Link Prediction

The link prediction task

Given a graph at time T, can we output a **ranked list** of links that are **predicted** to appear in the graph at time T+x ?



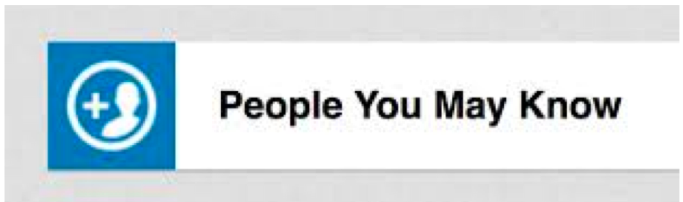


idea

We can build the list by using a measure of **similarity**/proximity between nodes

Applications

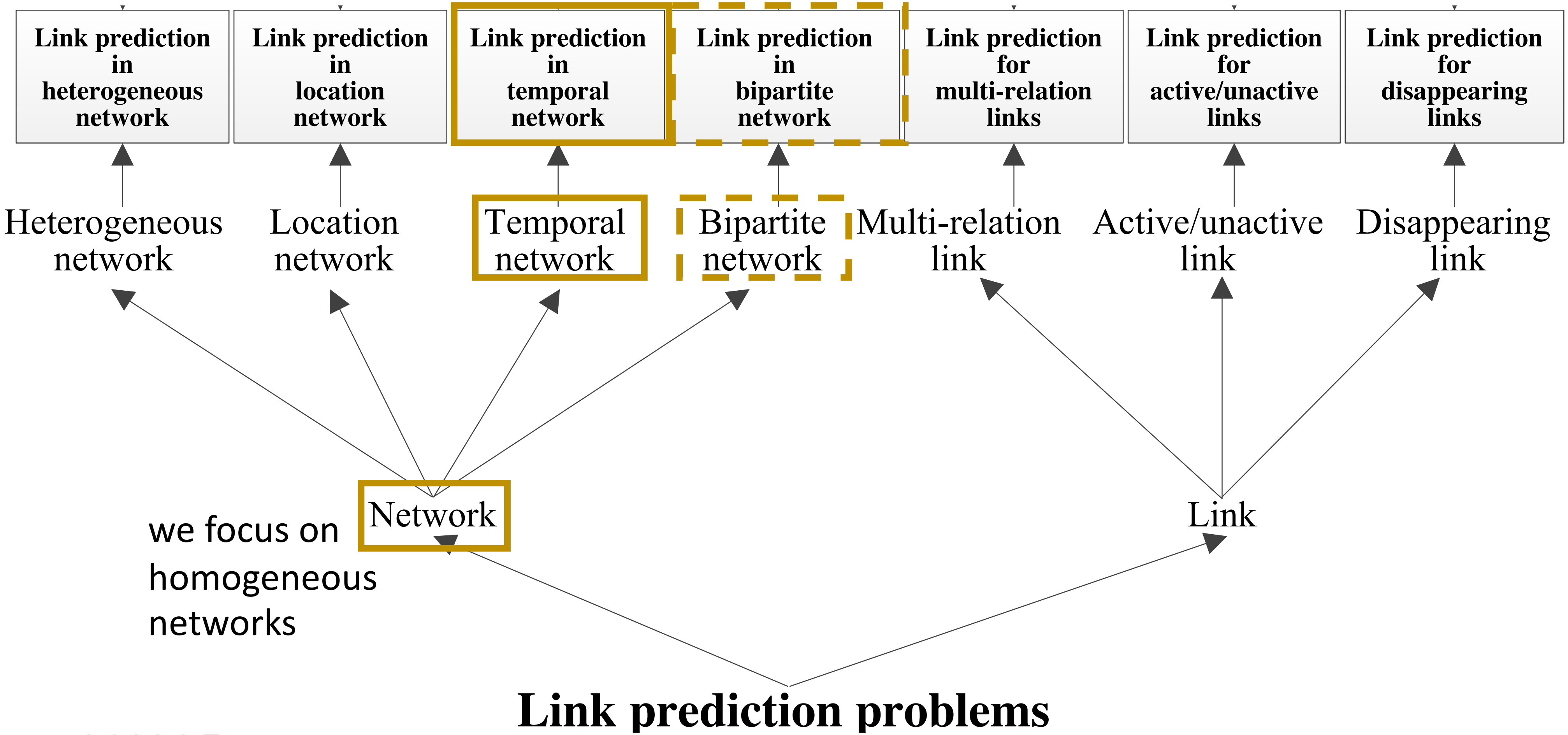
- Recommendation in **social** networks
- Finding experts and collaborations in **academic** social networks
- Reciprocal **relationships** prediction
- Network **completion** problem
- Social **tie** prediction
- …



