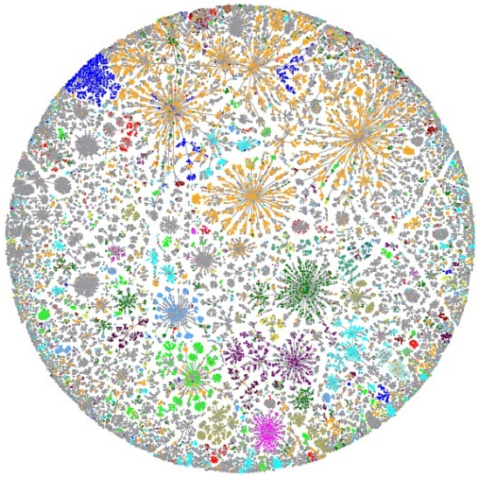
Mining Rich Graphs
Ranking, Classification, and Anomaly Detection

Leman Akoglu

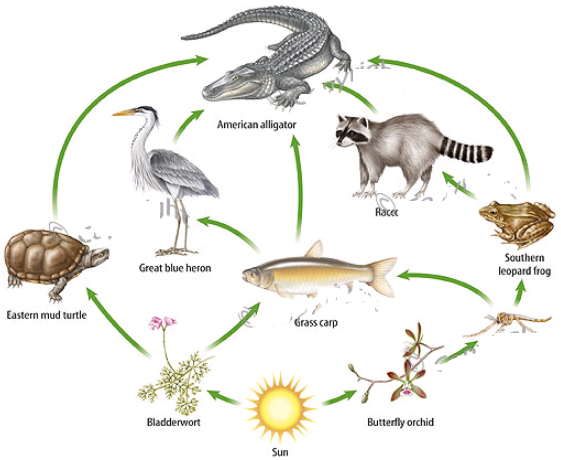
Feb 9th 2018

Carnegie Mellon University
Heinz College

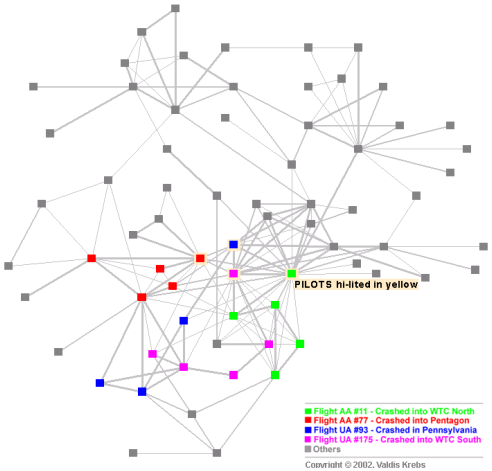
Networks are ubiquitous!



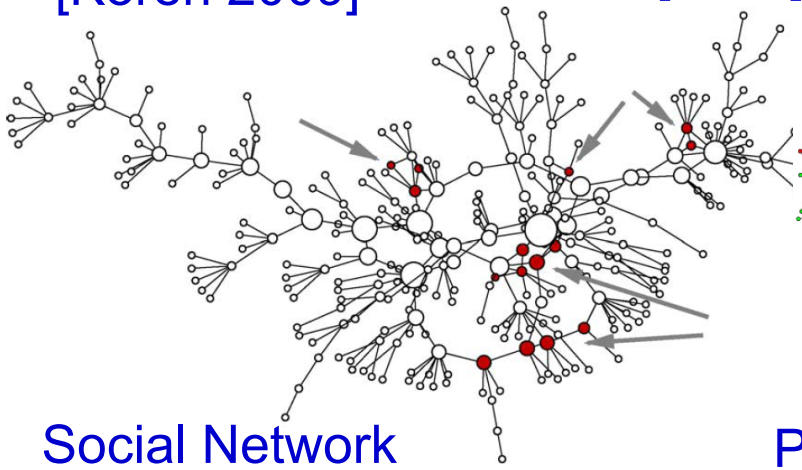
Internet Map
[Koren 2009]



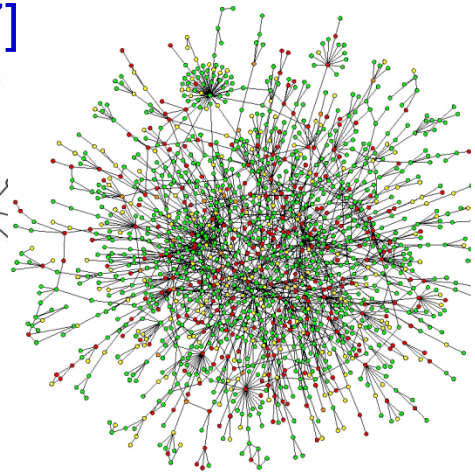
Food Web
[2007]



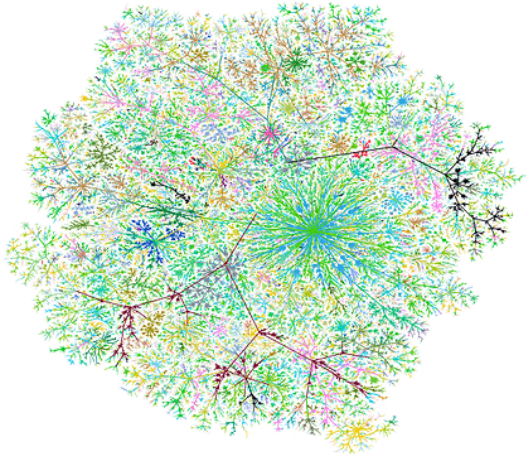
Terrorist Network
[Krebs 2002]



Social Network
[Newman 2005]

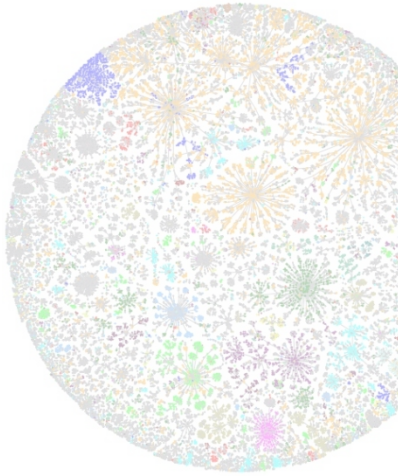


Protein Network
[Salthe 2004]



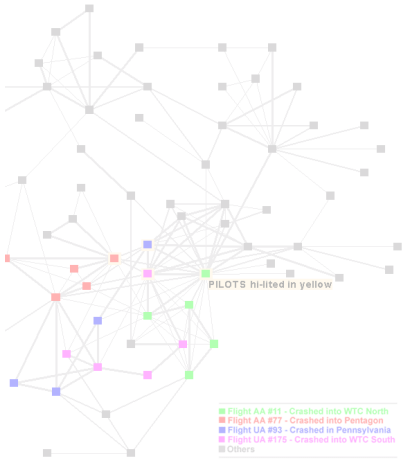
Web Graph

Graph problems

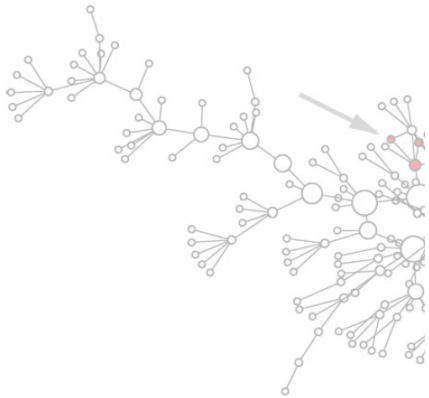


Internet Map
[Koren 2009]

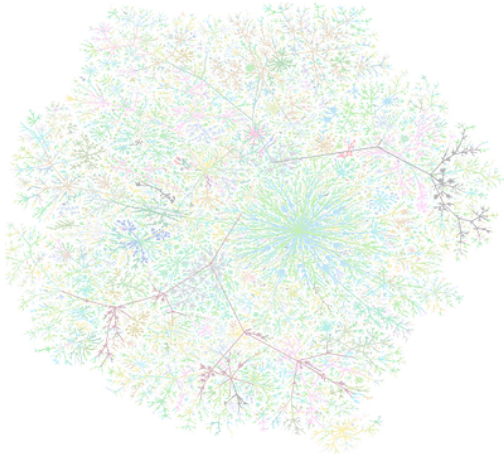
- ranking,
- classification,
- clustering & anomaly mining,
- link prediction,
- role discovery,
- similarity search,
- influence,
- evolution,
- ...



rrorist Network
Krebs 2002]

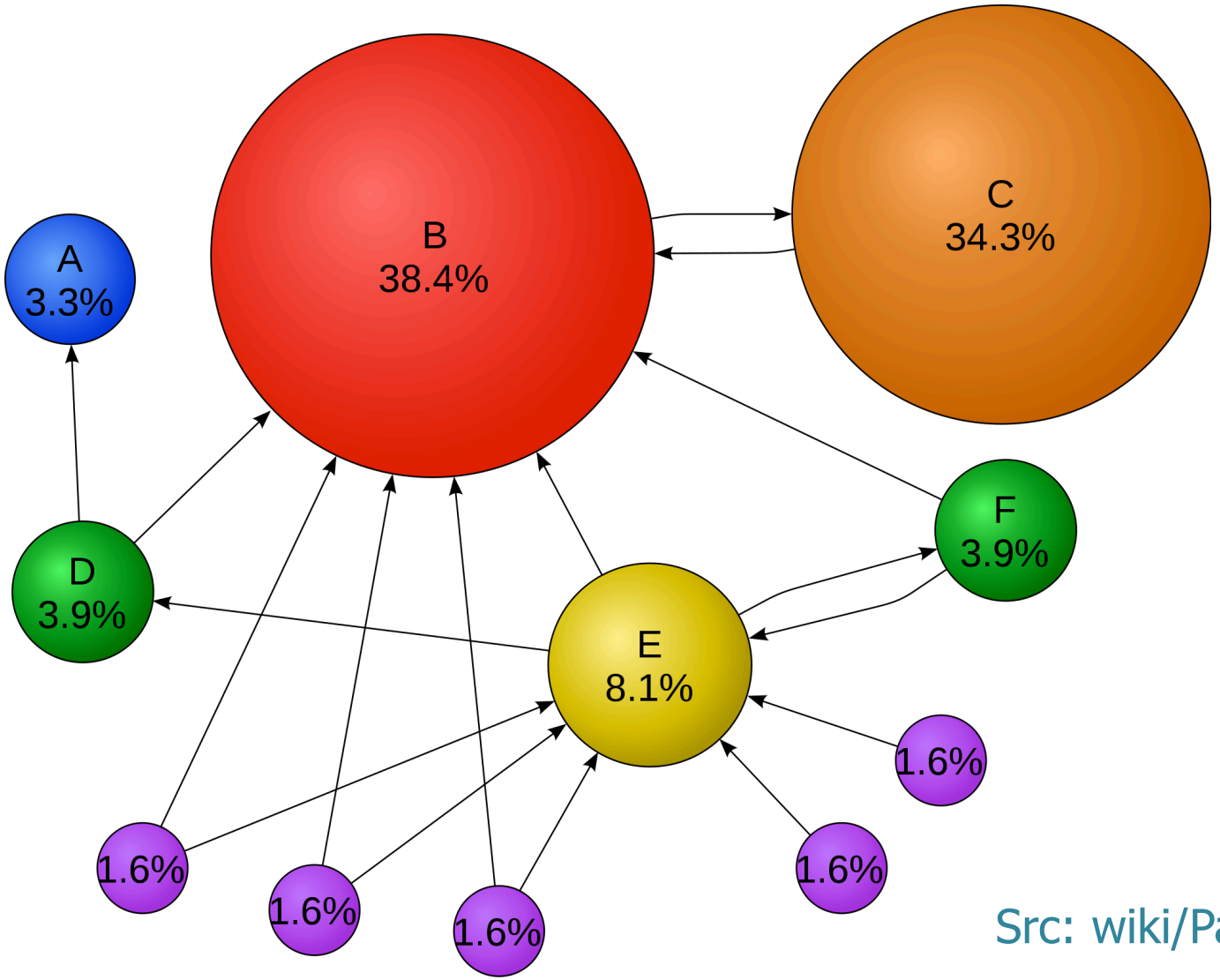


Social Network
[Newman 2005]



