# Computational Models for Social Influence and Diffusion

Yang Yang and Jie Tang

Zhejiang University

Tsinghua University

Homepage: http://yangy.org    http://keg.cs.tsinghua.edu.cn/jietang/

# How to Model the Diffusion of Social Influence in Networks?

# Compartmental Models in Epidemiology

- The **SIR** model, which is proposed by Kermack and McKendrick in the early 1900s.
- The model predicts infectious diseases

Susceptible $\longrightarrow$ Infectious $\longrightarrow$ Recovered

- Transition rates:

$$\frac{dS}{dt} = -\beta S(t)I(t)$$
$$\frac{dI}{dt} = \beta S(t)I(t) - \gamma I(t)$$
$$\frac{dR}{dt} = \gamma I(t)$$

*S(t) :* susceptible individuals at time t;

*I(t) :* infected individuals at time t;

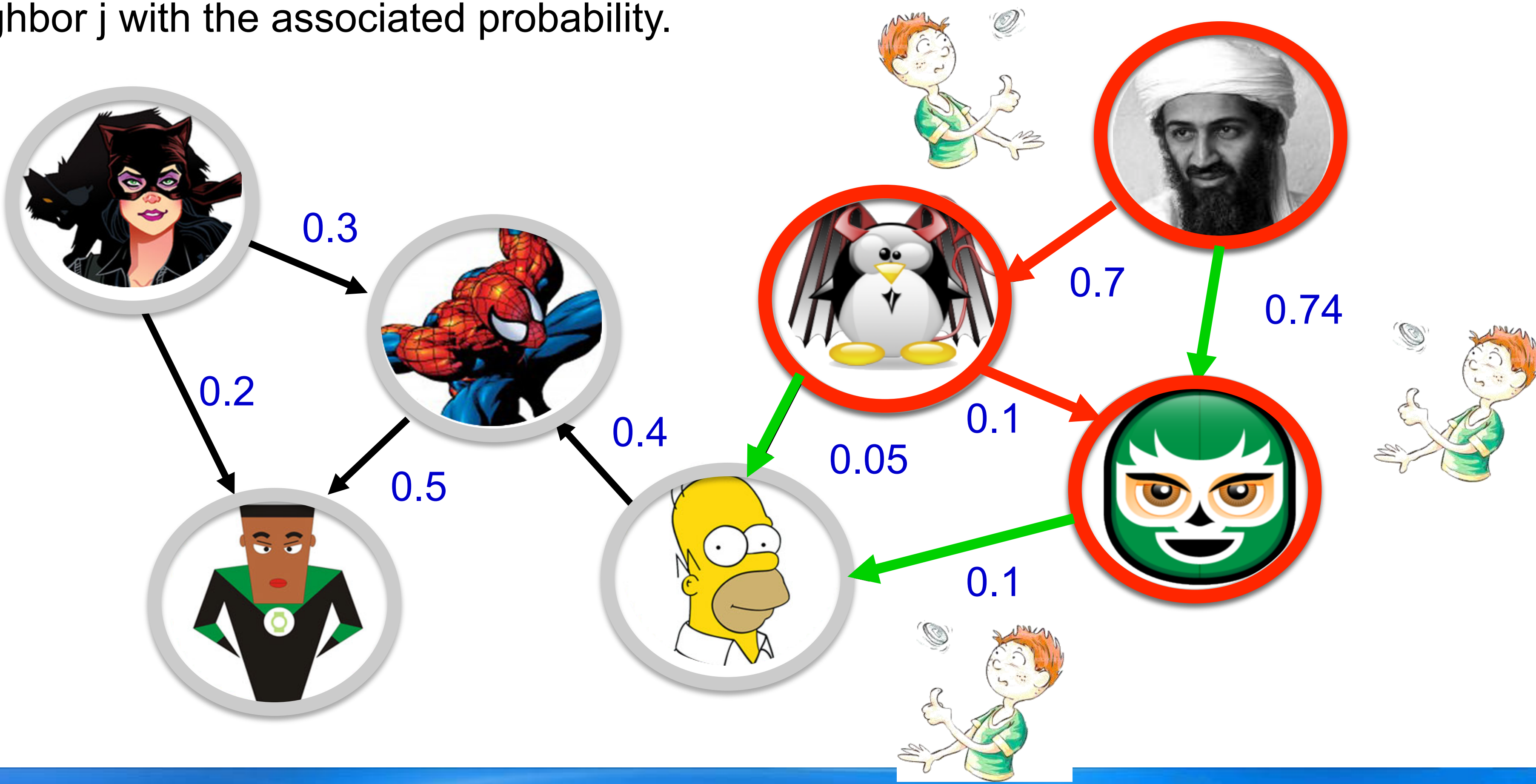
*R(t) :* recovered individuals at t ;

$\beta$ *:* the contact rate;

$\gamma$ *:* rate of recovery.