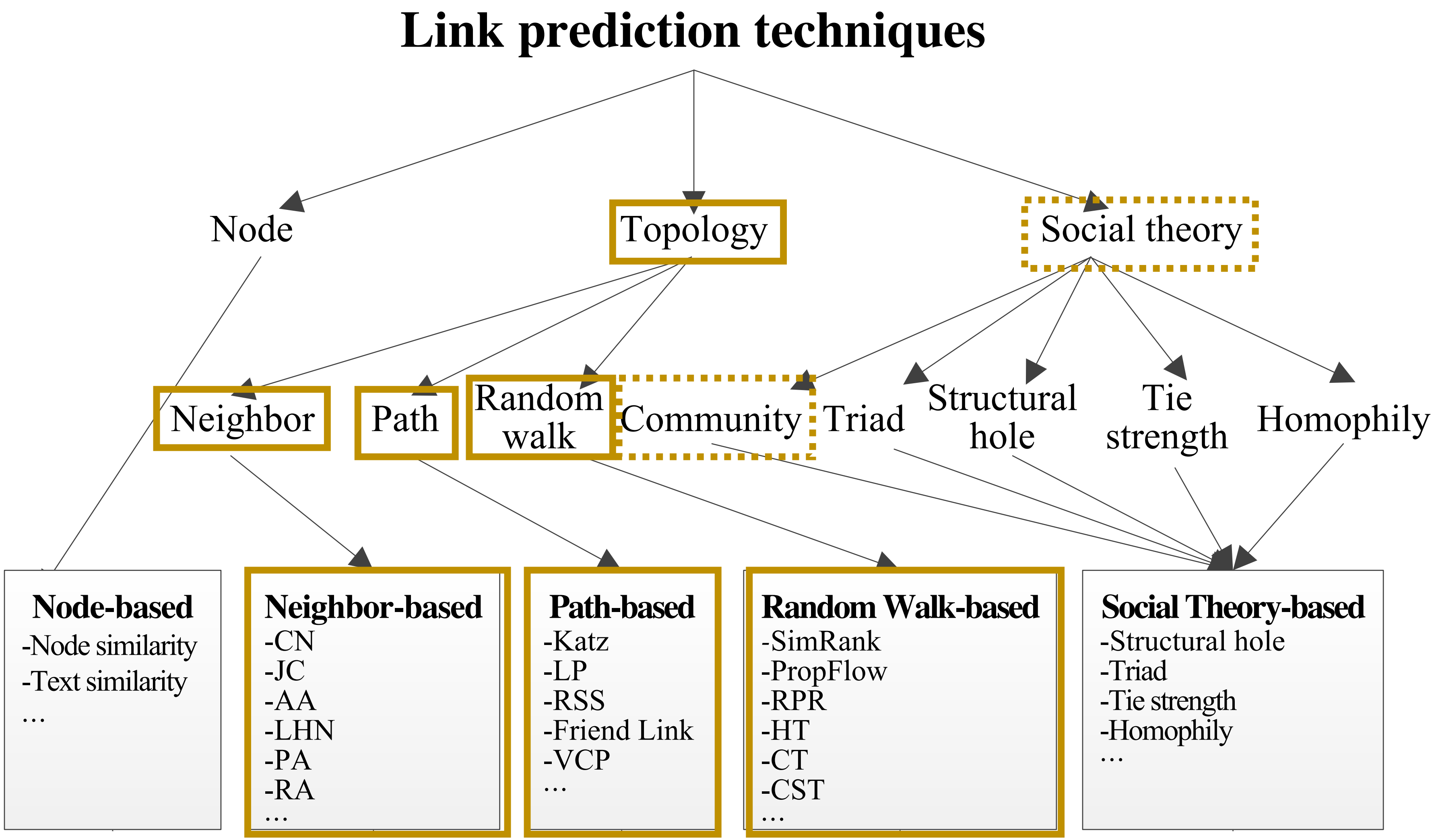
Link prediction problems

Wang, Xu, Wu, Zhou (2015) Link prediction in social networks: the state-of-the-art

https://link.springer.com/content/pdf/10.1007/s11432-014-5237-y.pdf



Link prediction techniques



Topology based techniques

Common neighbours

These **local** techniques are modification of a simple idea

Common neighbours - CN

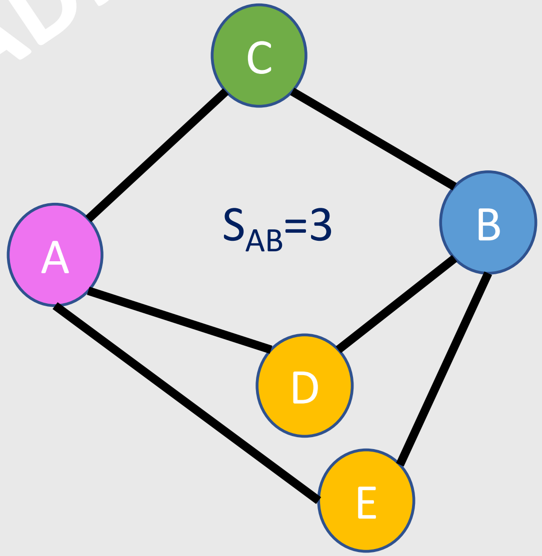
The more neighbours in common, the more likely the link to appear