Web Graph

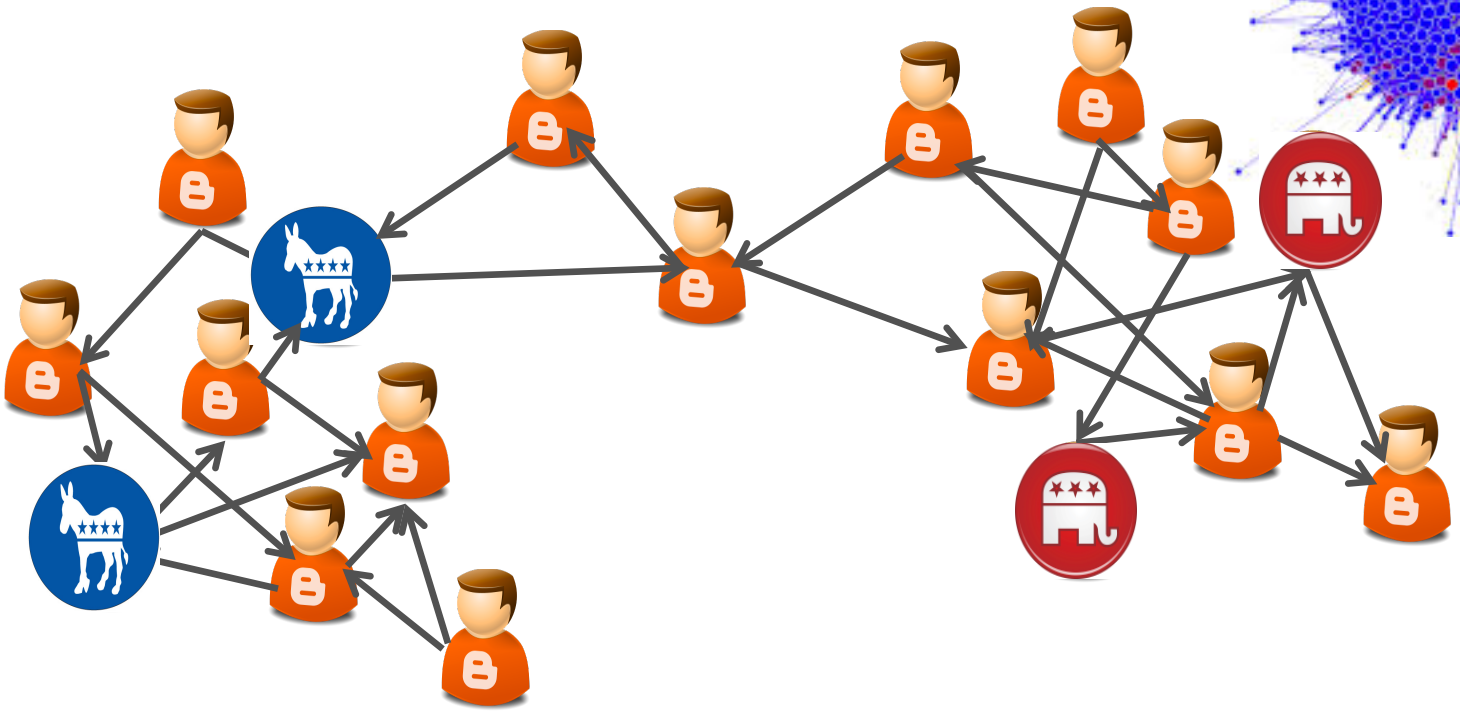
Ranking in networks



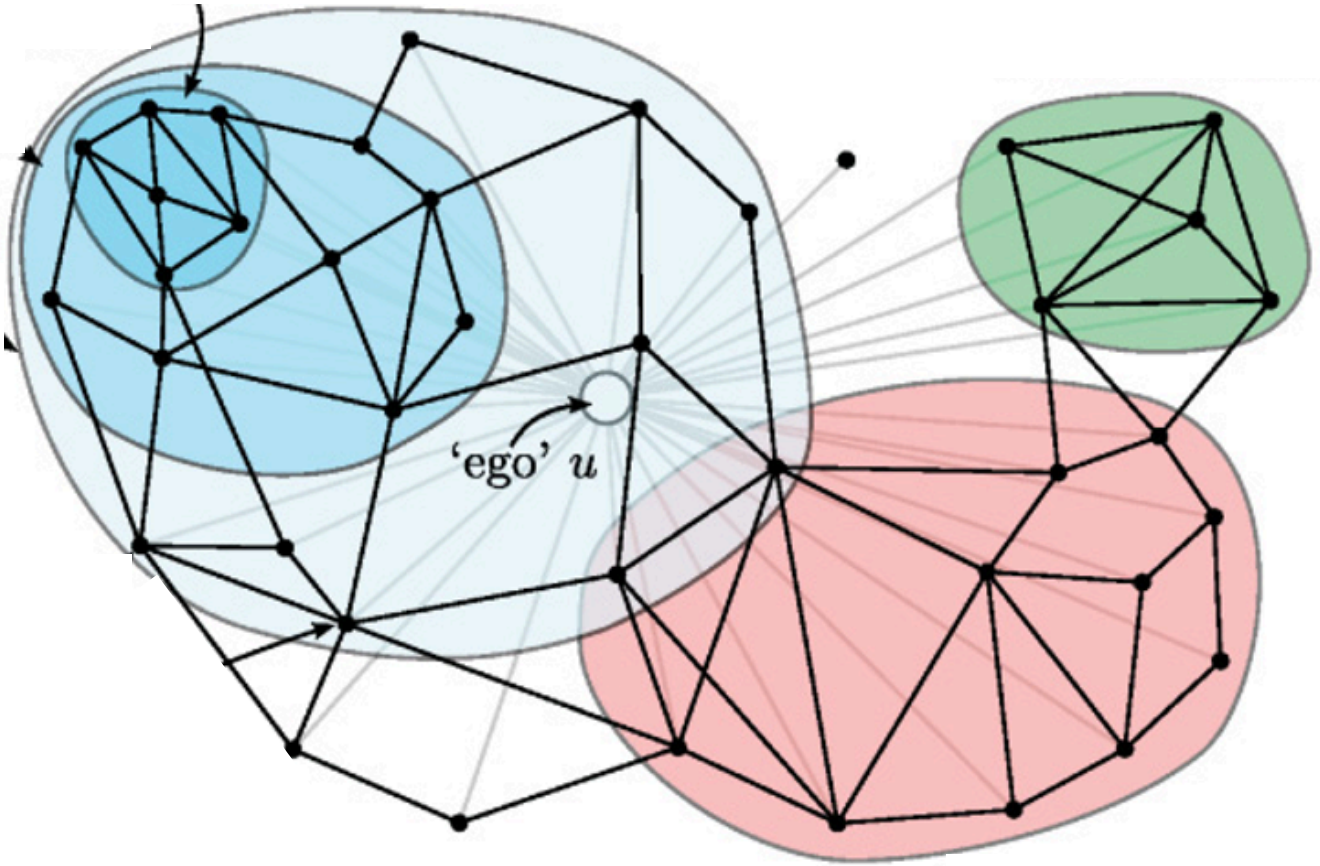
Src: wiki/PageRank

Classification in networks

Src: [Adamic+ 2005]

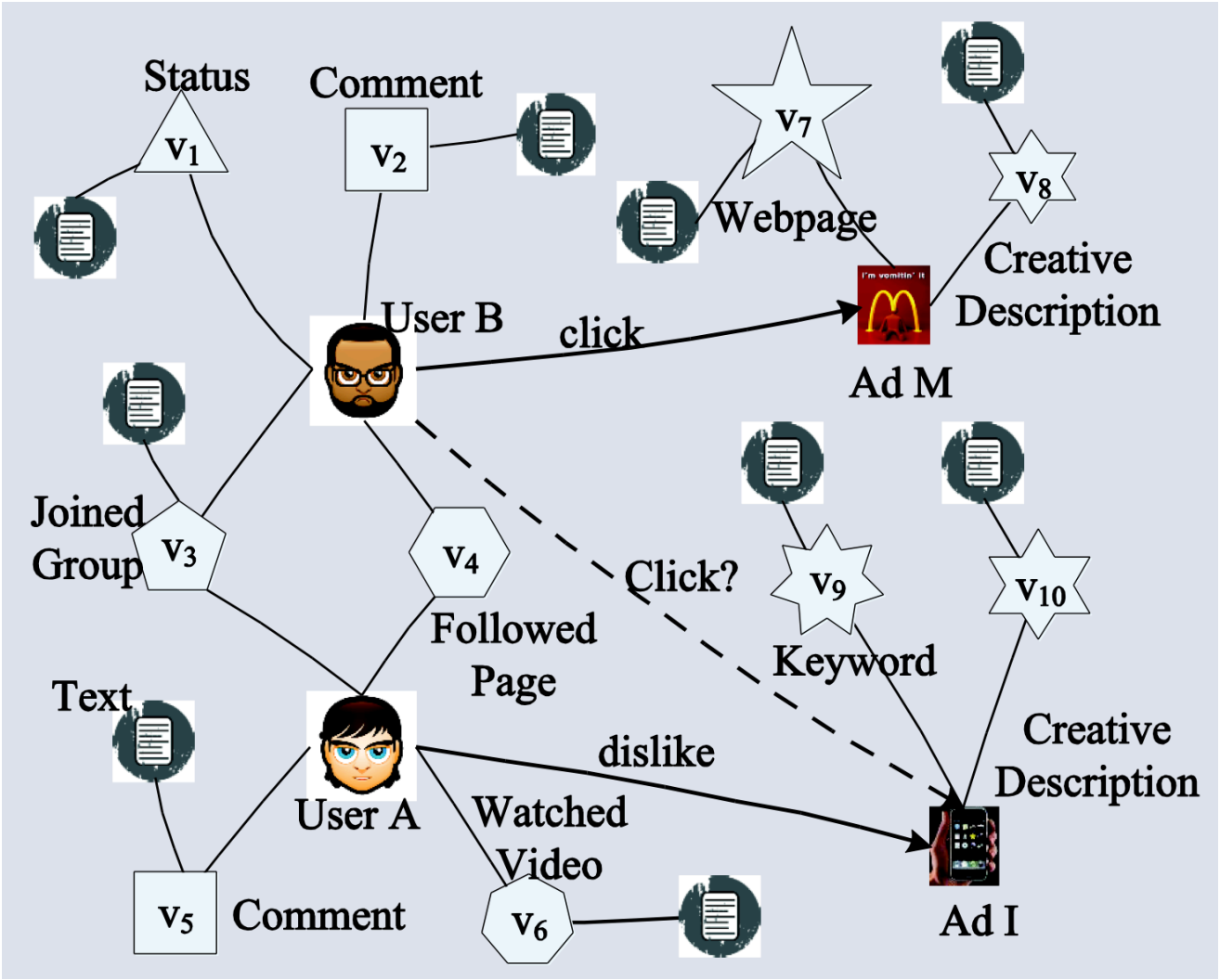


Community detection in networks



Src: [McAuley&Leckovec 2012]

Rich networks

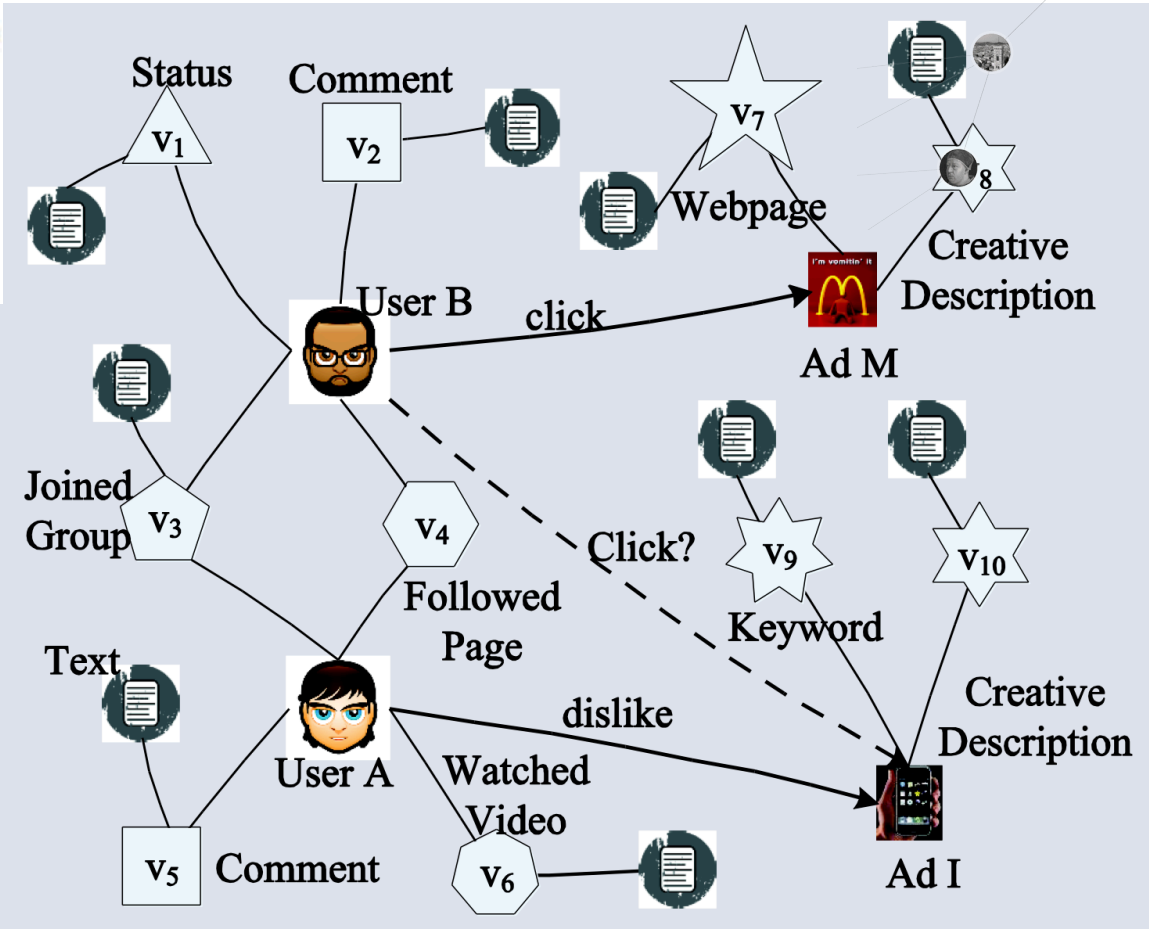


Rich networks also ubiquitous!