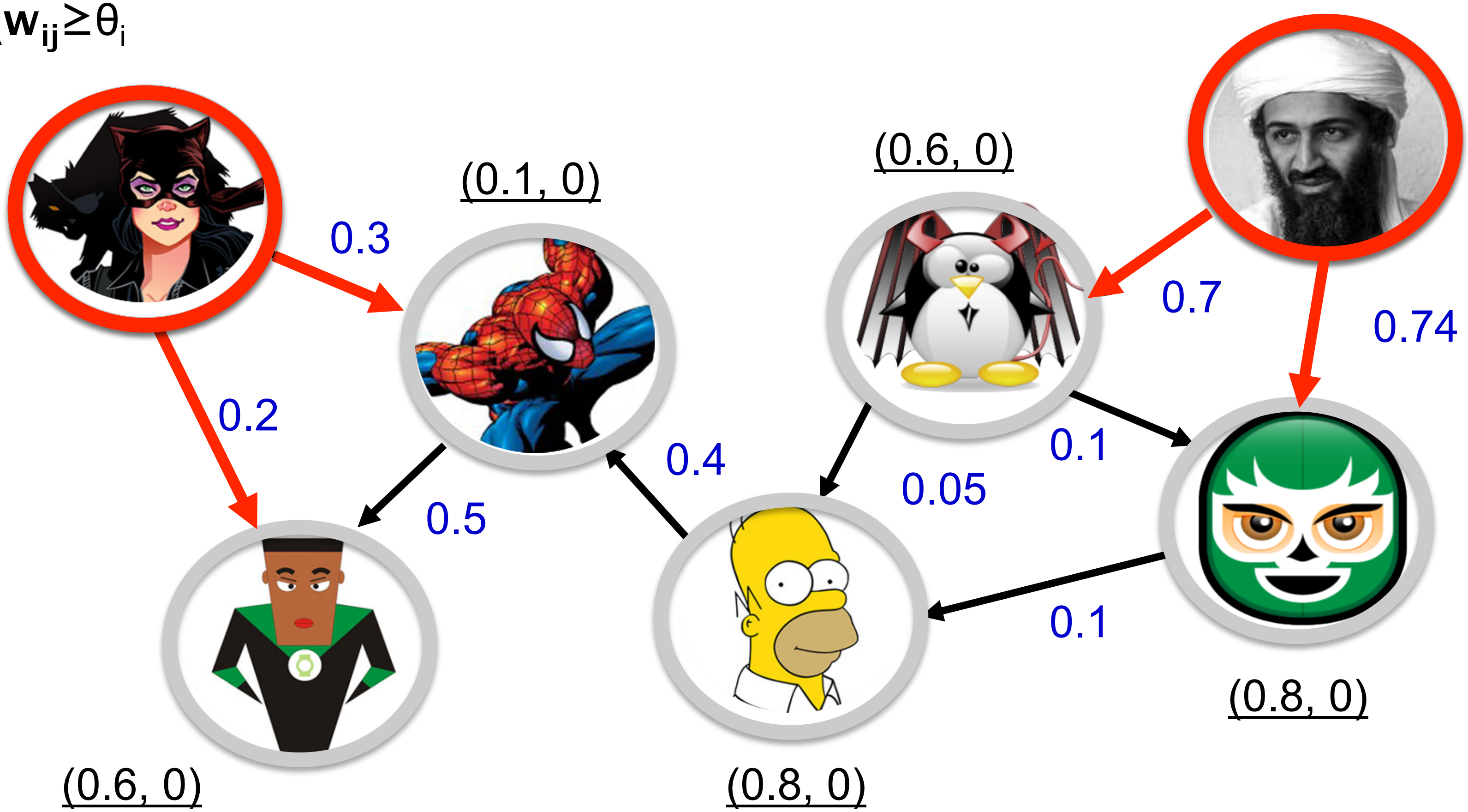
# Independent Cascade Model

- Each edge is associated with a probability $p_{ij}$
- At first time stamp, some nodes become *active* while others are left *inactive*.
- Once a node i becomes *active*, it has a single chance to activate each of its *inactive* neighbor j with the associated probability.



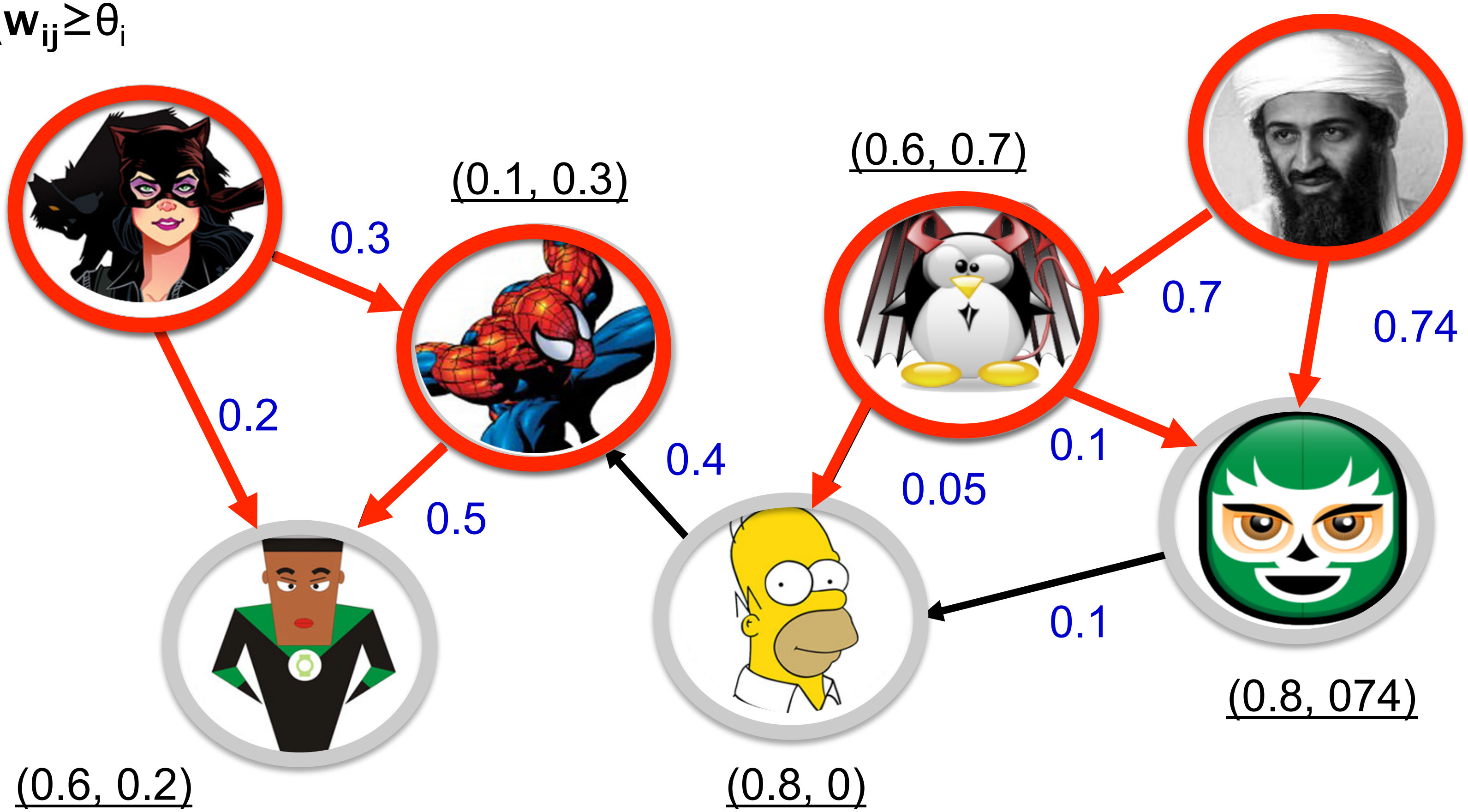
# Linear Threshold Model

- Each edge is associated with a weight $w_{ij}$, s.t. $\mathbf{\Sigma w_{ij} \leq 1}$
- For each node i, assign a random threshold $\theta_i \sim U[0, 1]$
- At first time stamp, some nodes become *active* while others are left *inactive*.
- A node i becomes *active* when its weighted active neighbors exceed the threshold $\mathbf{\Sigma_{j \in A} w_{ij}} \geq \theta_i$