intersection

$$S_{CN}(i,j) = |N_i \cap N_j|$$

(the set of) neighbours of *j*

SIMPLE TRIADIC CLOSURE



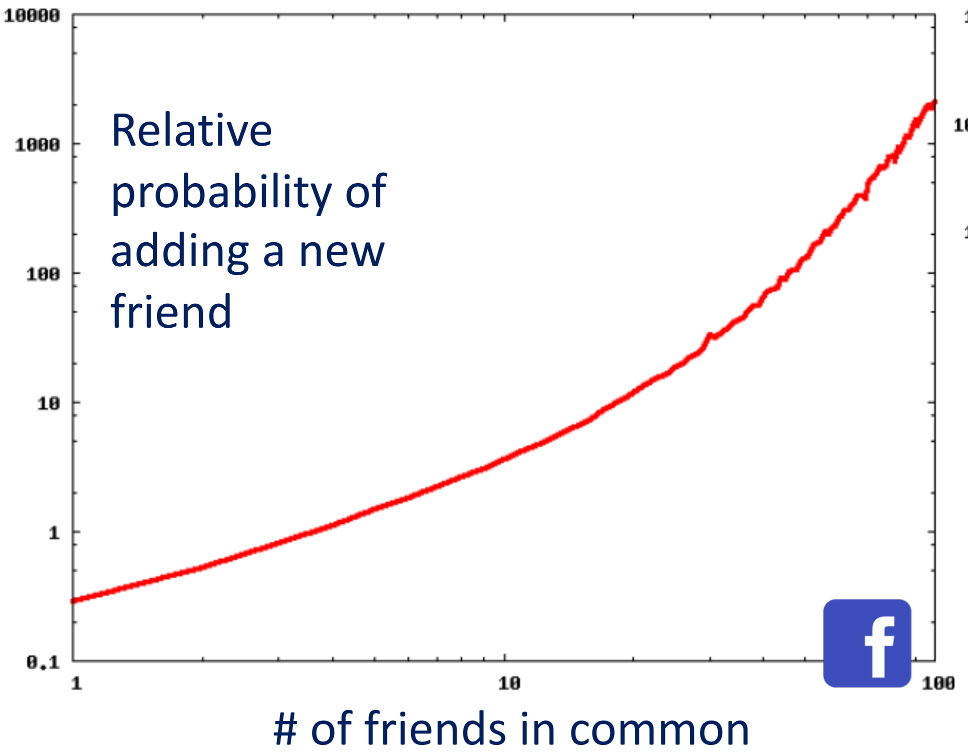
$$\mathbf{S_{CN} = A \cdot A}$$

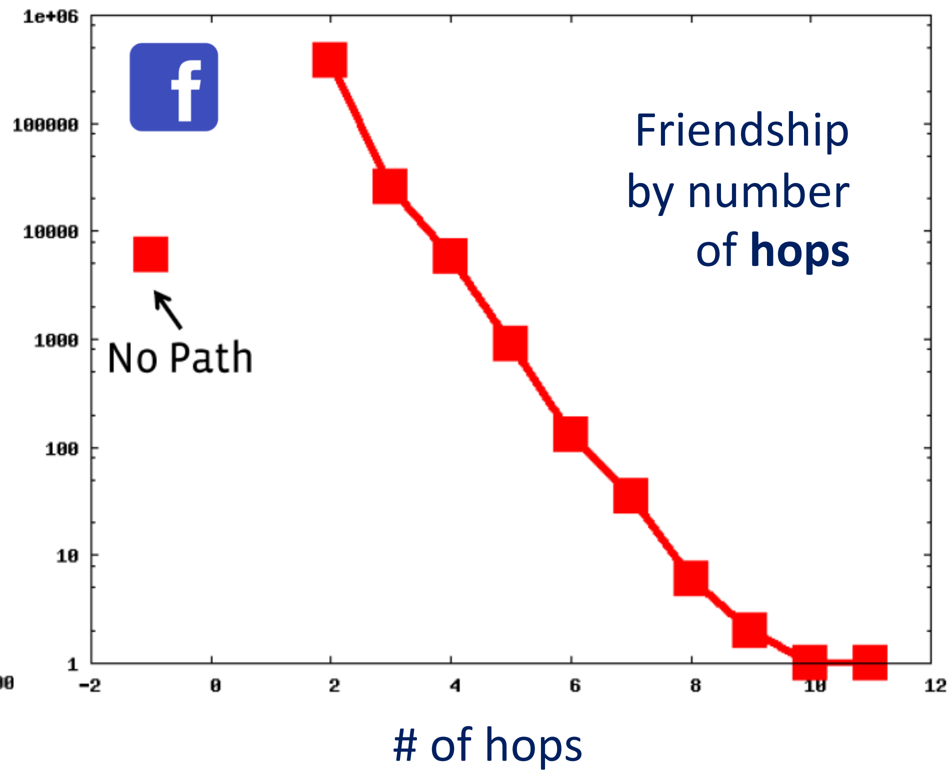
binary adjacency matrix of an undirected network

Rationale

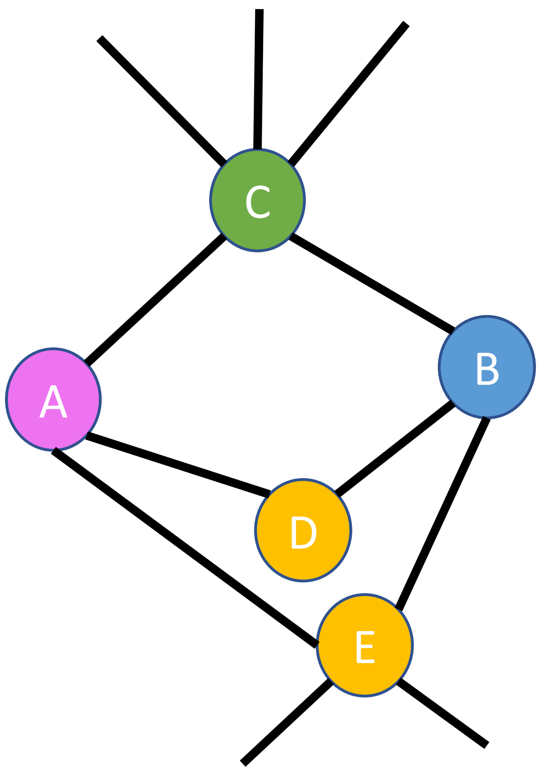
more **mutual friendships** help in becoming a friend

95% of the new friendships in facebook are **friend-of-a-friend**





Resource allocation



$S_{AB} = 1/5 + 1/2 + 1/4 = 19/20$

Resource allocation - RA

Punishes more heavily the **high-degree** common neighbours

$$S_{RA}(i,j) = \sum_{k \in N_i \cap N_j} 1 / |N_k|$$

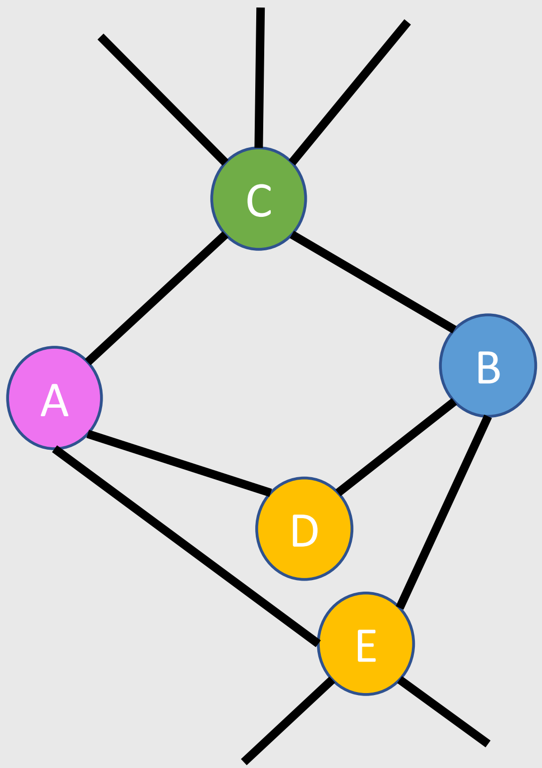
MiME

Adamic Adar

Adamic Adar - AA

Puts more **emphasis** on **less-connected** neighbours, which are more likely to make *i* and *j* meet together

$$S_{AA}(i,j) = \sum_{k \in N_i \cap N_j} 1 / \ln|N_k|$$



$S_{AB} = 1/\ln(5) + 1/\ln(2) + 1/\ln(4)$

… but very many variations exist

Path based techniques

These **global** techniques are a generalization of CN to take into account the (very many) paths of **length** $\ell \geq 2$

Kats

of paths of length ℓ between nodes *i* and *j*

$$\mathbf{S}_{\text{Katz}} = \sum_{\ell \geq 1} \beta^{\ell} \mathbf{A}^{\ell}$$

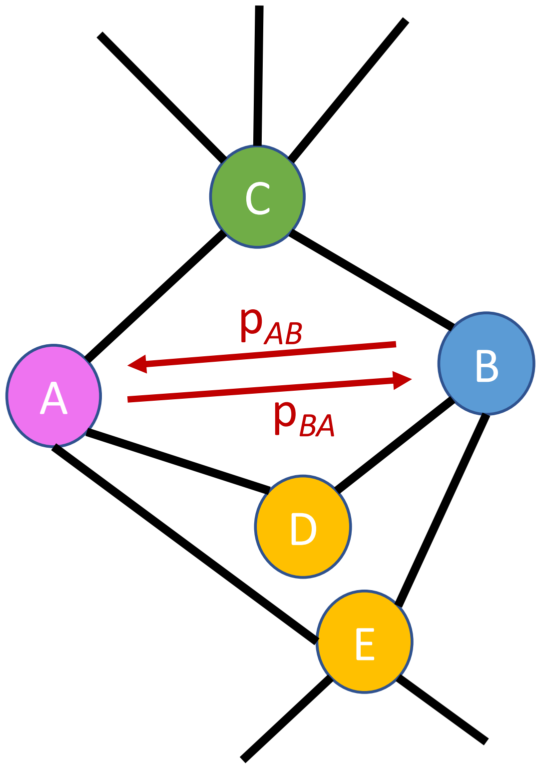
damping factor (weights more shorter paths), it needs to be sufficiently **small** $0 < \beta < 1$