twitter

WIKIPEDIA

DBpedia



Google

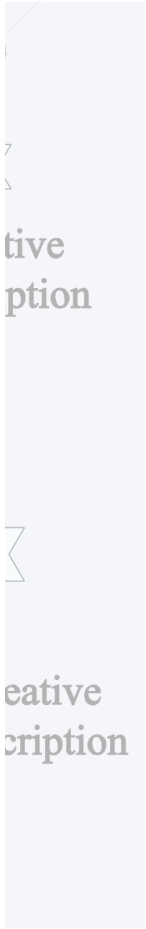
Freebase

facebook

Read the Web

Graph problems on rich networks

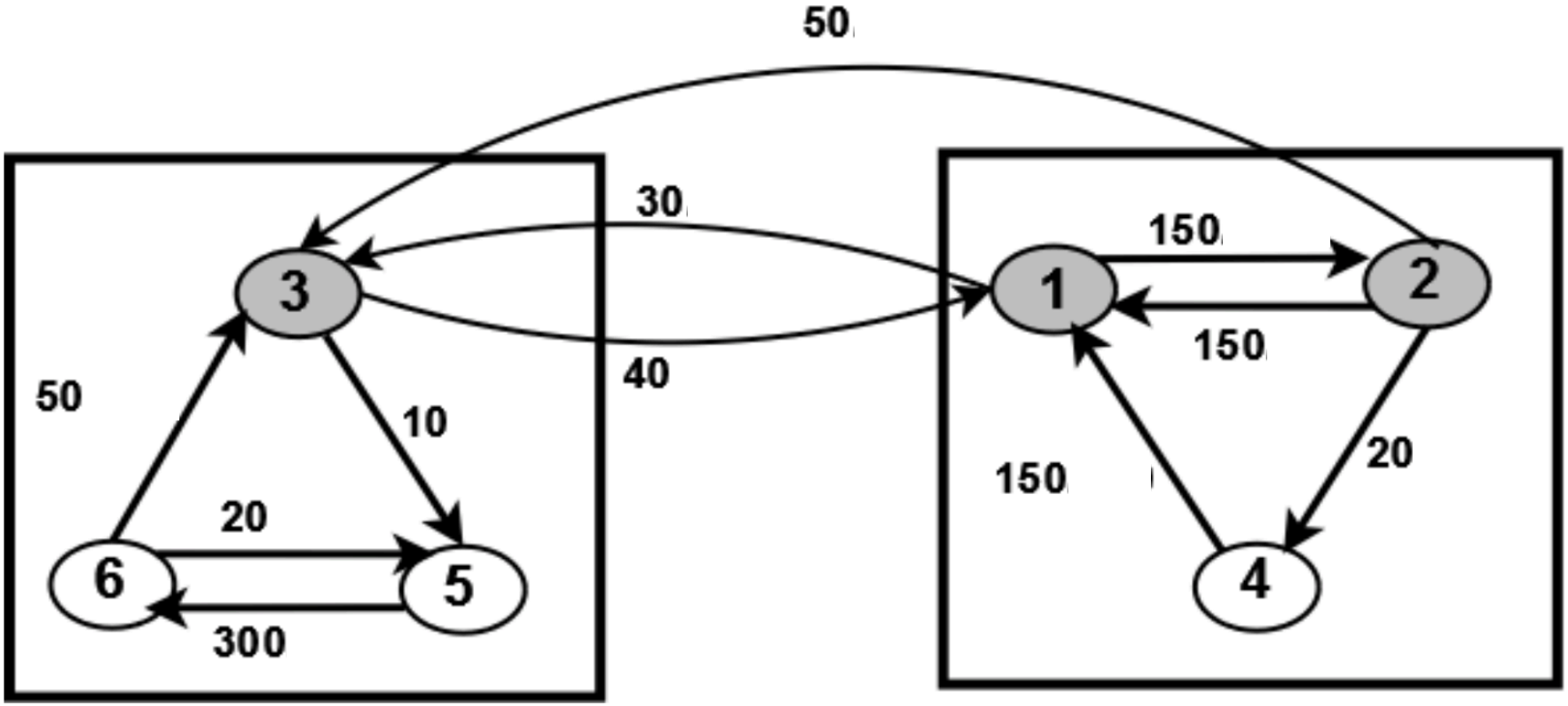
- ranking,
- clustering & anomaly mining,
- classification,
- link prediction,
- role discovery,
- similarity search,
- influence,
- evolution,
- …



Graph problems on rich networks

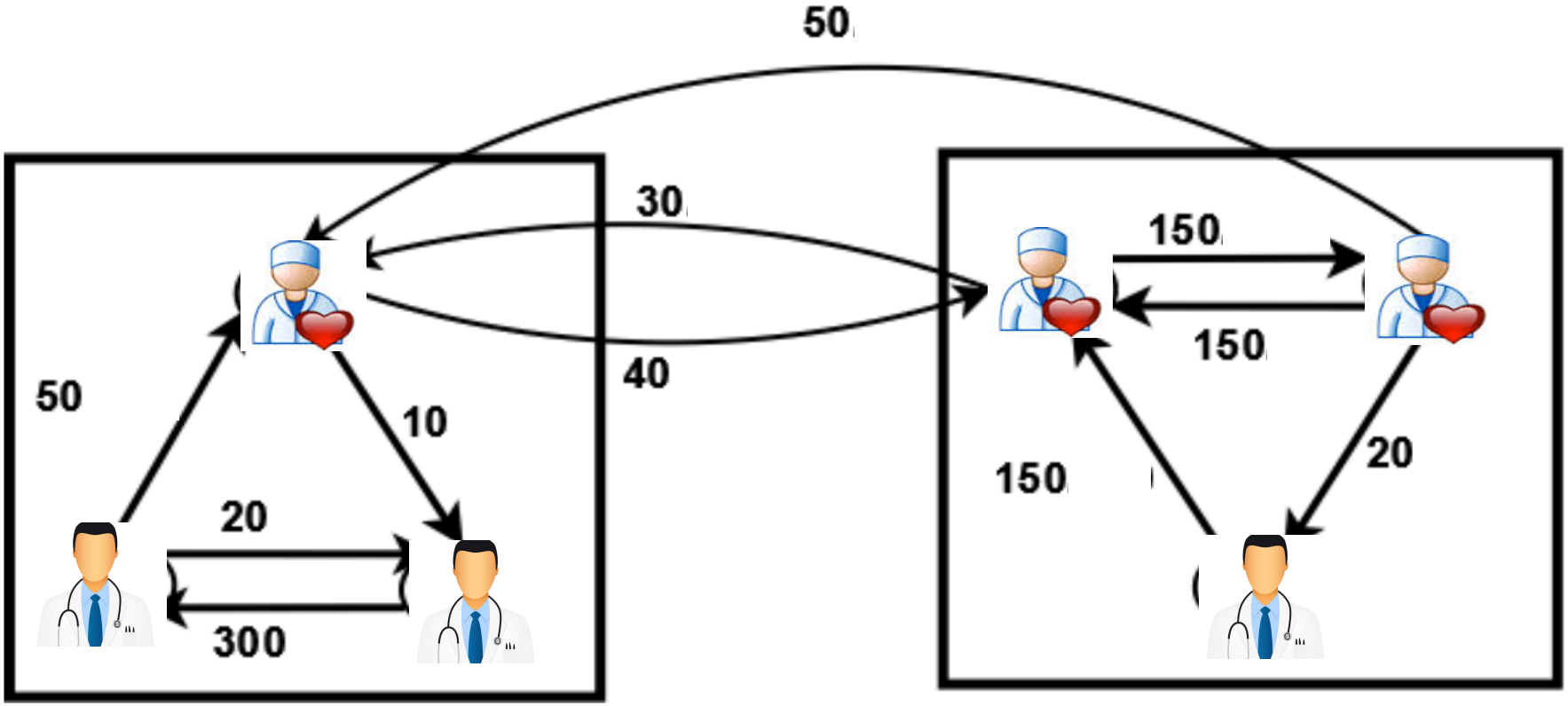
- ranking,
- clustering & anomaly mining,
- classification,
- link prediction,
- role discovery,
- similarity search,
- influence,
- evolution,
- …

Ranking in rich networks: example



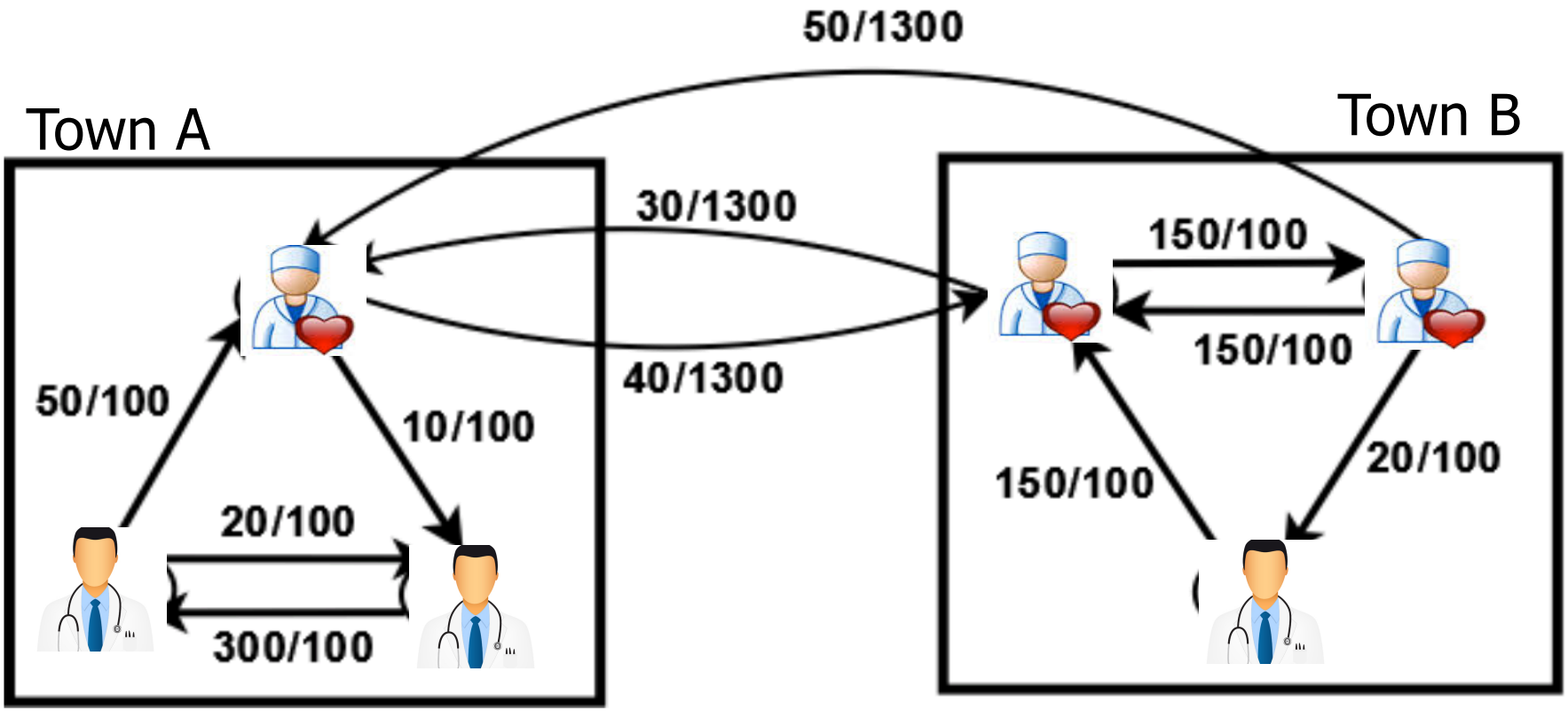
Medical referral network
(weighted, directed)