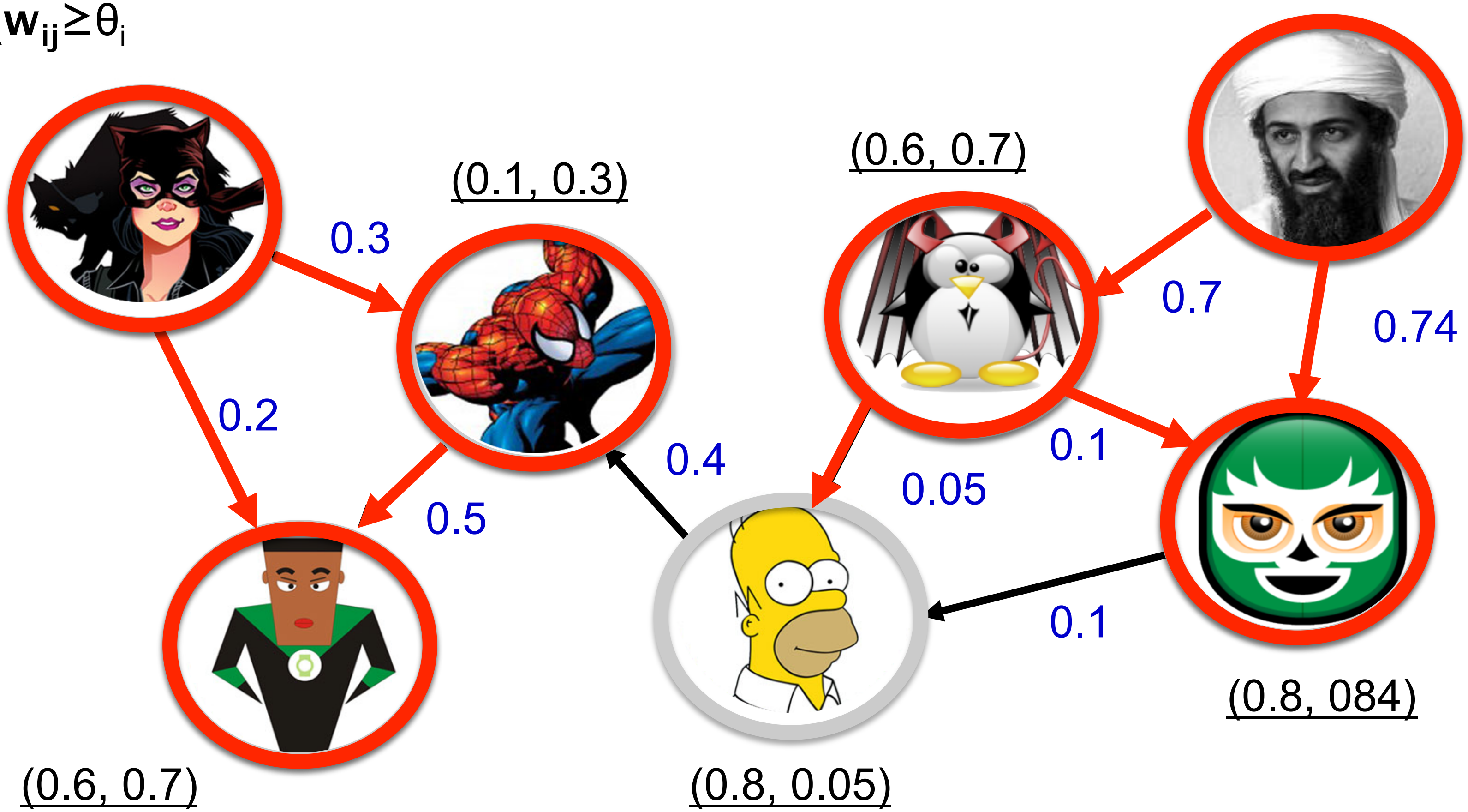
# Linear Threshold Model

- Each edge is associated with a weight $w_{ij}$, s.t. $\mathbf{\Sigma w_{ij} \leq 1}$
- For each node i, assign a random threshold $\theta_i \sim U[0, 1]$
- At first time stamp, some nodes become *active* while others are left *inactive*.
- A node i becomes *active* when its weighted active neighbors exceed the threshold $\mathbf{\Sigma_{j \in A} w_{ij}} \geq \theta_i$



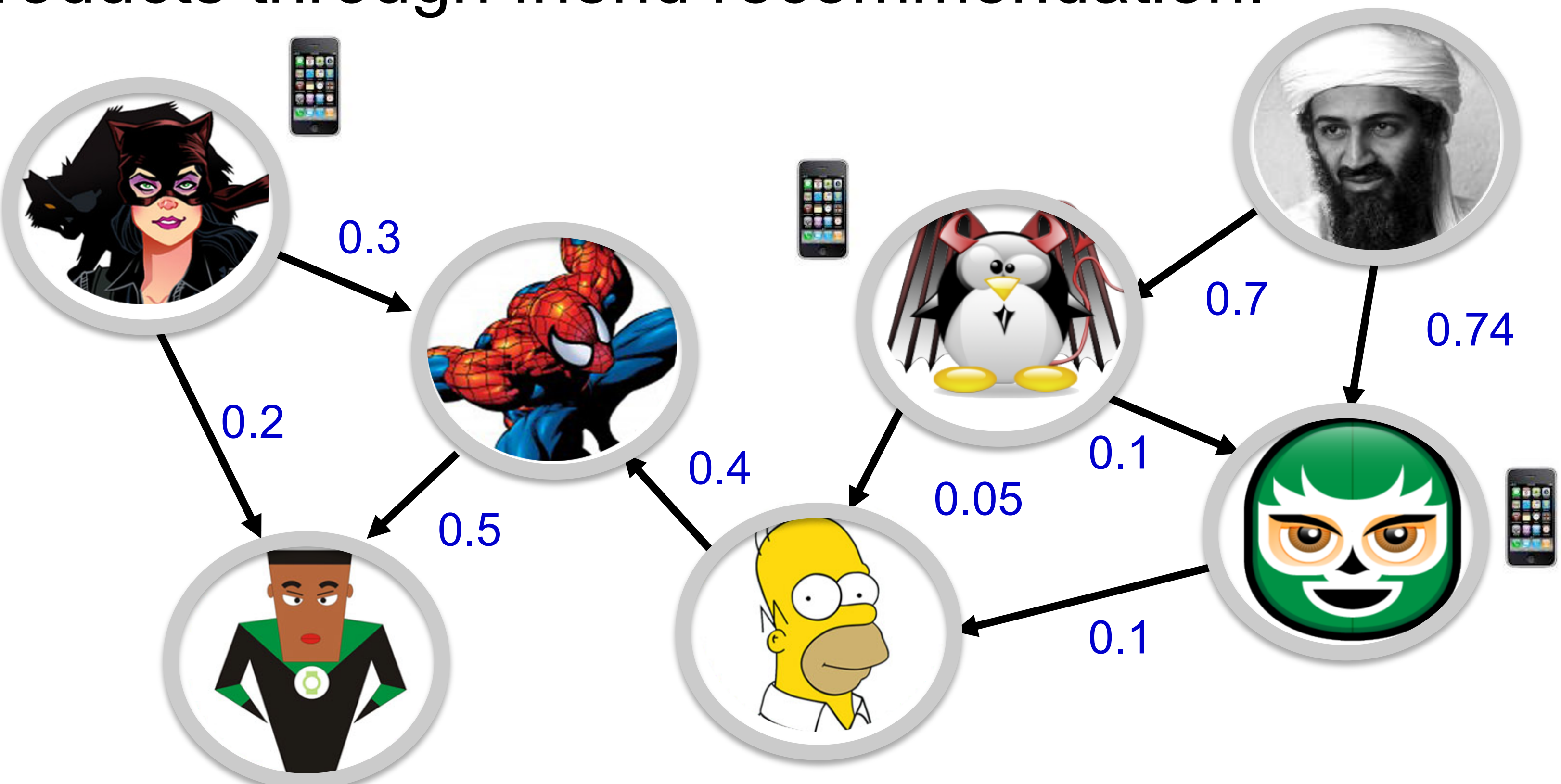
# Linear Threshold Model

- Each edge is associated with a weight $w_{ij}$, s.t. $\boldsymbol{\Sigma w_{ij} \leq 1}$
- For each node i, assign a random threshold $\theta_i \sim U[0, 1]$
- At first time stamp, some nodes become *active* while others are left *inactive*.
- A node i becomes *active* when its weighted active neighbors exceed the threshold $\boldsymbol{\Sigma_{j \in A} w_{ij}} \geq \theta_i$



# Influence Maximization

- Initially targeting a few “influential” seeds, to trigger a maximal number of individuals to adopt the opinions/ products through friend recommendation.