Local path - LP

$$\mathbf{S}_{\text{LP}} = \mathbf{A}^2 + \beta \mathbf{A}^3$$

Random walk based techniques

Some **global** techniques exploit the Local PageRank value



random walk with restart

teleportation to node i

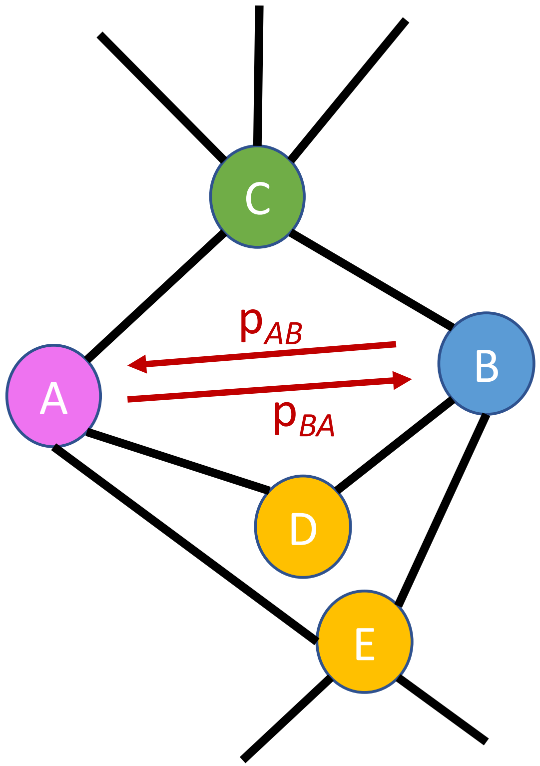
$$\mathbf{p}_i = c\,\mathbf{M}\,\mathbf{p}_i + (1-c)\,\mathbf{e}_i$$

Random walk with restart - RWR

$$S_{RWR}(i,j) = p_{ij} + p_{ji}$$

Random walk based techniques

Other exploit a pure
Random Walk



pure random walk steps

start from node *i*

$$\mathbf{p}_i(t) = \mathbf{M}^t \mathbf{e}_i$$

Local random walk - LRW

$$S_{LRW}(i,j/t) = |N_i|\, p_{ij}(t) + |N_j|\, p_{ji}(t)$$

Superposed random walk - SRW

$$S_{SRW}(i,j/t) = \sum_{u=1\ldots t} S_{LRW}(i,j/u)$$

MIME

Ingredients Networks - Pasta

Elena Camuffo, Laura Crosara, Matteo Moro

pairings		CN	AA	RA	KA	LP	RW
Nutmeg	Fresh chilli	x			x	x	
Liquid fresh cream	Carrots	x			x	x	
Tomato sauce	Pine nuts	x			x	x	
Butter	Mussels	x			x	x	
Salt	Nduja						x
Pig cheek	Pumpkin		x				
Pig cheek	Ricotta cheese	x					
Sausage	Pecorino			x			
Whole milk	Beans			x			
Whole milk	Onions golden		x		x	x	

ITALY

pairings		CN	AA	RA	KA	LP	RW
cheese	sesame	x			x	x	
macrophyll	bean			x			
salt	sweet sauce		x				x
cabbage	lemon			x			
lemon	mushrooms maitake			x			
chicken	vegetables			x			
cabbage	cheese parmigiano			x			
consomme	perilla	x			x	x	
egg	lemon	x		x	x	x	
bacon	vinegar	x			x	x	

JAPAN

pairings		CN	AA	RA	KA	LP	RW
fresh cream	chili	x		x	x	x	
black pepper	potato	x					
spices	bacon	x			x	x	
carrots	nuts		x				
canned tomatoes	pesto	x			x	x	
carrots	pesto		x				
salt	pig cheek						x
lemon juice	chicken broth		x				
rosemary	chicken broth			x			
fresh cream	sugar	x		x	x	x	

TAIWAN

MIME

Ingredients Networks - Pasta

ITALY

New Ingredient	Recipe
Black pepper	Durum wheat semolina, Water, Ricotta salata, Eggplant, Garlic, Vine-ripened tomatoes, Basil, Salt, Extra virgin olive oil
Vegetable broth	Semolina durum whole wheat, Water, Fresh onion, Mushrooms, Bacon, Cannellini beans, Rosemary, Extra virgin olive oil, Black pepper, Salt
apple	onion, anchovies, water, olive oil
Brandy	Chicken breast, Noodles, Potatoes, Snow peas, Carrots, Celery, Mushrooms, Leeks, Water, Fresh ginger, Parsley, Extra virgin olive oil, Black pepper, Salt
Almonds	streaky pork, durum wheat semolina, water, minced garlic, plum, cauliflower, mushroom, soft-boiled eggs, rice wine, salt, flour

TAIWAN

New Ingredient	Recipe
mushroom	onion, meat, red wine, concentrated tomato paste, chicken broth, bay leaves, sugar, salt, durum wheat semolina, water, cheese, fresh thyme, black pepper
chia	streaky pork, durum wheat semolina, water, minced garlic, plum, cauliflower, mushroom, soft-boiled eggs, rice wine, salt, flour
cheese	durum wheat semolina, water, bacon, asparagus, shrimp, garlic, black pepper, rose salt, paprika, parsley leaf, cheese
basil leaves	durum wheat semolina, water, onion, cream, chicken breast, squid
avocado	durum wheat semolina, water, bacon, large tomatoes, green pepper, mushroom, cheese, ketchup, salt, black pepper

JAPAN

New Ingredient	Recipe
consomme	durum wheat semolina, water, salmon, olives oil
tomato	onion, bacon, garlic, olives oil, cream, salt, cheese, durum wheat semolina, water, juice, nut
soy sauce	chicken, salt, durum wheat semolina, water, avocado, clams, mayonnaise, onion, cod roe
onion	durum wheat semolina, water, saury, salt
pepper	durum wheat semolina, water, salmon, olives oil