Ranking in rich networks: example



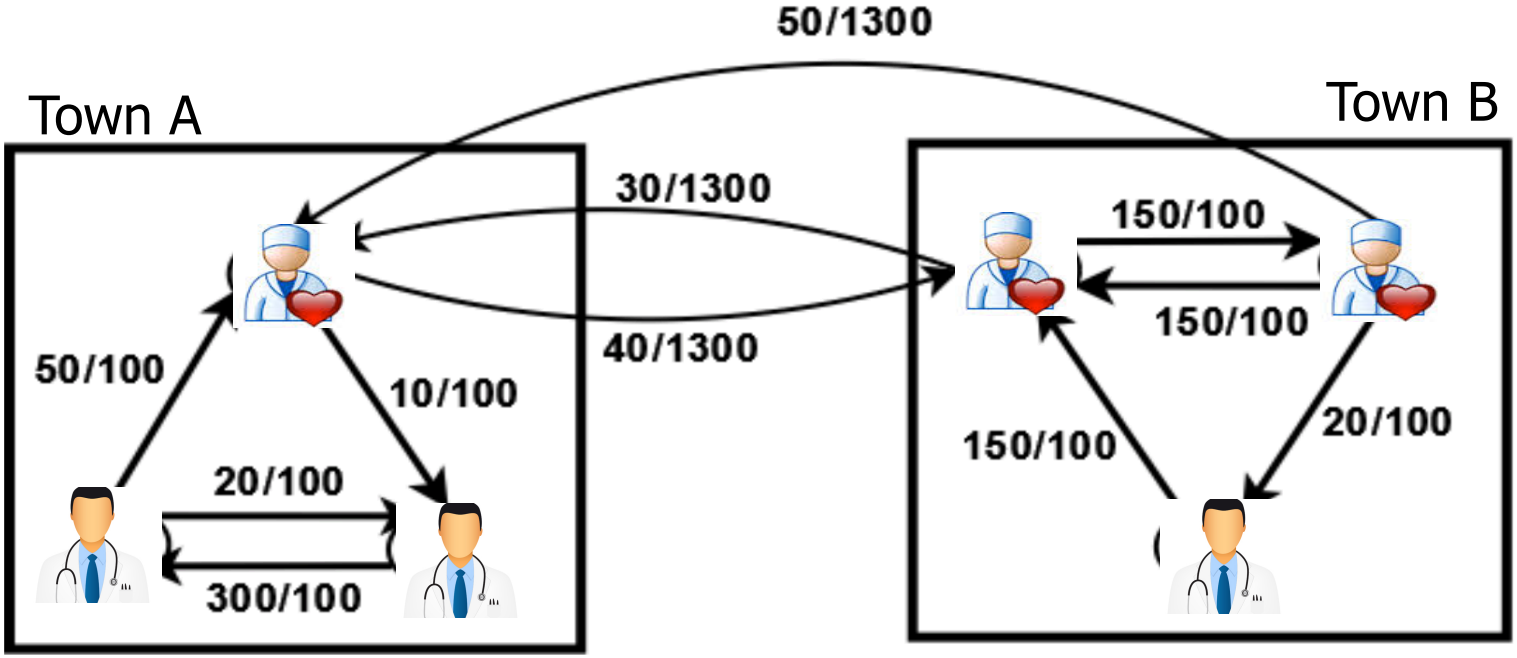
Medical referral network
+ physician expertise

Ranking in rich networks: example



Medical referral network
+ physician expertise
+ location

Ranking in rich networks



Ranking Problem: Which are the top k nodes of a certain type?

e.g.: Who are the best cardiologists in the network, in my town, etc.?

Ranking in Heterogeneous Networks with Geo-Location Information
Abhinav Mishra & Leman Akoglu SIAM SDM 2017.

Modeling the ranking problem

Goal: ranking in directed heterogeneous information networks (HIN) with geo-location

- HINside model
  1. Relation strength
  2. Relation distance
  3. Neighbor authority
  4. Authority transfer rates
  5. Competition
  - ❖ Closed form solution
- Parameter estimation

HINside model

Relation Strength and Distance

- edge weights

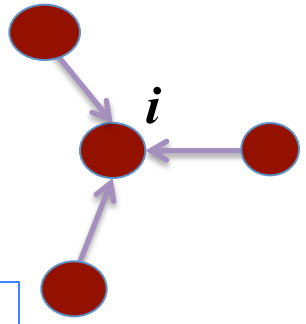
$$W(i,j) = \log(w(i,j) + 1)$$

- pair-wise distances

$$D(i,j) = \log(d(l_i, l_j) + 1)$$

$$M = W \odot D \tag{3.1}$$

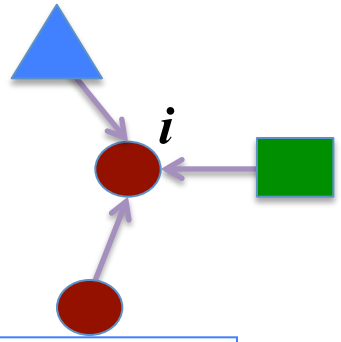
HINside model



In-neighbor authority

$$(3.2) \qquad r_i = \sum_{j \in \mathcal{V}} M(j,i)\, r_j$$

r_i : authority score of node i



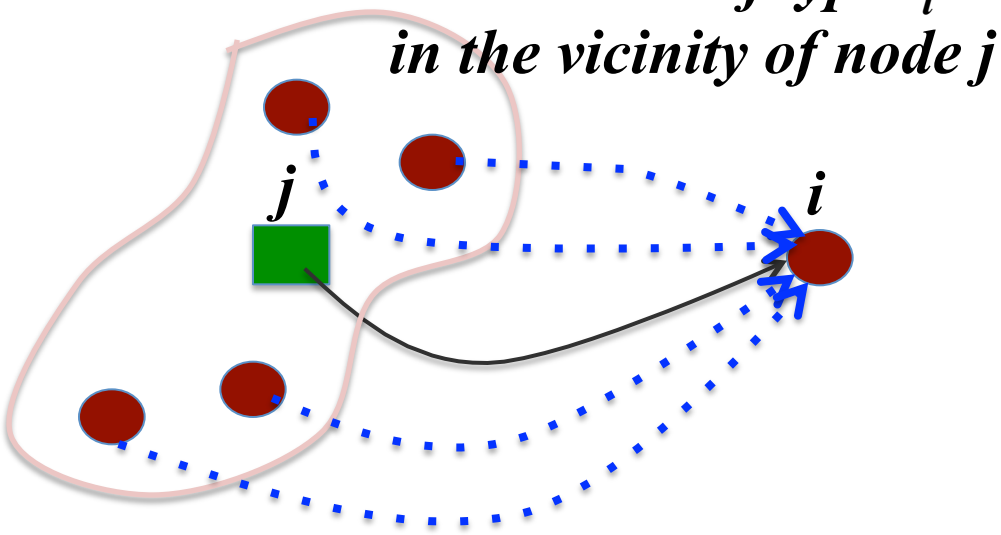
Authority Transfer Rates (ATR)

$$(3.3) \qquad r_i = \sum_{j \in \mathcal{V}} \Gamma(t_j, t_i)\, M(j,i)\, r_j$$

t_i : type of node i

HINside model

Competition



$$N(u,v) = \begin{cases} g(d(l_u, l_v)) & u, v \in \mathcal{V},\ u \neq v \\ 0 & u = v \end{cases}$$

for monotonically decreasing $g(z) = e^{-z}$

$$(3.4)\quad r_i = \sum_j \Gamma(t_j, t_i)\, M(j,i)\, \Big(r_j + \sum_{v:t_v=t_i} N(v,j)\, r_v \Big)$$

Closed-form

- Authority scores vector **r** written in closed form (& computed by power iterations) as :

$$\mathbf{r} = \left[L' + (L'N' \odot E)\right] \mathbf{r} = H\,\mathbf{r}$$

  - $L = M \odot (T\,\Gamma\,T')$
    - T (n x m) where $T(i,c) = 1$ if $t_i = \mathcal{T}(c)$
    - Γ (m x m) **authority transfer rates (ATR)**
  - where $E(u,v) = \begin{cases} 1 & if\ t_u = t_v \\ 0 & \text{otherwise} \end{cases}$

    $E = TT'$

n: #nodes m: #types

Modeling the ranking problem

Goal: ranking in directed heterogeneous information networks (HIN) with geo-location

- HINside model
  1. Relation strength
  2. Relation distance
  3. Neighbor authority
  4. Authority transfer rates
  5. Competition
  - ❖ Closed form solution

➡ Parameter estimation

Parameter estimation

- HINside's parameters consist of the m^2 authority transfer rates (ATR)

$$(3.4)\quad r_i = \sum_j \Gamma(t_j, t_i)\, M(j,i)\,\Big(r_j + \sum_{v:t_v=t_i} N(v,j)\, r_v \Big)$$

  - r_i as a **vector-vector product**

$$r_i = \sum_t \Gamma(t, t_i) \sum_{j:t_j=t} \Big[M(j,i)\big(r_j + \sum_{v:t_v=t_i} N(v,j)\, r_v\big)\Big]$$

$$r_i = \sum_t \Gamma(t, t_i) X(t,i)$$

$$= \Gamma'(t_i, :) \cdot X(:, i) = \mathbf{\Gamma}'_{t_i} \cdot \mathbf{x}_i$$

$$= f(\mathbf{x}_i) = < \mathbf{w}, \mathbf{x}_i >$$

An alternating optimization scheme:

- $\Gamma \longrightarrow \mathbf{r} \longrightarrow X \xrightarrow{\text{estimate}} \Gamma$

Given: graph G, (partial) lists ranking a subset of nodes of a certain type

- Randomly initialize Γ^0 , $k = 0$
- Compute authority scores **r** using Γ^0
- **Repeat**
  - X^k ← compute feature vectors using **r**
  - Γ^{k+1} ← learn new parameters by learning-to-rank
  - compute authority scores **r** using Γ^{k+1}
- **Until** convergence

An alternating optimization scheme:

- $\Gamma \longrightarrow \mathbf{r} \longrightarrow X \xrightarrow{\text{estimate}} \Gamma$

Given: graph G, (partial) lists ranking a subset of nodes of a certain type

- Randomly initialize Γ^0 , $k = 0$
- Compute authority scores **r** using Γ^0
- **Repeat**
  - X^k ← compute feature vectors using **r**
  - Γ^{k+1} ← learn new parameters by learning-to-rank
  - compute authority scores **r** using Γ^{k+1}
- **Until** convergence

RankSVM formulation

Cross-entropy based objective by gradient descent

- Given partial ranked lists;
  - create all pairs (u, v)
  - add training data $\{((\mathbf{x}_d^1, \mathbf{x}_d^2), y_d)\}_{d=1}^{|\mathcal{D}|}$