D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In KDD’03, pages 137–146, 2003

# Influence Maximization

- Influence spread *F(S)*
  - *S* is the initial set of activated nodes, i.e., "seed set"
  - Defined as the expected number of active nodes in the end
- Objective
  - For a given budget *k*
  - Find $S^* = \arg\max F(S)$, $|S| = k$
- Challenge
  - The optimization problem is NP-hard

D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In KDD'03, pages 137–146, 2003

# Greedy Algorithm

- Initialize the seed set as an empty set $S \leftarrow \varnothing$
- For *k* times, select a node i which can optimize the marginal gain:

$$i \leftarrow \arg\max[F(S \cup \{i\}) - F(S)]$$

$$S \leftarrow S \cup \{i\}$$

- A performance guarantee?
  - The solution obtained by Greedy is better than 63% (1-1/e) of the optimal solution

$$F(S) \geq (1 - \frac{1}{e})F(S^*)$$

D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In KDD'03, pages 137–146, 2003

# Key Question

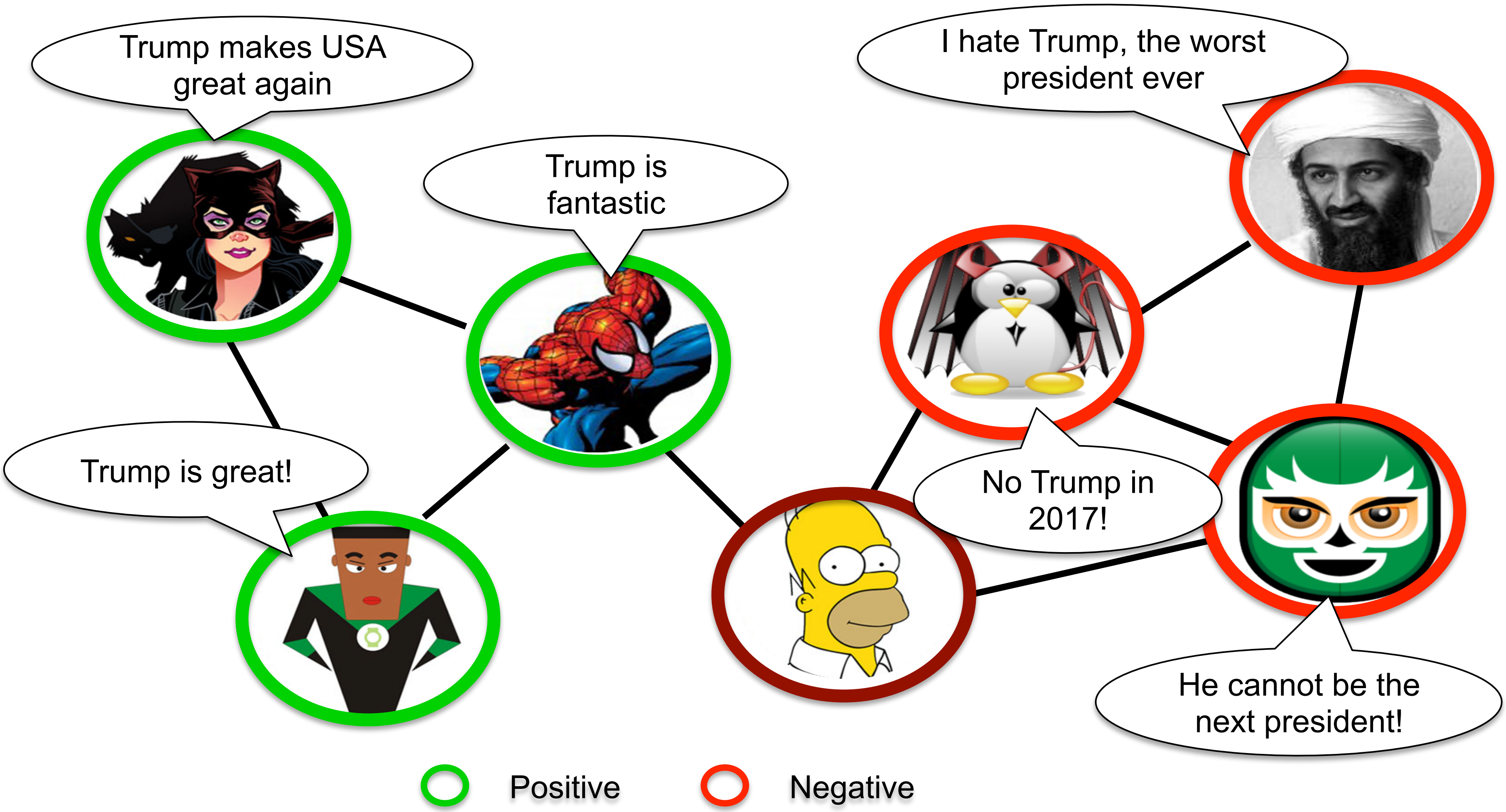
- How to obtain the weighted edges used in IC or LT models?

- How shall we learn the influence between two particular individuals?
  - Factors that affects social influence
    - Users' personal interests to a topic
    - Users' social roles

# How Does Personal Interest Affect Social Influence?

Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. **Social Influence Analysis in Large-scale Networks.** KDD 2009.

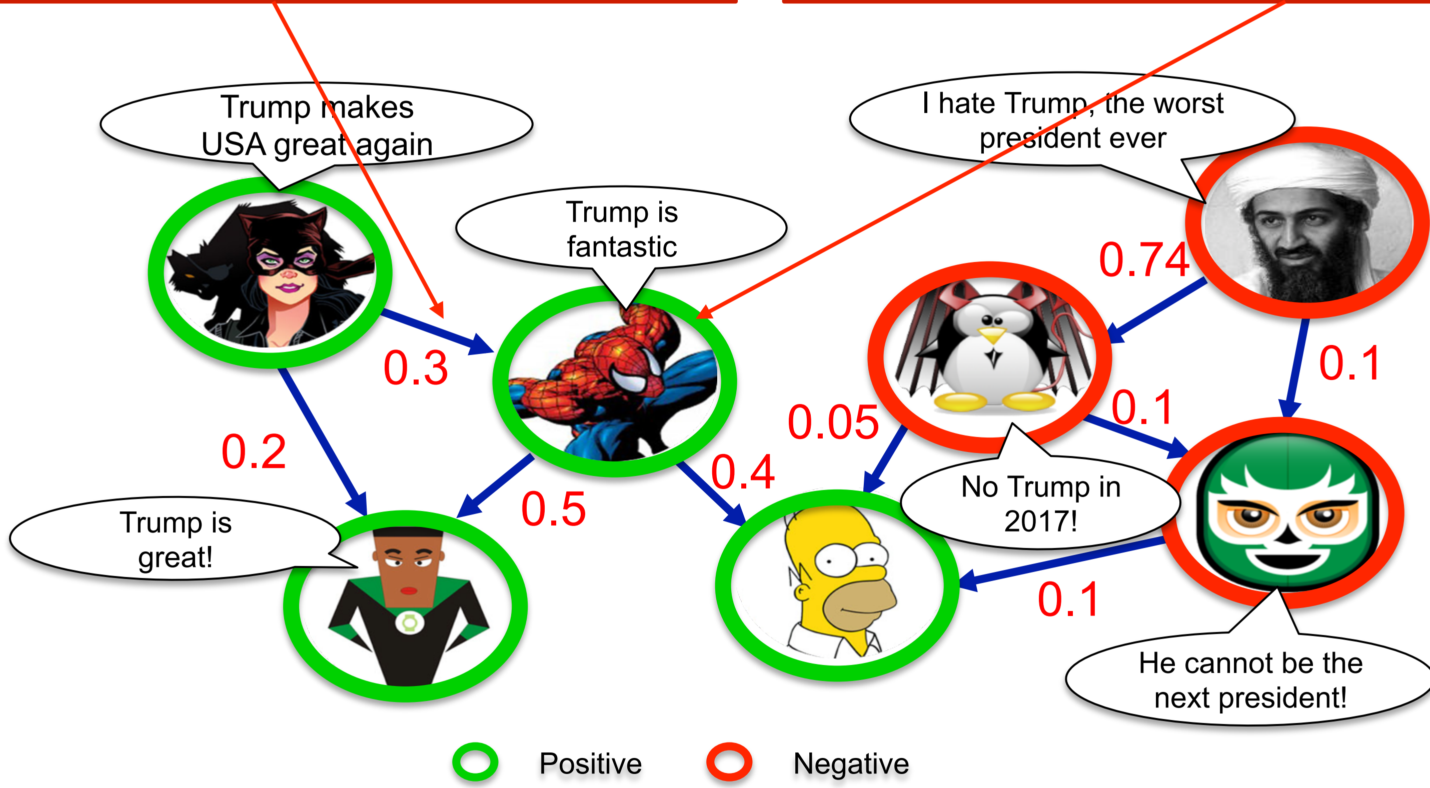
# User Opinion and Influence: “Love Trump”