Performance comparison

Lü, Zhou, “Link prediction in complex networks: A survey,” 2011

https://www.sciencedirect.com/science/article/pii/S037843711000991X

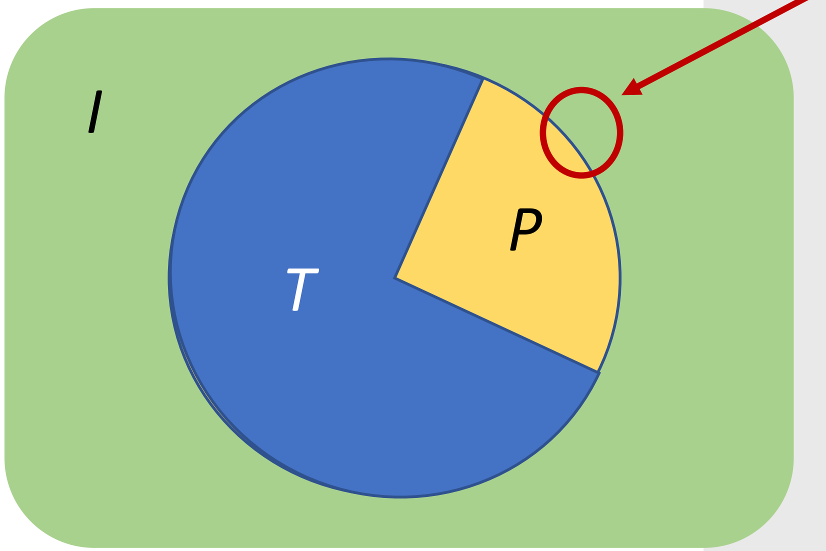
Precision

Start from a friendship network with **active edges** *E*, and divide them into:

- a **probe** set *P* (a small subset of it)
- a **test** set *T* (the remaining edges)

Build the **similarity** values, S, by exploiting the test set *T*

Denote the **inactive** edges set with *I*



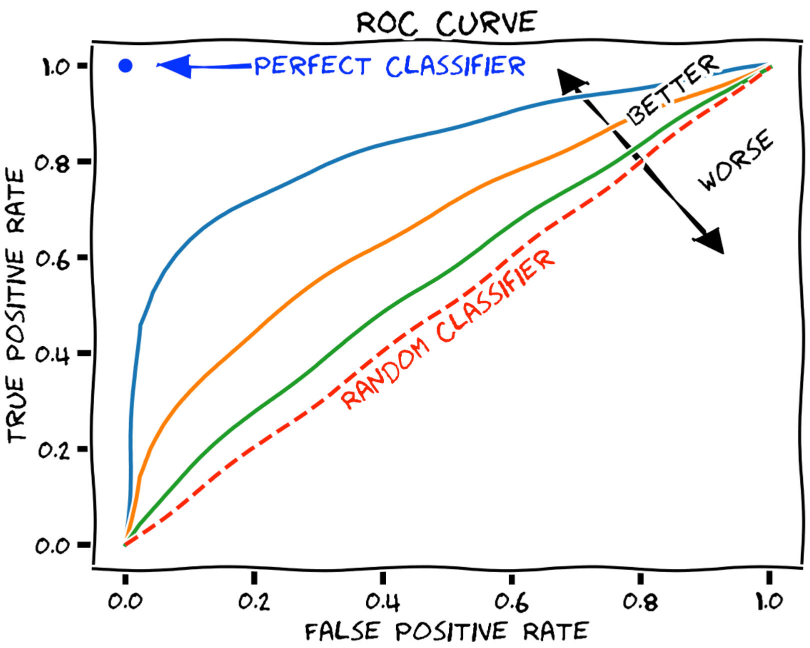
Precision

Percentage of the **top L links**, ranked according to the similarity measure S, that belong to the probe set P

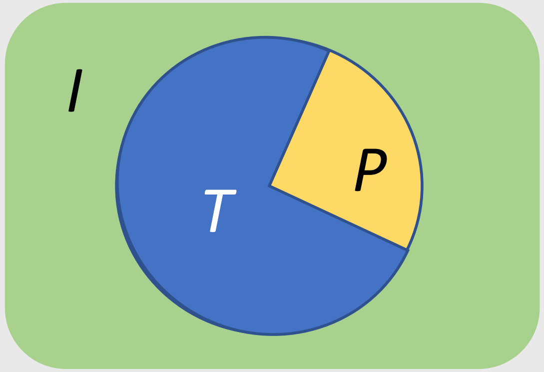
AUC = Area under the ROC curve

receiver operating characteristic

P(S_{Probe} > threshold)



P($S_{Inactive}$ > threshold)



I = inactive, P = probe, T = test

$$A = \int_{x=0}^{1} \mathrm{TPR}(\mathrm{FPR}^{-1}(x))\, dx$$

MIME.

AUC explained

Area under the curve

When using normalized units, the area under the curve (often referred to as simply the AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative').

$$\mathrm{TPR}(T) = \int_T^{\infty} f_P(x)\, dx$$

$$\mathrm{FPR}(T) = \int_T^{\infty} f_I(x)\, dx$$

PDFs of P and I values

$$A = \int_0^1 \mathrm{TPR}(\mathrm{FPR}^{-1}(x))\, dx = \int_{-\infty}^{+\infty} \mathrm{TPR}(T)\mathrm{FPR}'(T)\, dT$$

$$= \int_{-\infty}^{+\infty} \int_T^{\infty} f_P(x) f_I(T)\, dx\, dT$$

$$= \int_{-\infty}^{+\infty} \int_{-\infty}^{\infty} \eta(x > T) f_P(x) f_I(T)\, dx\, dT$$

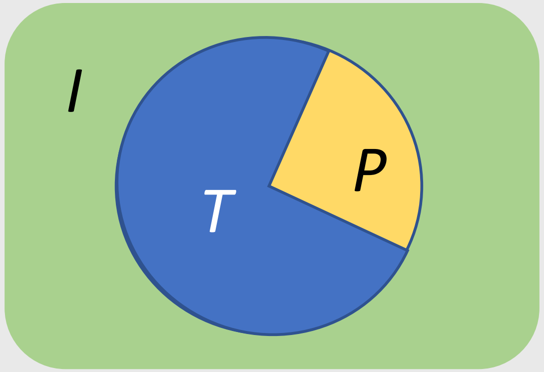
$$= P(S_P > S_I)$$

AUC expression

Derive the probability that **similarity is larger in *P* than in *I*,** i.e., the probability that a correct estimate is obtained

$$AUC = \sum_{p \in P,\, i \in I} \frac{(S(p)>S(i)?1:0)}{|P|\,|I|}$$

AUC = 1 corresponds to a **perfect** classifier



I = inactive, P = probe, T = test

Performance - Neighbour

protein-protein interaction network

co-authorships in network science

power grid

US political blogs

router-level Internet

air transport. system

Indices	PPI	NS	Grid	PB	INT	USAir
CN	0.889	**0.933**	**0.590**	0.925	**0.559**	0.937
Salton	0.869	0.911	0.585	0.874	0.552	0.898
Jaccard	0.888	**0.933**	**0.590**	0.882	**0.559**	0.901
Sørensen	0.888	**0.933**	**0.590**	0.881	**0.559**	0.902
HPI	0.868	0.911	0.585	0.852	0.552	0.857
HDI	0.888	**0.933**	**0.590**	0.877	**0.559**	0.895
LHN1	0.866	0.911	0.585	0.772	0.552	0.758
PA	0.828	0.623	0.446	0.907	0.464	0.886
AA	0.888	0.932	**0.590**	0.922	**0.559**	0.925
RA	**0.890**	**0.933**	**0.590**	**0.931**	**0.559**	**0.955**