    $((\mathbf{x}_u, \mathbf{x}_v), 1)$ if u ranked ahead of v

    $((\mathbf{x}_u, \mathbf{x}_v), -1)$ otherwise
  - for each type t, solve:

$$\min_{\mathbf{\Gamma}_t} \ ||\mathbf{\Gamma}_t||_2^2 \ + \ \gamma \sum_{d \in \mathcal{D}} \epsilon_d$$

$$\text{s.t. } \mathbf{\Gamma}_t'(\mathbf{x}_d^1 - \mathbf{x}_d^2) y_d \geq 1 - \epsilon_d, \ \forall d \in \mathcal{D} \text{ and } t_{\mathbf{x}_d^1}, t_{\mathbf{x}_d^2} = t$$

$$\epsilon_d \geq 0, \ \forall d \in \mathcal{D}$$

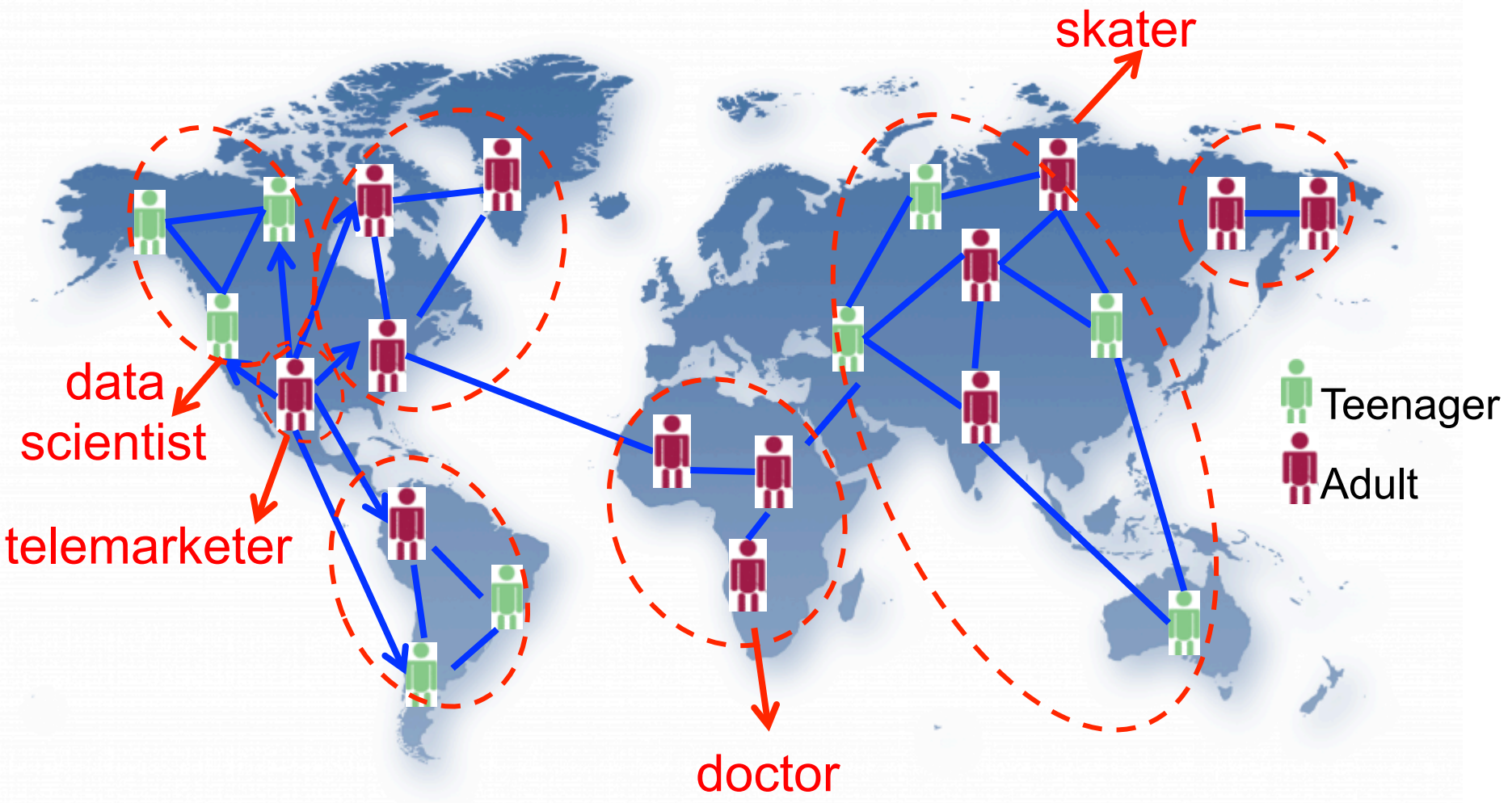
$$\mathbf{\Gamma}_t(c) \geq 0, \ \forall c = 1, \ldots, m$$

Graph problems on rich networks

- ranking,
- clustering & anomaly mining,
- classification,
- link prediction,
- role discovery,
- similarity search,
- influence,
- evolution,
- …

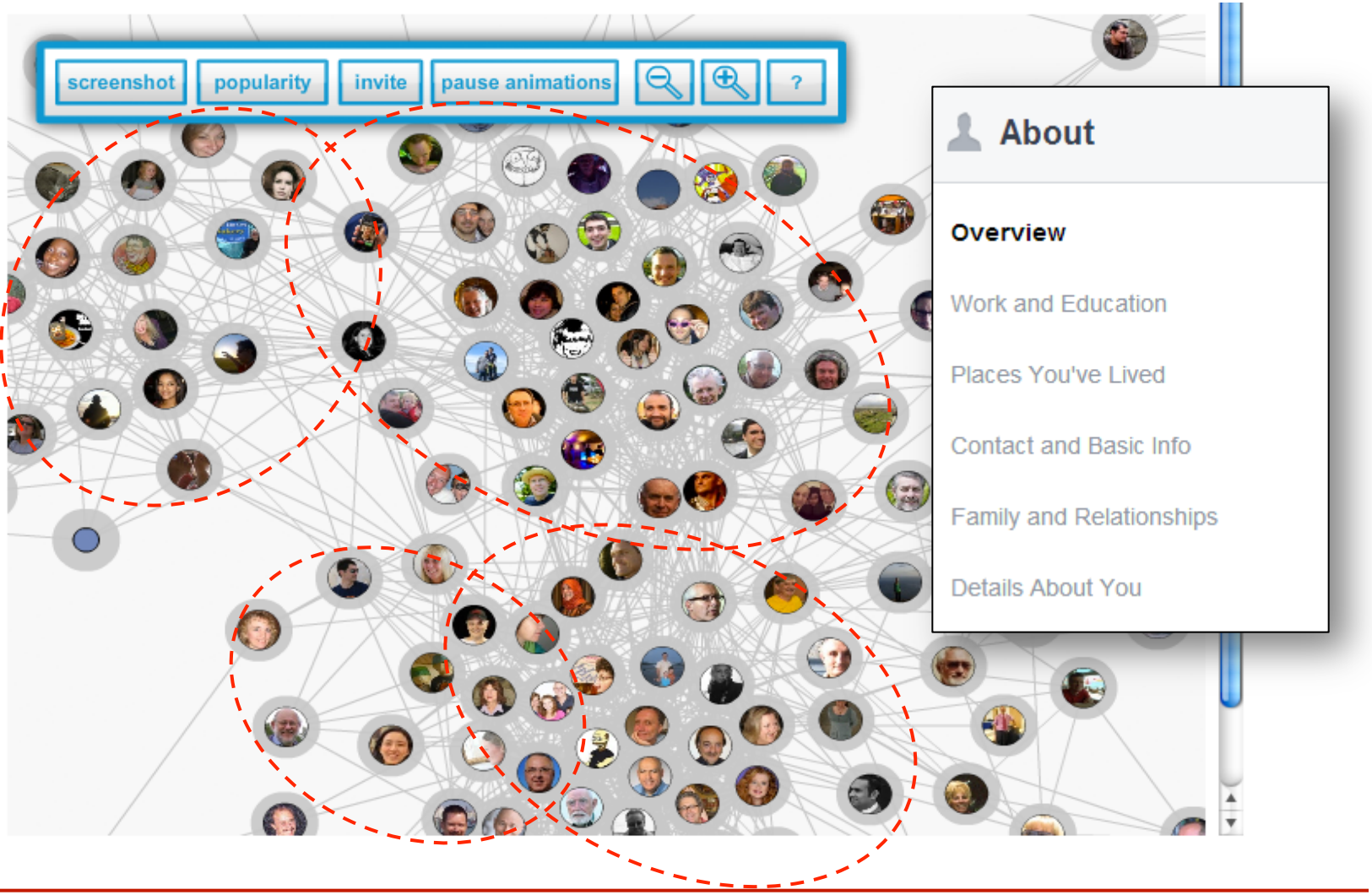
Attributed graphs

Attributed graph: each node has 1+ properties



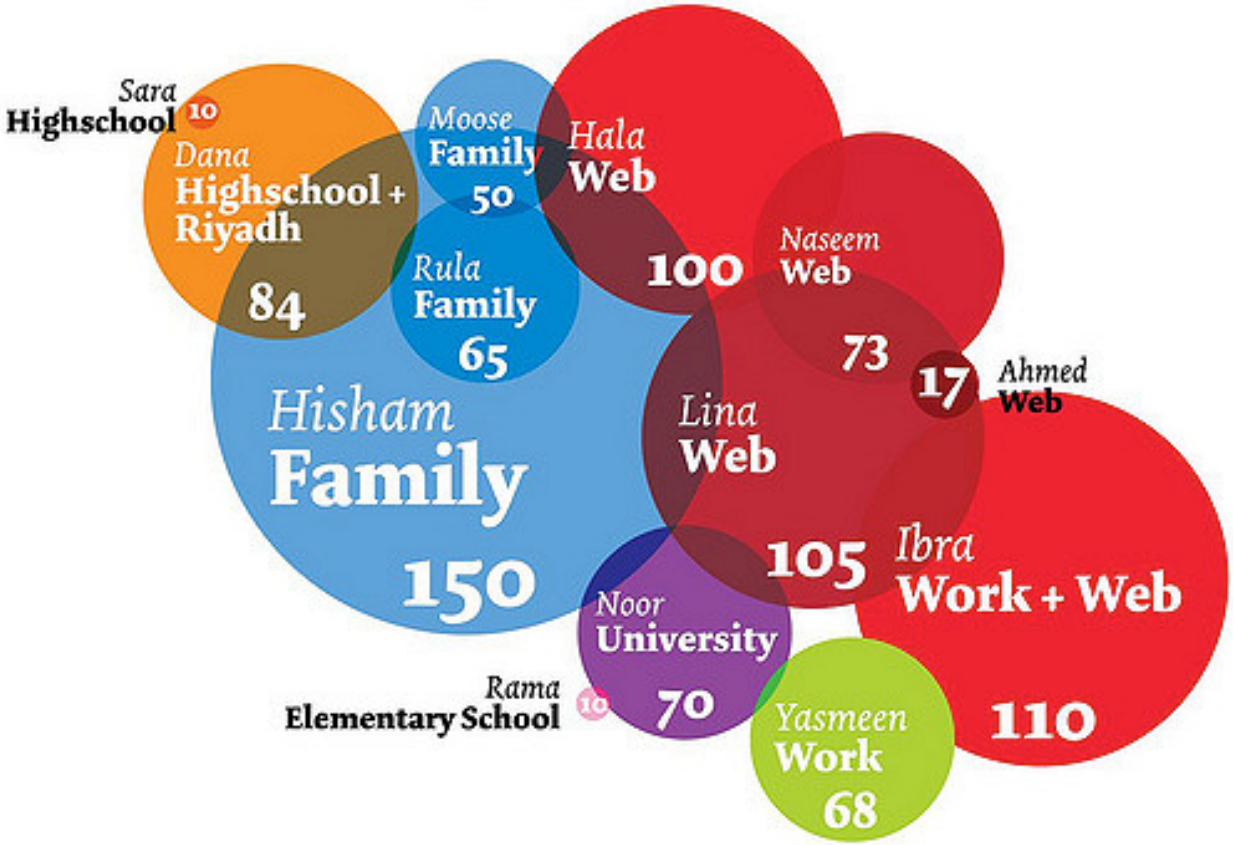
Communities in rich networks

Attributed graph: each node has 1+ properties



Anomalous subgraphs

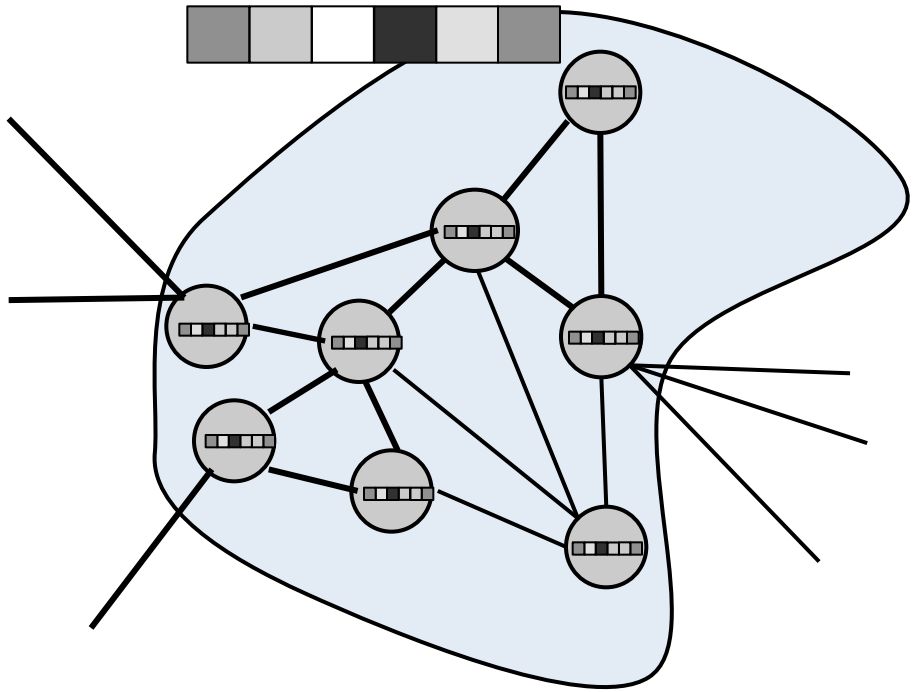
Given a set of attributed subgraphs* (e.g. Google+ circles), **Find** **poorly-defined** ones



* social circles, communities, egonetworks, …

Communities in attributed networks

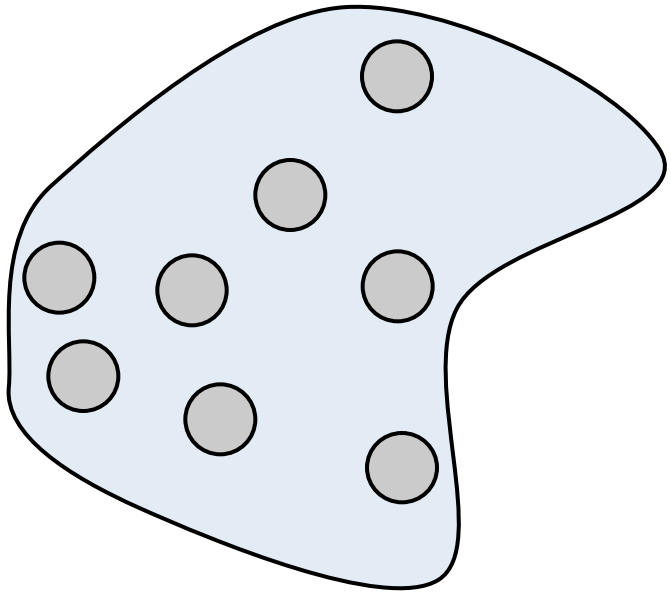
Given an attributed subgraph*, how to **quantify** its quality?