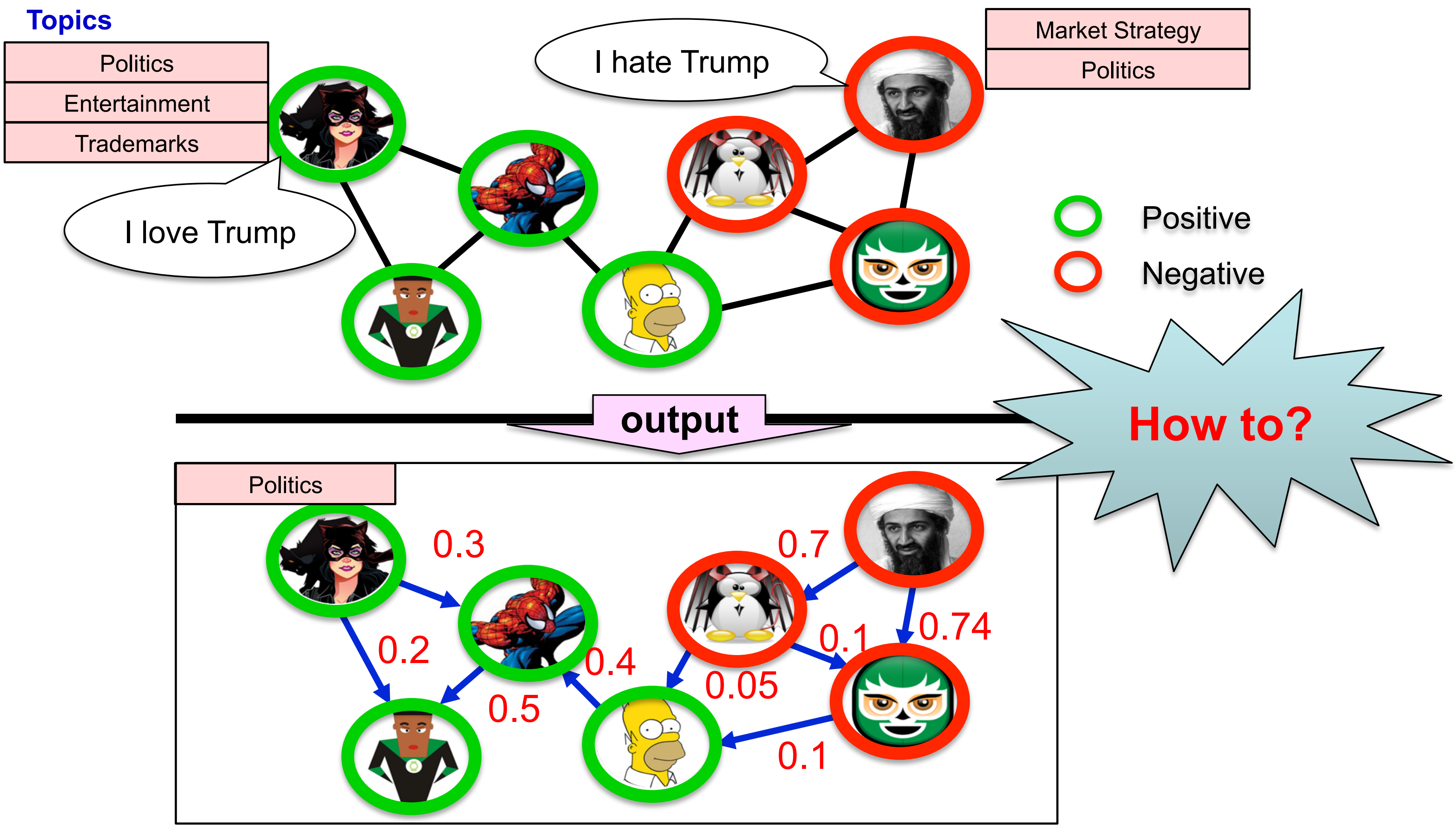
# Learn Multiple Aspect Social Influence

① Who influenced who? What is the **influence probability**?

② How to differentiate social influences from **multiple aspects**?