Resource allocation

- 90% of edges in the test set *T*
- test set chosen at random
- average over 10 tests

Performance - Path

protein-protein interaction network | co-authorships in network science | power grid | US political blogs | router-level Internet | air transport system

AUC	PPI	NS	Grid	PB	INT	USAir
LP	0.970	**0.988**	0.697	**0.941**	0.943	**0.960**
LP*	0.970	**0.988**	0.697	0.939	0.941	0.959
Katz	**0.972**	**0.988**	**0.952**	0.936	**0.975**	0.956
LHN2	0.968	0.986	0.947	0.769	0.959	0.778

Precision	PPI	NS	Grid	PB	INT	USAir
LP	**0.734**	**0.292**	**0.132**	**0.519**	**0.557**	**0.627**
LP*	**0.734**	**0.292**	**0.132**	0.469	0.121	**0.627**
Katz	0.719	0.290	0.063	0.456	0.368	0.623
LHN2	0	0.060	0.005	0	0	0.005

… but very low precision values !!!!

Kats for AUC

LP for precision

- 90% of edges in the test set *T*
- test set chosen at random
- L=100 for precision

Performance – Random walk

AUC	CN	RA	LP	ACT	RWR	HSM	LRW	SRW
USAir	0.954	0.972	0.952	0.901	0.977	0.904	0.972(2)	**0.978**(3)
NetScience	0.978	0.983	0.986	0.934	**0.993**	0.930	0.989(4)	0.992(3)
Power	0.626	0.626	0.697	0.895	0.760	0.503	0.953(16)	**0.963**(16)
Yeast	0.915	0.916	0.970	0.900	0.978	0.672	0.974(7)	**0.980**(8)
C.elegans	0.849	0.871	0.867	0.747	0.889	0.808	0.899(3)	**0.906**(3)
Precision	CN	RA	LP	ACT	RWR	HSM	LRW	SRW
USAir	0.59	0.64	0.61	0.49	0.65	0.28	0.64(3)	**0.67**(3)
NetScience	0.26	0.54	0.30	0.19	**0.55**	0.25	0.54(2)	0.54(2)
Power	0.11	0.08	**0.13**	0.08	0.09	0.00	0.08(2)	0.11(3)
Yeast	0.67	0.49	0.68	0.57	0.52	0.84	**0.86**(3)	0.73(9)
C.elegans	0.12	0.13	**0.14**	0.07	0.13	0.08	**0.14**(3)	**0.14**(3)

... but simple RA/LP methods
still behave very well !

Random walk
methods

Adding a learning technique

Backstrom, Lescovec, “Supervised random walks: predicting and recommending links in social networks,” 2011

https://dl.acm.org/doi/pdf/10.1145/1935826.1935914

Supervised random walk - SRW

idea

In random walk with restart, we add **fractional weights** to the adjacency matrix **A**, and optimize them in order to find the best **fit** to the existent, i.e., we require S(*I*) < S(*P*)

model

$$a_{ij} = \frac{1}{1 + e^{-\langle \beta , \psi_{ij} \rangle}}$$

parameters vector, to be estimated via a best fit

features vector, i.e., things we know about link *i*-*j*: when it was created, # of exchanged messages, # photos *i* and *j* appeared in, etc.

MIME.

Facebook Island 2009 example

Features:

node

- age, gender, degree

edge

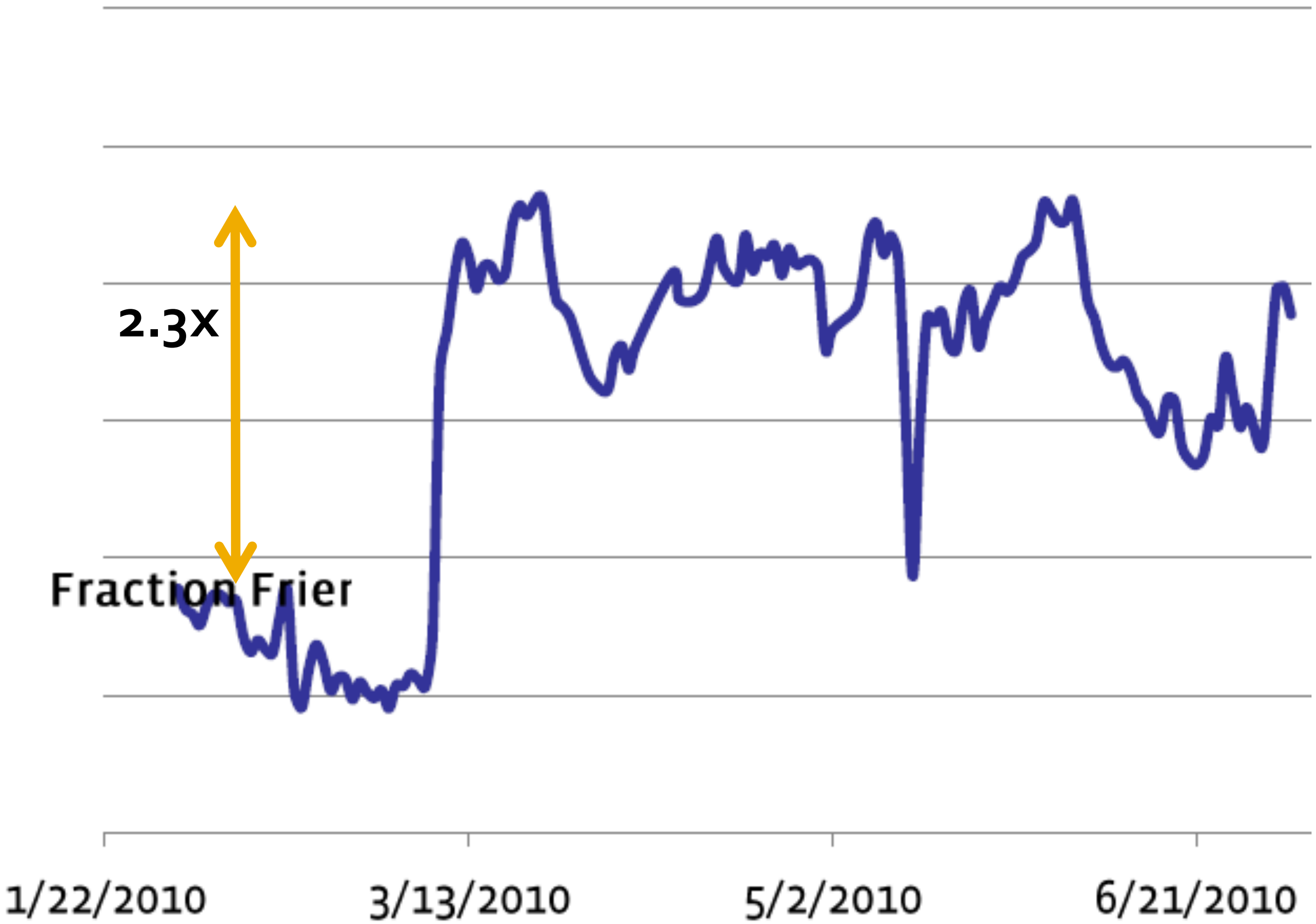
- age of an edge
- communication
- profile visits
- co-tagged photos