* social circles, communities, egonetworks, …

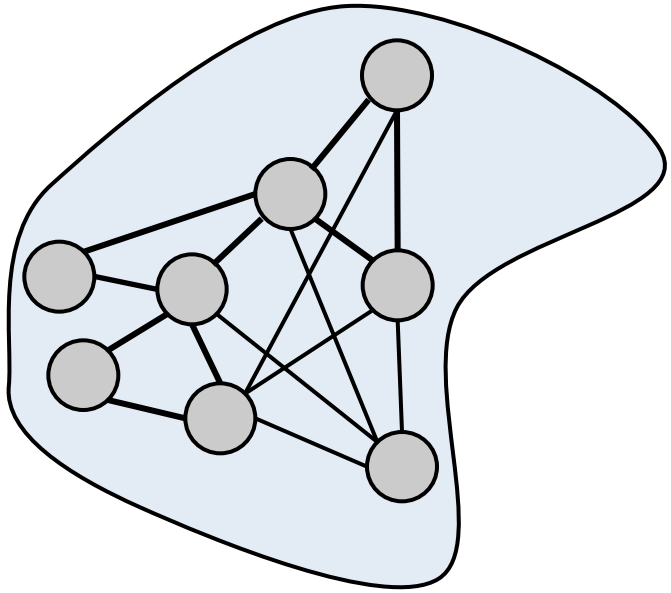
Communities in attributed networks

- **Given** a subgraph,
  how to **quantify** its quality?



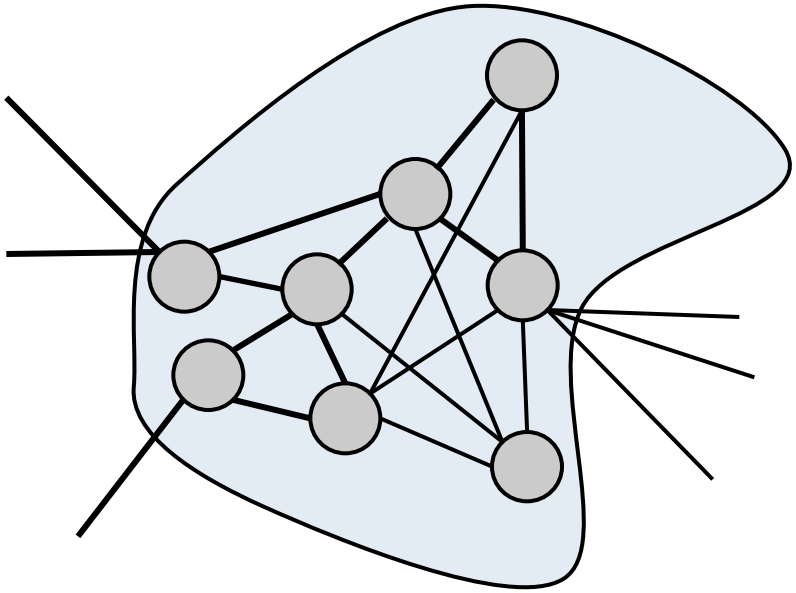
Communities in attributed networks

- **Given** a subgraph,
  how to **quantify** its quality?
  - Structure-only
    - Internal measures
      - e.g. average degree



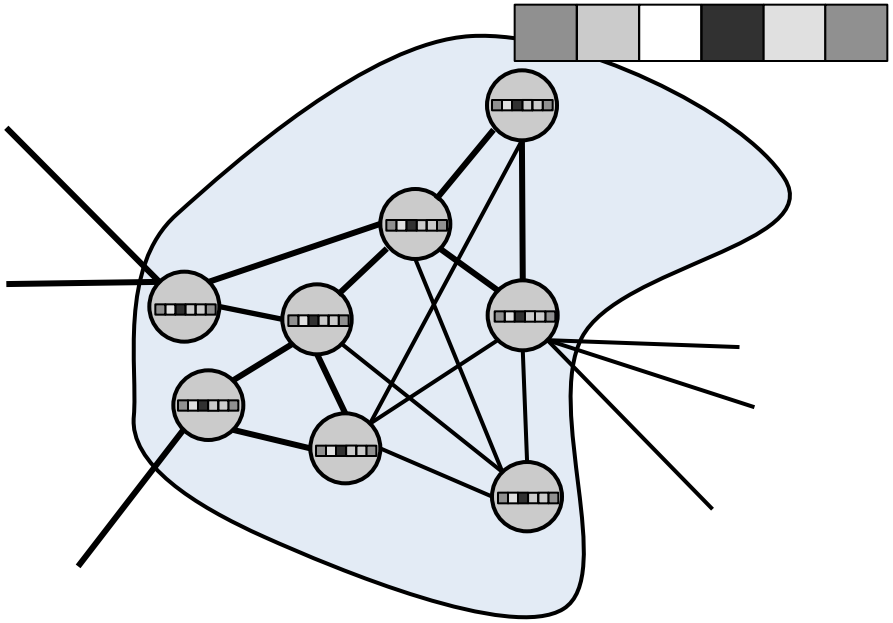
Communities in attributed networks

- **Given** a subgraph, how to **quantify** its quality?
  - Structure-only
    - Internal-only
      - average degree
    - Boundary-only
      - cut edges
    - Internal + Boundary
      - conductance



Communities in attributed networks

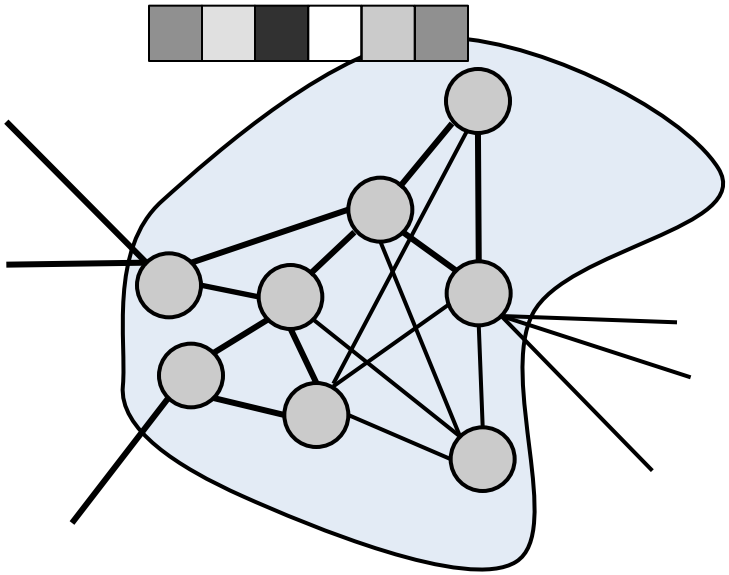
- **Given** an attributed subgraph, how to **quantify** its quality?
  - Structure-only
    - Internal-only
      - average degree
    - Boundary-only
      - cut edges
    - Internal + Boundary
      - conductance
  - Structure + Attributes?

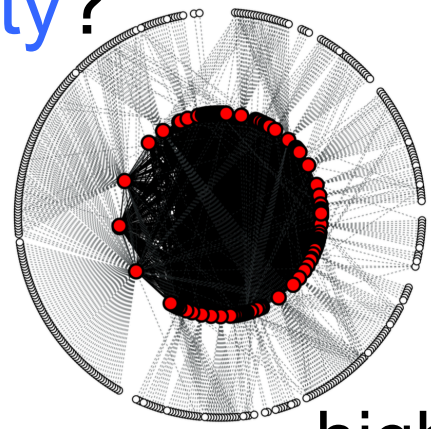


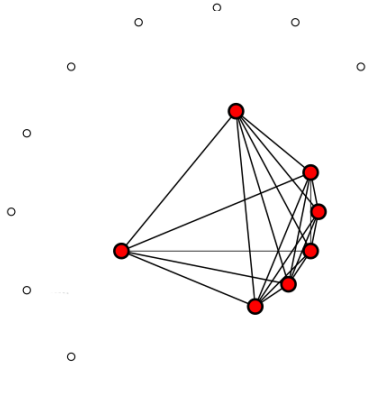
Scalable Anomaly Ranking of Attributed Neighborhoods
Bryan Perozzi and Leman Akoglu SIAM SDM 2016.

What's an Anomaly, Anyhow?

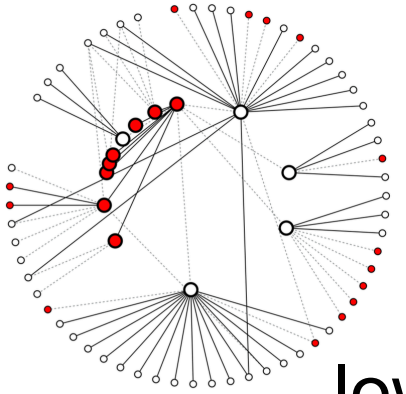
- **Given** an attributed subgraph
  how to **quantify** quality?

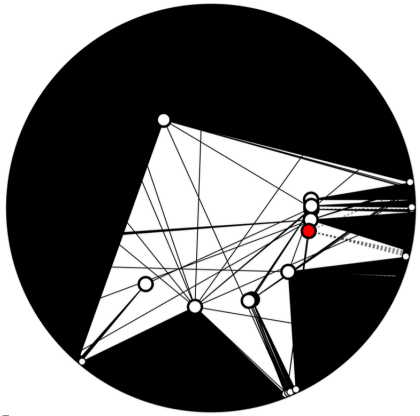






high

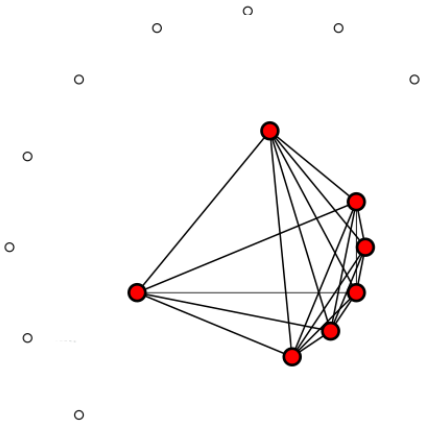




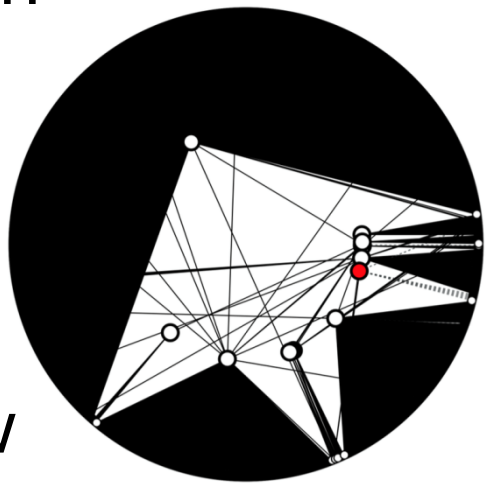
low

Normality (intuition)