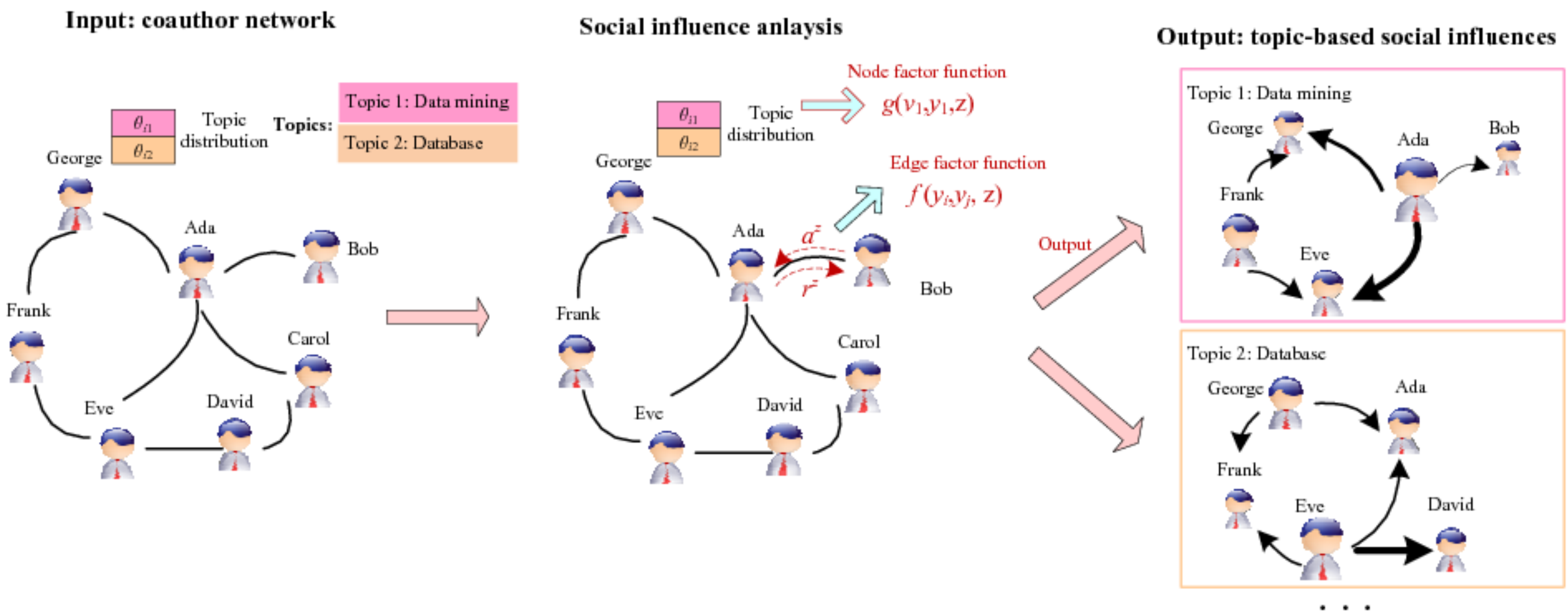


# Formulation: Learning Topic-based Social Influence

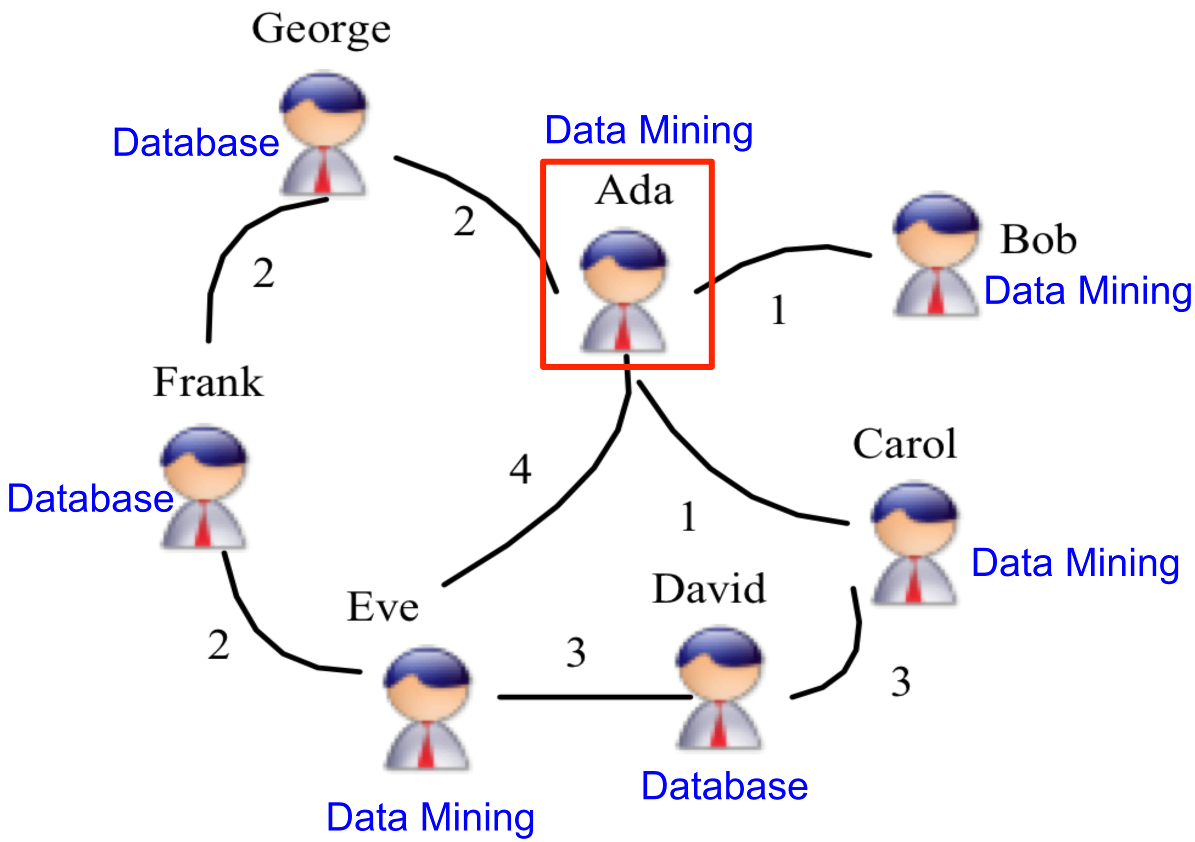


# Learning Topic-based Social Influence

- Social network -> Topical influence network



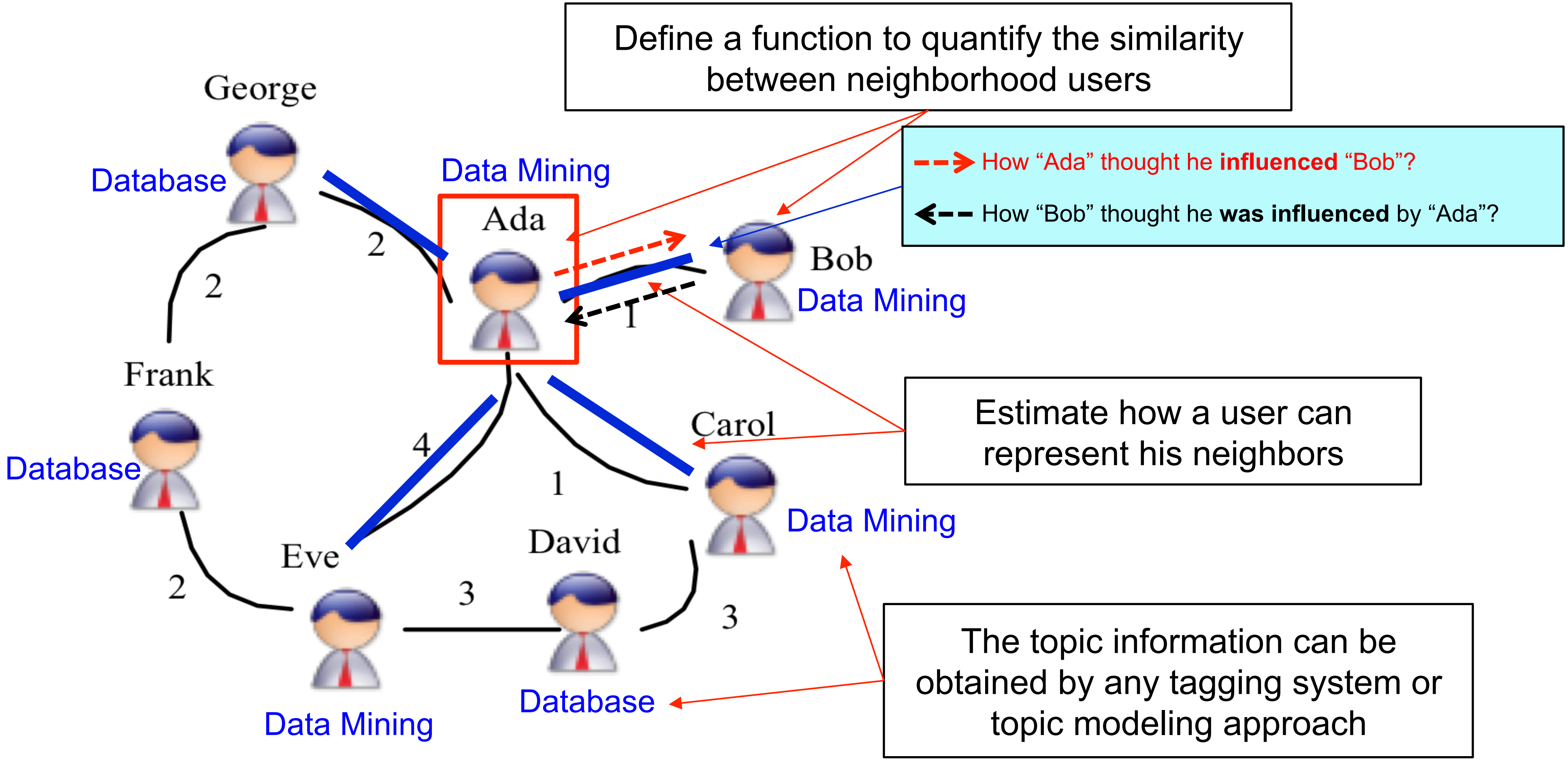
# The Solution: Topical Affinity Propagation