Learning Method	AUC	Prec@20
Random Walk with Restart	0.81725	6.80
Degree	0.58535	3.25
DT: Node features	0.59248	2.38
DT: Path features	0.62836	2.46
DT: All features	0.72986	5.34
LR: Node features	0.54134	1.38
LR: Path features	0.51418	0.74
LR: All features	0.81681	7.52
SRW: one edge type	**0.82502**	6.87
SRW: multiple edge types	**0.82799**	**7.57**

decision tree

logistic regression

Fraction of friending from PYMK



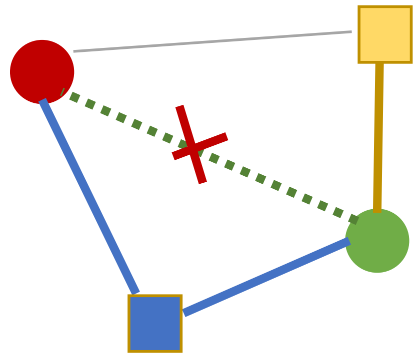
Bipartite graphs

Daminelli, Thomas, Duràn, Cannistraci, “Common neighbours and the local-community-paradigm ... bipartite networks,” 2015

https://iopscience.iop.org/article/10.1088/1367-2630/17/11/113037/pdf

Problem

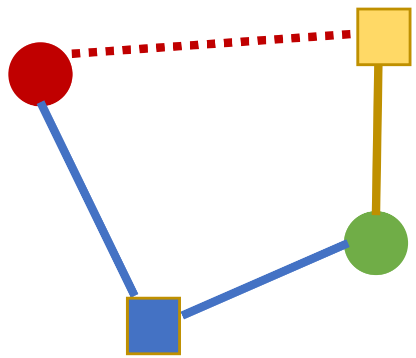
In **bipartite** graphs nodes sharing a common neighbour cannot be linked



in bipartite networks a friend-of-a-friend **cannot** be a friend

idea

extend the idea of **2-hops neighbour** into that of **3-hops neighbour**



a **three-hop** connection is required

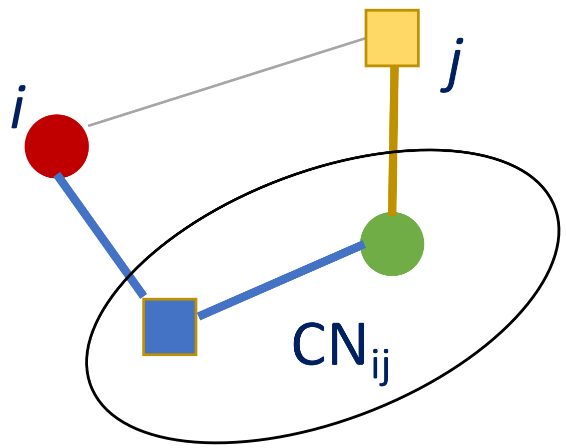
Common neighbours set

Identify nodes in all 3-hops connections between $\boxed{i}$ and $\boxed{j}$ with

circles

squares

$$CN_{i,j} = \boxed{\{N_{Ni} \cap N_j\}} \cup \boxed{\{N_i \cap N_{Nj}\}}$$



Common neighbours - CN

$$S_{CN}(i,j) = |CN_{i,j}|$$

Adamic Adar - AA

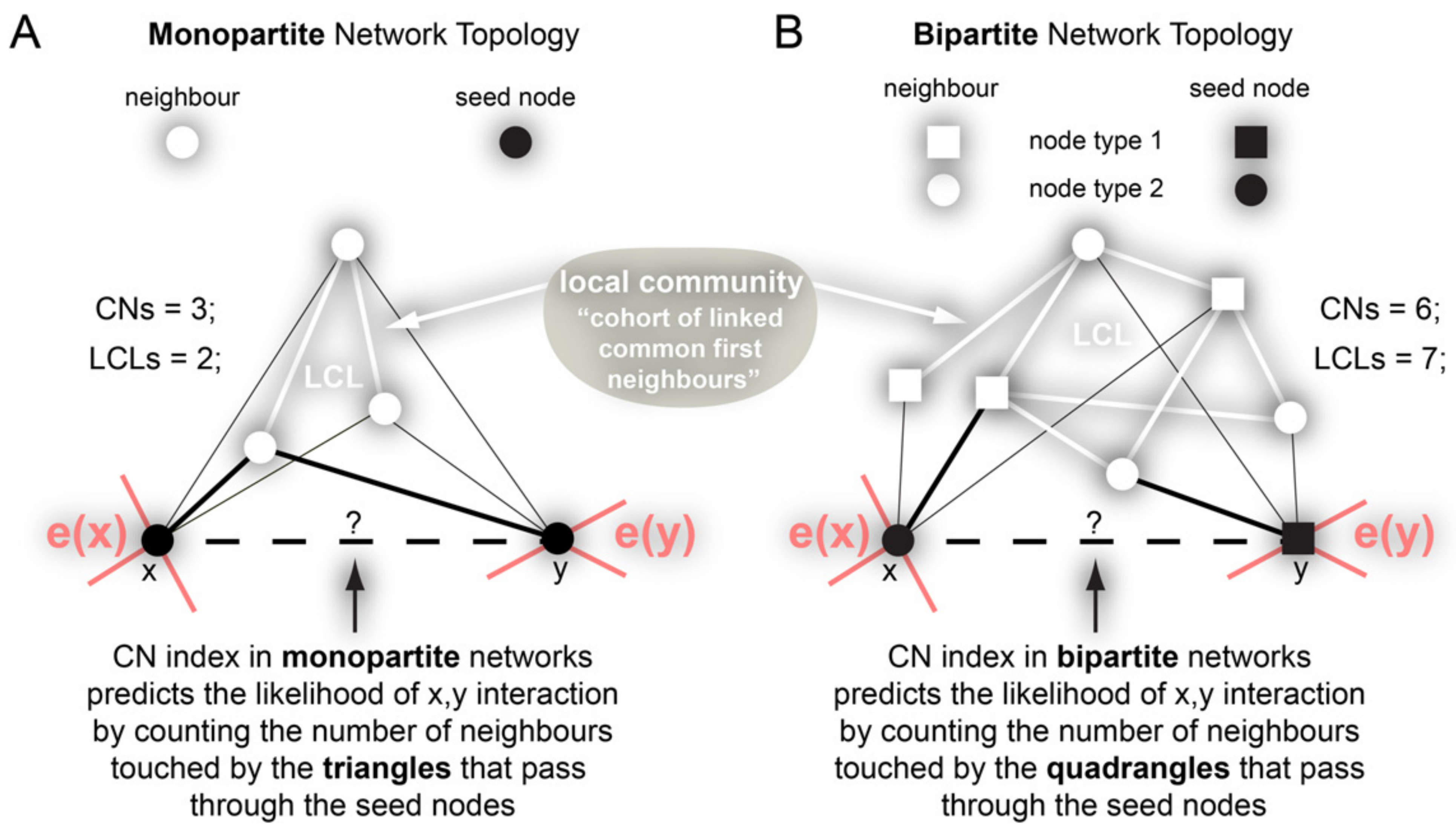
$$S_{AA}(i,j) = \sum_{k \in CN_{i,j}} 1 / \ln|N_k|$$