- **Given** an attributed subgraph how to **quantify** quality?
  - Internal
    - structural density



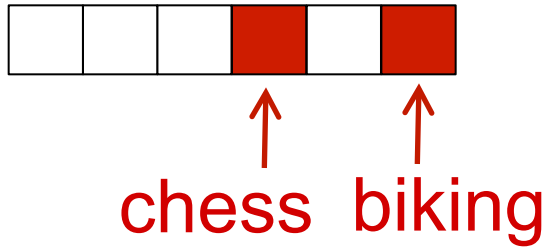
high



low

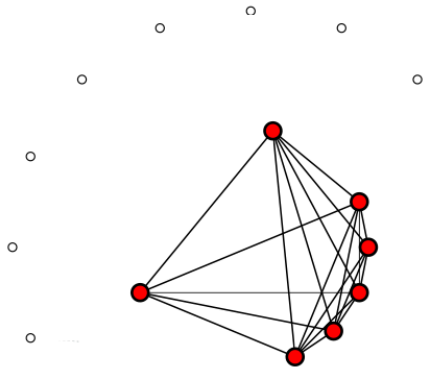
Normality (intuition)

- **Given** an attributed subgraph how to **quantify** quality?
  - Internal
    - structural density AND
    - attribute coherence
      - *neighborhood* ***"focus"***

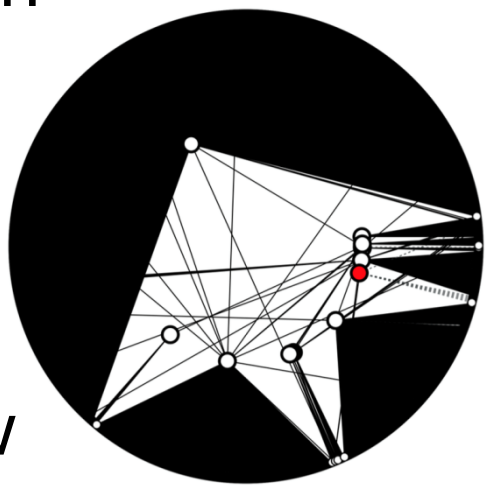


Normality (intuition)

- **Given** an attributed subgraph how to **quantify** quality?
  - Internal
    - structural density AND
    - attribute coherence
      - *neighborhood* ***"focus"***
  - Boundary
    - structural sparsity, OR
    - external separation
      - ***"exoneration"***



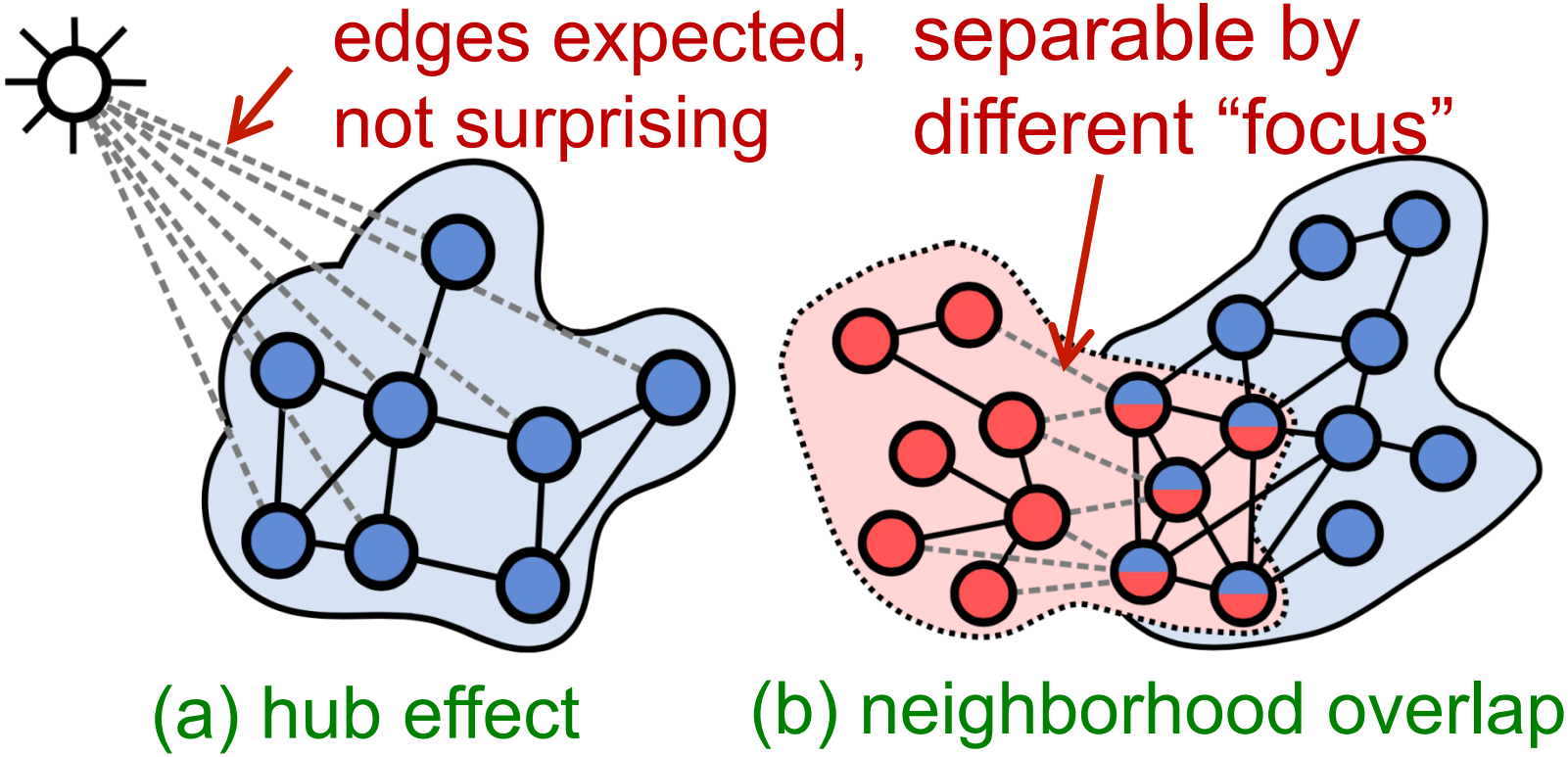
high



low

Normality (intuition)

- Motivation:
  - no good cuts in real-world graphs [Leskovec+ '08]
  - social circles overlap [McAuley+ '14]
- ***"exoneration"*** : by (a) null model, (b) attributes



The measure of Normality

$$N = I + E = \sum_{i \in C, j \in C} \left(A_{ij} - \frac{k_i k_j}{2m}\right) s(\mathbf{x_i}, \mathbf{x_j} | \mathbf{w})$$

$$- \sum_{\substack{i \in C, b \in B \\ (i,b) \in \mathcal{E}}} \left(1 - \min(1, \frac{k_i k_b}{2m})\right) s(\mathbf{x_i}, \mathbf{x_b} | \mathbf{w})$$

1

The measure of Normality

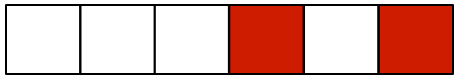
Null model

$$N = I + E = \sum_{i \in C, j \in C} \left(A_{ij} - \frac{k_i k_j}{2m}\right) s(\mathbf{x_i}, \mathbf{x_j} | \mathbf{w})$$

internal consistency

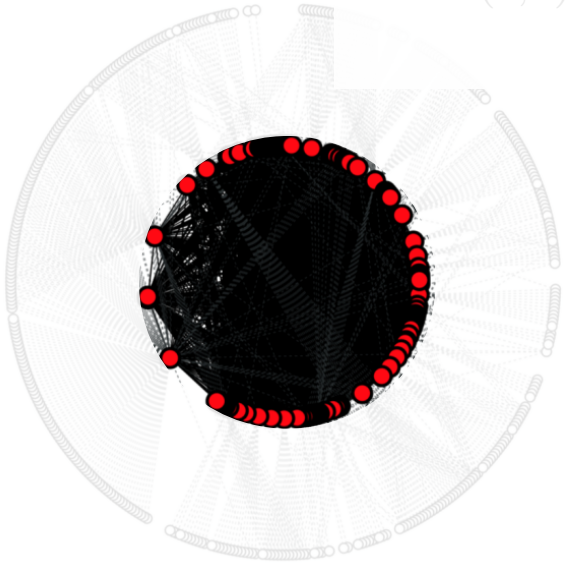
similarity

"focus" vector



chess biking

1



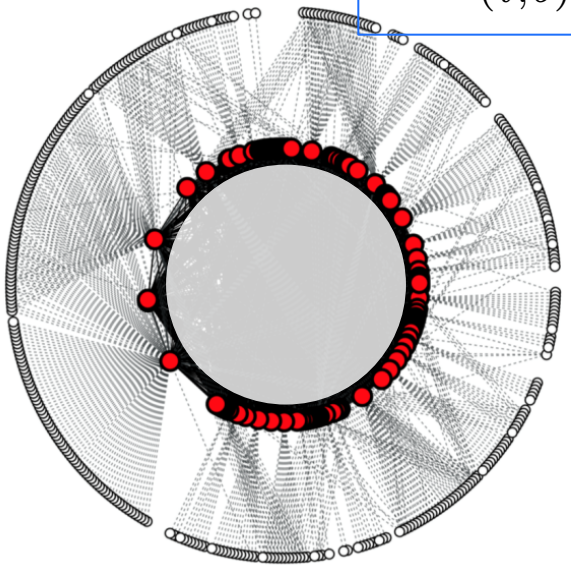
The measure of Normality

$$N = I + E = \sum_{i \in C, j \in C} \left(A_{ij} - \frac{k_i k_j}{2m}\right) s(\mathbf{x_i}, \mathbf{x_j} | \mathbf{w})$$

external separability

$$- \sum_{\substack{i \in C, b \in B \\ (i,b) \in \mathcal{E}}} \left(1 - \min(1, \frac{k_i k_b}{2m})\right) s(\mathbf{x_i}, \mathbf{x_b} | \mathbf{w})$$

1



Anomaly Mining of Entity Neighborhoods (AMEN)

- Given an attributed subgraph, can we find the attribute weights?

$$N = I + E = \sum_{i \in C, j \in C} \left(A_{ij} - \frac{k_i k_j}{2m}\right) s(\mathbf{x_i}, \mathbf{x_j} | \mathbf{w}) - \sum_{\substack{i \in C, b \in B \\ (i,b) \in \mathcal{E}}} \left(1 - \min(1, \frac{k_i k_b}{2m})\right) s(\mathbf{x_i}, \mathbf{x_b} | \mathbf{w})$$

1

latent

$$\mathbf{w_C}^T \cdot \left[\sum_{i \in C, j \in C} \left(A_{ij} - \frac{k_i k_j}{2m}\right) s(\mathbf{x_i}, \mathbf{x_j}) - \sum_{\substack{i \in C, b \in B \\ (i,b) \in \mathcal{E}}} \left(1 - \min(1, \frac{k_i k_b}{2m})\right) s(\mathbf{x_i}, \mathbf{x_b}) \right]$$

2

Optimizing Normality

$$N = I + E = \sum_{i \in C, j \in C} \left(A_{ij} - \frac{k_i k_j}{2m}\right) s(\mathbf{x_i}, \mathbf{x_j} | \mathbf{w}) - \sum_{\substack{i \in C, b \in B \\ (i,b) \in \mathcal{E}}} \left(1 - \min(1, \frac{k_i k_b}{2m})\right) s(\mathbf{x_i}, \mathbf{x_b} | \mathbf{w}) \quad (1)$$

$$\max_{\mathbf{w_C}} \quad \mathbf{w_C}^T \cdot \left[\sum_{i \in C, j \in C} \left(A_{ij} - \frac{k_i k_j}{2m}\right) s(\mathbf{x_i}, \mathbf{x_j}) - \sum_{\substack{i \in C, b \in B \\ (i,b) \in \mathcal{E}}} \left(1 - \min(1, \frac{k_i k_b}{2m})\right) s(\mathbf{x_i}, \mathbf{x_b}) \right] \quad (2)$$

$$\max_{\mathbf{w_C}} \quad \mathbf{w_C}^T \cdot (\hat{\mathbf{x}}_I + \hat{\mathbf{x}}_E) \quad (3)$$

$$\text{s.t.} \quad \|\mathbf{w_C}\|_p = 1, \ \ \mathbf{w_C}(f) \geq 0, \ \forall f = 1 \ldots d$$

Optimizing Normality

$$\max_{\mathbf{w_C}} \quad \mathbf{w_C}^T \cdot \underbrace{(\hat{\mathbf{x}}_I + \hat{\mathbf{x}}_E)}_{\mathbf{x}}$$

$$\text{s.t.} \quad \|\mathbf{w_C}\|_p = 1, \ \mathbf{w_C}(f) \geq 0, \ \forall f = 1 \ldots d$$

$p = 1$: $\mathbf{w_C}(f) = 1$ **one** attribute f with largest **x**

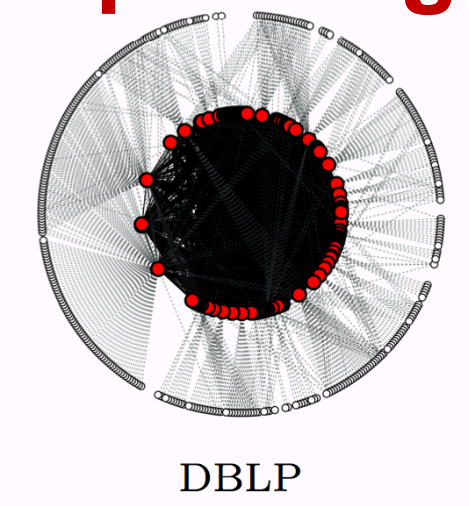
$p = 2$: $\mathbf{w_C}(f) = \frac{\mathbf{x}(f)}{\sqrt{\sum_{\mathbf{x}(i)>0} \mathbf{x}(i)^2}}$ **all** f with positive **x**

Linear in number of attributes!