Basic Idea:
If a user is located in the center of a community, and is “similar” to the other users, then she/he would have a strong influence on the other users.
—Homophily theory

[1] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. **Social Influence Analysis in Large-scale Networks.** KDD 2009. (Top 10 cited paper among all papers published at KDD in the past 10 years)

# The Solution: Topical Affinity Propagation



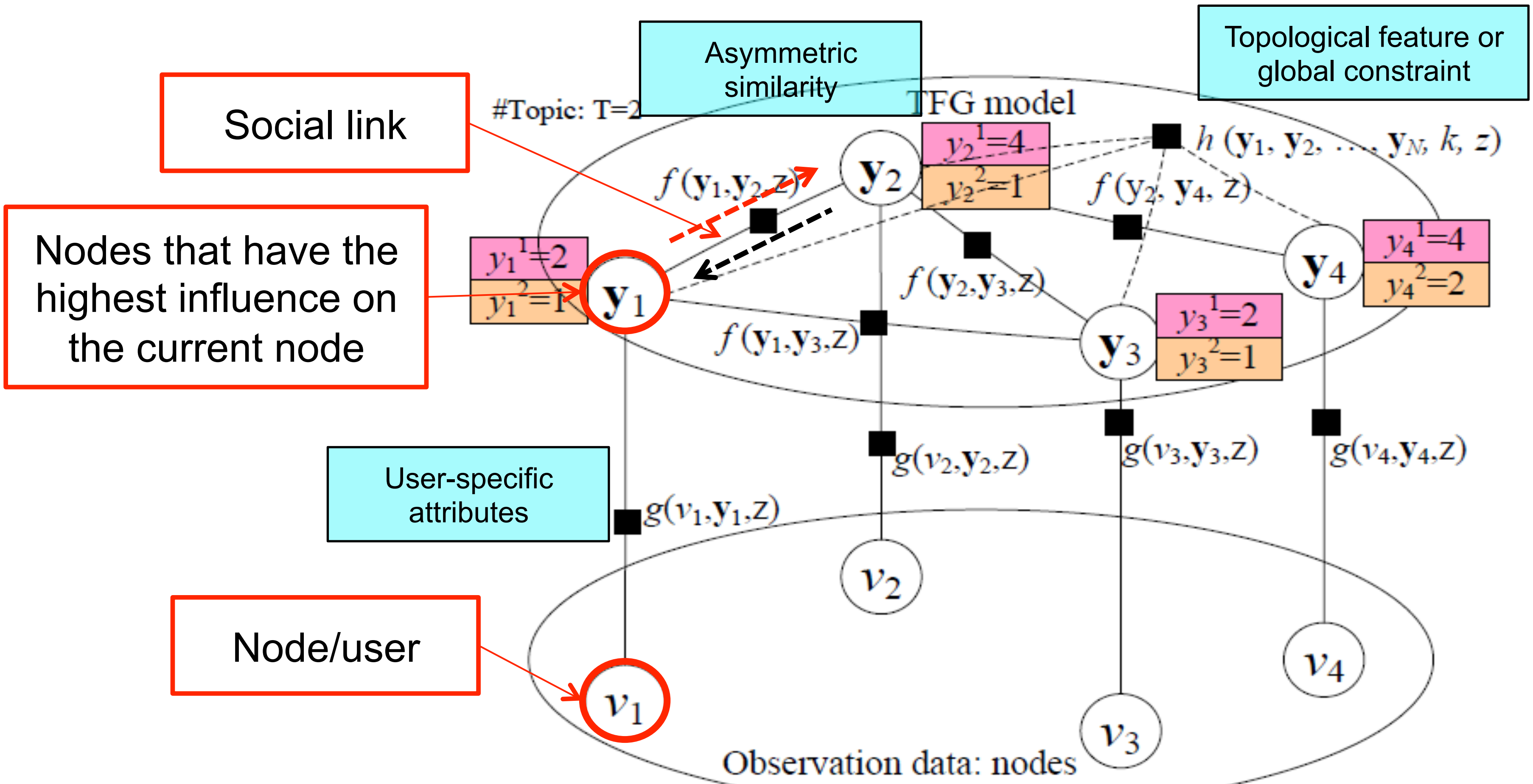

[1] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. **Social Influence Analysis in Large-scale Networks.** KDD 2009. (Top 10 cited paper among all papers published at KDD in the past 10 years)

# The Solution: Topical Affinity Propagation

- Topical Affinity Propagation
  - Topical Factor Graph model
  - Efficient learning algorithm
  - Distributed implementation

[1] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. **Social Influence Analysis in Large-scale Networks.** KDD 2009.
(Top 10 cited paper among all papers published at KDD in the past 10 years)

# Topical Factor Graph (TFG) Model




**The problem is cast as** identifying which node has the **highest probability** to **influence** another node on a **specific topic** along with the edge.

# Topical Factor Graph (TFG)

Objective function:

$$P(\mathbf{v}, \mathbf{Y}) = \frac{1}{Z} \prod_{k=1}^{N} \prod_{z=1}^{T} \boxed{h(\mathbf{y}_1, \ldots, \mathbf{y}_N, k, z)}$$

$$\prod_{i=1}^{N} \prod_{z=1}^{T} \boxed{g(v_i, \mathbf{y}_i, z)} \prod_{e_{kl} \in E} \prod_{z=1}^{T} \boxed{f(\mathbf{y}_k, \mathbf{y}_l, z)}$$

1. How to define?

2. How to optimize?

- The learning task is to find a configuration for all $\{y_i\}$ to maximize the joint probability.