Resource allocation - RA

$$S_{RA}(i,j) = \sum_{k \in CN_{i,j}} 1 / |N_k|$$

MIME

Local community



Local community degree

For each node k in the local community $CN_{i,j}$ identify the number of **neighbours** of k that belong to the **community**

$$g(k) = |CN_{i,j} \cap N_k|$$

i.e., the # of local community links

Common neighbours - CAR

$$S_{CAR}(i,j) = \sum_{k \in CN_{i,j}} g(k)$$

Adamic Adar - CAA

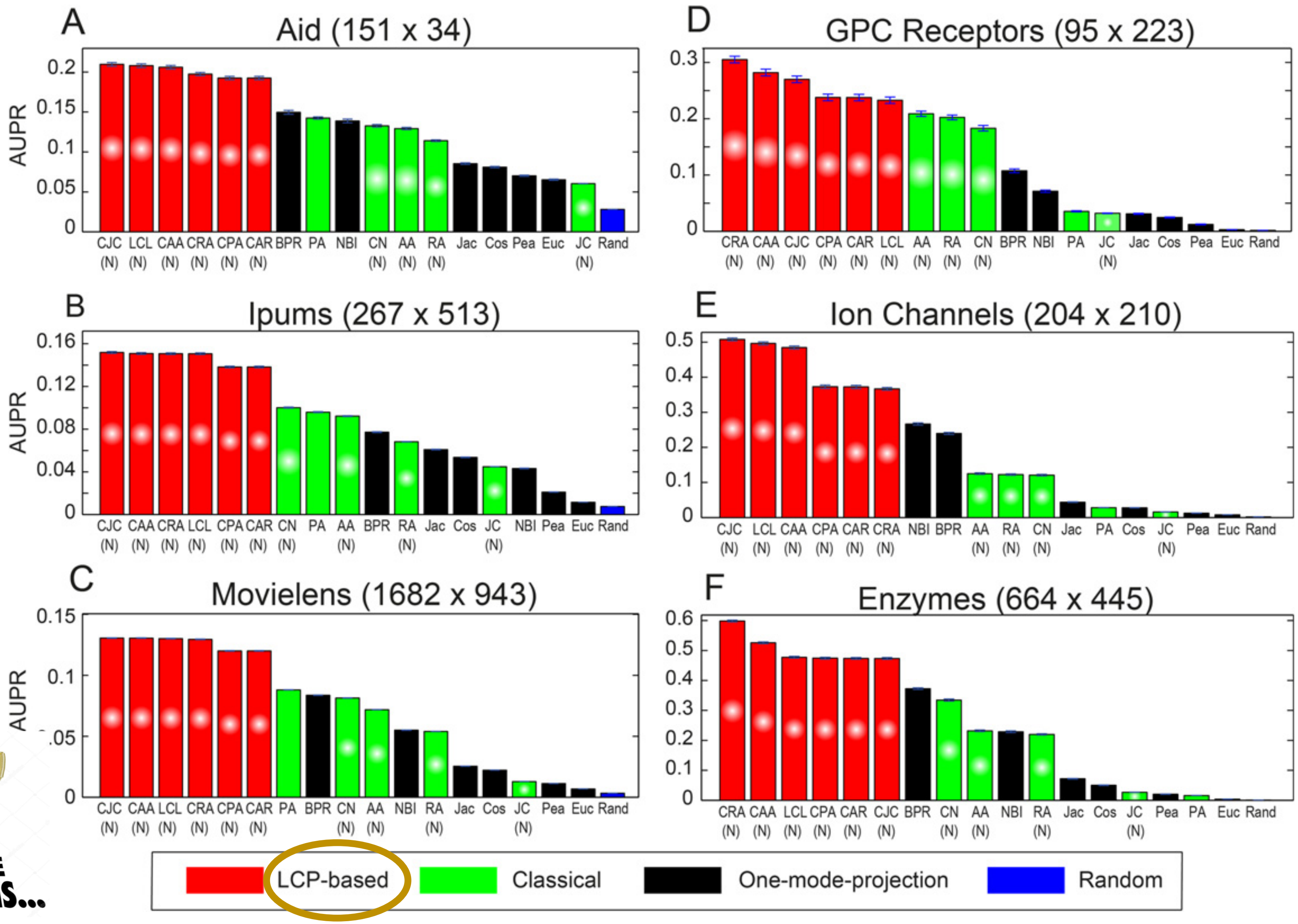
$$S_{CAA}(i,j) = \sum_{k \in CN_{i,j}} g(k) / \ln|N_k|$$

Resource allocation - CRA

$$S_{CRA}(i,j) = \sum_{k \in CN_{i,j}} g(k) / |N_k|$$

MIME.

Performance



Questions ?