Normality becomes

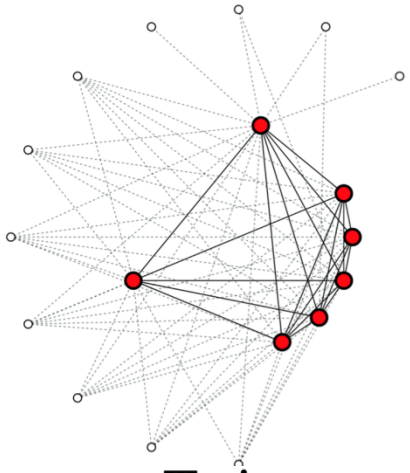
$$N = \mathbf{w_C}^T \cdot \mathbf{x} = \|\mathbf{x}_+\|_2$$

Example neighborhoods



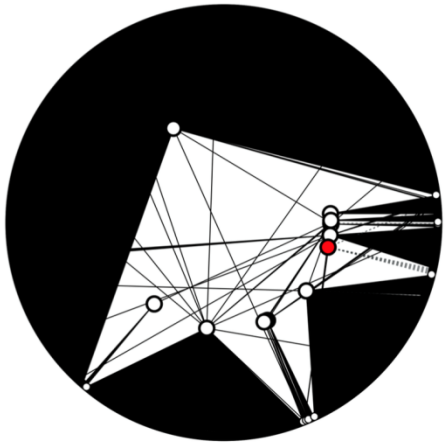
DBLP

$L_1 = 0.979,\ L_2 = 2.17$



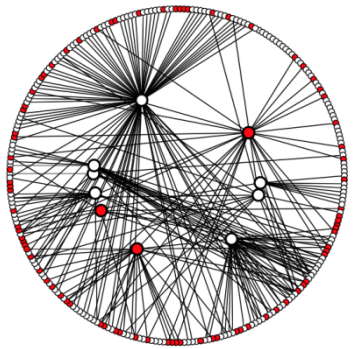
Twitter

$L_1 = 0.724,\ L_2 = 1.10$



Citeseer

$L_1 = L_2 = -0.956$



Google+

$L_1 = L_2 = -0.873$

Graph problems on rich networks

- ranking,
- clustering & anomaly mining,
- classification,
- link prediction,
- role discovery,
- similarity search,
- influence,
- evolution,
- …

Motivating Problem

Connotation Mining:

finding dash of sentiment beneath
"seemingly objective" **words** & **senses**

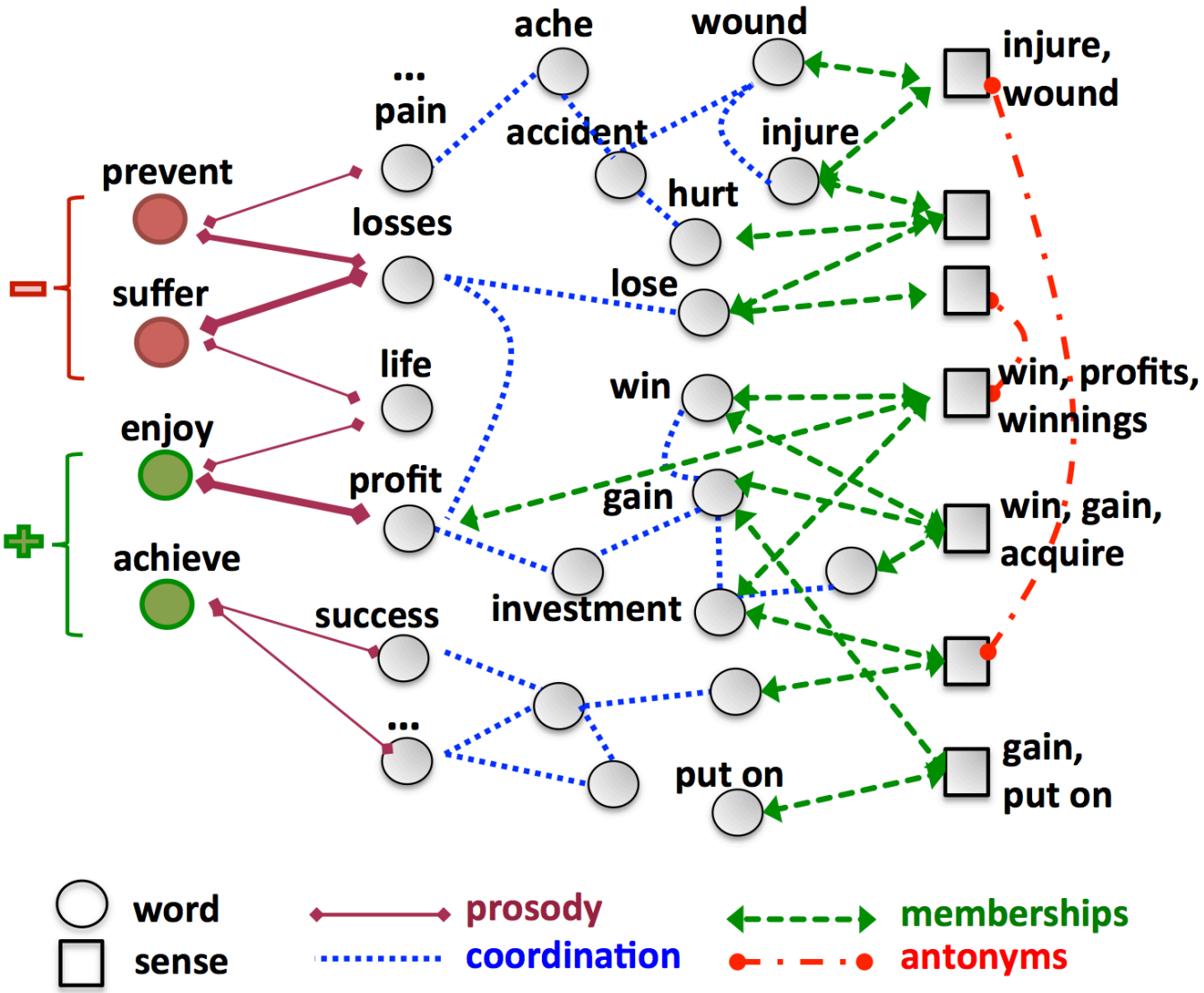
cheesecake

emission

fine

fine

Words+Senses edge-typed network



Classification

- A collective classification approach
  - Objective utilizes pairwise Markov Random Fields

$$\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) = \frac{1}{\mathcal{Z}(\mathbf{x})} \prod_{Y_i \in \mathcal{Y}} \phi_i(y_i) \prod_{(Y_i,Y_j)\in E} \psi_{ij}^{t}(y_i, y_j)$$

edge type

Node labels as random variables

edge potential (label-label)

$$\phi_i(y_i) = \psi_i(y_i) \prod_{(Y_i,X_j)\in E} \psi_{ij}^{t}(y_i)$$

prior belief

edge potential (label-observed label)

Edge potentials depend on edge type

t: t_1	A	
P	+	−
+	1-ϵ	ϵ
−	ϵ	1-ϵ

(t_1) *pred-arg*

t: t_2	A	
A	+	−
+	1-2ϵ	2ϵ
−	2ϵ	1-2ϵ

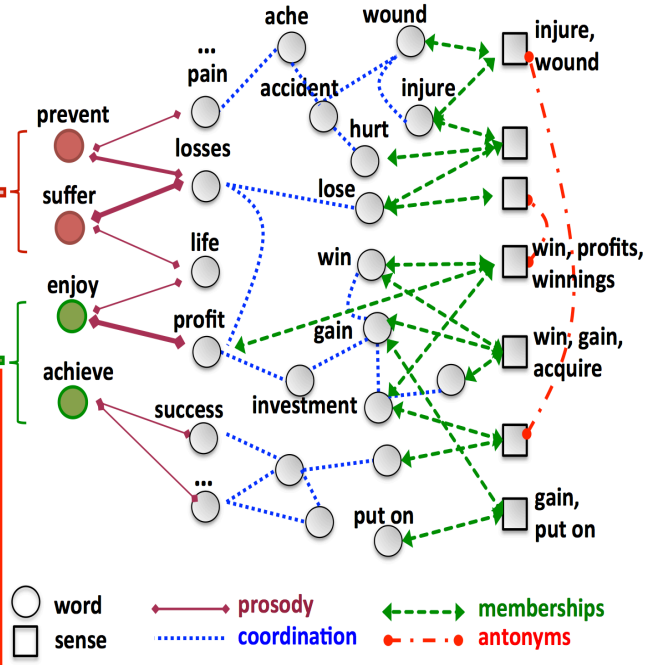
(t_2) *arg-arg*

t: t_3	A	
S	+	−
+	1-ϵ	ϵ
−	ϵ	1-ϵ

(t_3) *syn-arg*
(synonym relations)

t: t_4	S	
S	+	−
+	ϵ	1-ϵ
−	1-ϵ	ϵ

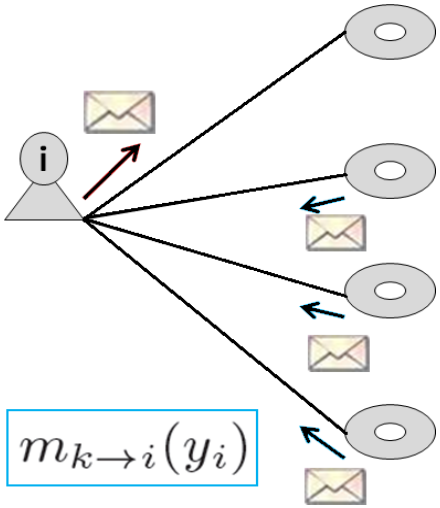
(t_4) *syn-syn*
(antonym relations)



Inference

- A collective classification approach
  - Objective utilizes pairwise Markov Random Fields
    - Inference problem (NP-hard)
- Loopy Belief Propagation (LBP)

1) Repeat for each node:



$$m_{i \to j}(y_j) = \alpha \sum_{y_i \in \mathcal{L}_{T_i}} \left(\phi_i(y_i) \, \psi_{ij}^{t}(y_i, y_j) \right)$$

edge type

edge potential

2) At convergence: $b_i(y_i) = \beta \; \phi_i(y_i) \prod_{Y_j \in \mathcal{Y}_{\mathcal{N}_i}} m_{j \to i}(y_i)$

Summary

- Ranking in node-typed graphs with location
  - motivating domain: physician referrals
  - HINside model for ranking w/ parameter learning
- Anomalous subgraphs in node-attributed graphs
  - motivating domain: social networks
  - AMEN model for quality scoring
- Classification in edge-typed graphs
  - motivating application: connotation mining/NLP
  - LBP with type-specific edge potentials

References

- ***Ranking in Heterogeneous Networks with Geo-Location Information***
  Abhinav Mishra & Leman Akoglu. SIAM SDM 2017.
  Code: https://github.com/abhimm/HINSIDE

- ***Scalable Anomaly Ranking of Attributed Neighborhoods***
  Bryan Perozzi & Leman Akoglu. SIAM SDM 2016.
  Code: https://github.com/phanein/amen

- ***ConnotationWordNet: Learning Connotation of the Word +Sense Network***
  Jun S. Kang, Song Feng, Leman Akoglu, Yejin Choi. ACL 2014.
  http://www3.cs.stonybrook.edu/~junkang/connotation_wordnet/