# How to define (topical) feature functions?

Similarity: $w_{ij}^z = \theta_j^z \alpha_{ij}$

– Node feature function

$$g(v_i, \mathbf{y}_i, z) = \begin{cases} \dfrac{\boxed{w_{iy_i^z}^z}}{\sum_{j \in NB(i)} (w_{ij}^z + w_{ji}^z)} & y_i^z \neq i \\ \dfrac{\sum_{j \in NB(i)} w_{ji}^z}{\sum_{j \in NB(i)} (w_{ij}^z + w_{ji}^z)} & y_i^z = i \end{cases}$$

– Edge feature function

$$f(y_i, y_j) = \begin{cases} w[v_i \sim v_j] & y_i = y_j \\ 1 - w[v_i \sim v_j] & y_i \neq y_j \end{cases}$$

or simply binary

– Global feature function

$$h(\mathbf{y}_1, \ldots, \mathbf{y}_N, k, z) = \begin{cases} 0 & \text{if } y_k^z = k \text{ and } y_i^z \neq k \text{ for all } i \neq k \\ 1 & \text{otherwise.} \end{cases}$$

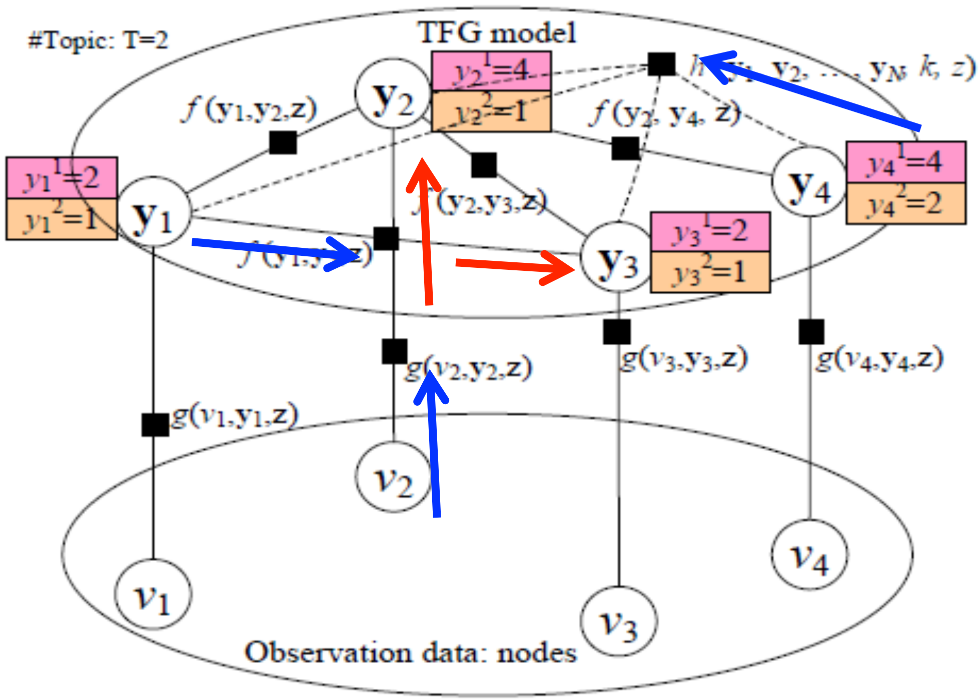
# Model Learning Algorithm

Sum-product:

$$\boxed{m_{y\to f}(y,z)} = \prod_{f'\sim y\setminus f} m_{f'\to y}(y,z) \prod_{z'\neq z} \prod_{f'\sim y\setminus f} m_{f'\to y}(y,z')^{(\tau_{z'z})}$$

$$\boxed{m_{f\to y}(y,z)} = \boxed{\sum_{\sim\{y\}}} \left( f(Y,z) \prod_{y'\sim f\setminus y} m_{y'\to f}(y',z) \right) + \sum_{z'\neq z} \tau_{z'z} \sum_{\sim\{y\}} \left( f(Y,z') \prod_{y'\sim f\setminus y} m_{y'\to f}(y',z') \right) \quad (4)$$

Marginal function for y on topic z




- Low efficiency!
- Not easy for distributed learning!

# New TAP Learning Algorithm

1. Introduce two new variables *r* and *a, to* replace the original message *m*.

2. Design new update rules:

How user *i* thought he **influenced** user *j*?

$m_{ij}$

$$r_{ij}^{z} = b_{ij}^{z} - \max_{k \in NB(j)} \{b_{ik}^{z} + a_{ik}^{z}\}$$

$$a_{jj}^{z} = \max_{k \in NB(j)} \min\{r_{kj}^{z}, 0\}$$

$$a_{ij}^{z} = \min(\max\{r_{jj}^{z}, 0\}, -\min\{r_{jj}^{z}, 0\} - \max_{k \in NB(j)\backslash\{i\}} \min\{r_{kj}^{z}, 0\}), i \in NB(j)$$

How user *j* thought he **was influenced** by user *i*?

# The TAP Learning Algorithm

**Input**: $G = (V, E)$ and topic distributions $\{\theta_v\}_{v \in V}$
**Output**: topic-level social influence graphs $\{G_z = (V_z, E_z)\}_{z=1}^{T}$

```
Calculate the node feature function g(v_i, y_i, z);
Calculate b^z_ij according to Eq. 8;
Initialize all {r^z_ij} ← 0;
repeat
    foreach edge-topic pair (e_ij, z) do
        Update r^z_ij according to Eq. 5;
    end
    foreach node-topic pair (v_j, z) do
        Update a^z_jj according to Eq. 6;
    end
    foreach edge-topic pair (e_ij, z) do
        Update a^z_ij according to Eq. 7;
    end
until convergence;
foreach node v_t do
    foreach neighboring node s ∈ NB(t) ∪ {t} do
        Compute μ^z_st according to Eq. 9;
    end
end
Generate G_z = (V_z, E_z) for every topic z according to {μ^z_st};
```

$$b_{ij}^z = \log \frac{g(v_i, \mathbf{y}_i, z)|_{y_i^z = j}}{\sum_{k \in NB(i) \cup \{i\}} g(v_i, \mathbf{y}_i, z)|_{y_i^z = k}}$$

$$r_{ij}^z = b_{ij}^z - \max_{k \in NB(j)} \{b_{ik}^z + a_{ik}^z\}$$

$$a_{jj}^z = \max_{k \in NB(j)} \min\{r_{kj}^z, 0\}$$

$$a_{ij}^z = \min(\max\{r_{jj}^z, 0\}, -\min\{r_{jj}^z, 0\} - \max_{k \in NB(j) \setminus \{i\}} \min\{r_{kj}^z, 0\}), i \in NB(j)$$

$$\mu_{st}^z = \frac{1}{1 + e^{-(r_{ts}^z + a_{ts}^z)}}$$

# Distributed TAP Learning

- Map-Reduce
  - Map: (key, value) pairs
    - $e_{ij}/a_{ij} \rightarrow e_{i*}/a_{ij}$; $e_{ij}/b_{ij} \rightarrow e_{i*}/b_{ij}$; $e_{ij}/r_{ij} \rightarrow e_{*j}/r_{ij}$ .
  - Reduce: (key, value) pairs
    - $e_{ij}$ / * $\rightarrow$ new $r_{ij}$; $e_{ij}$/* $\rightarrow$ new $a_{ij}$
- For the global feature function

THEOREM 1. *If the global feature function $h$ can be factorized into $h = \prod_{k=1}^{N} h_k$, for every $i \in \{1, \ldots, N\}, y_i \neq k, y_i' \neq k, h_k(y_1, \ldots, y_i, \ldots, y_N) = h_k(y_1, \ldots, y_i', \ldots, y_N)$, then the message passing update rules can be simplified to influence update rules.* ■

# Experiment

- Data set: (ArnetMiner.org and Wikipedia)
  - **Coauthor** dataset： 640,134 authors and 1,554,643 coauthor relations
  - **Citation** dataset: 2,329,760 papers and 12,710,347 citations between these papers
  - **Film** dataset: 18,518 films, 7,211 directors, 10,128 actors, and 9,784 writers
- Evaluation measures
  - Case study
  - CPU time
  - Application

# Influential nodes on different topics

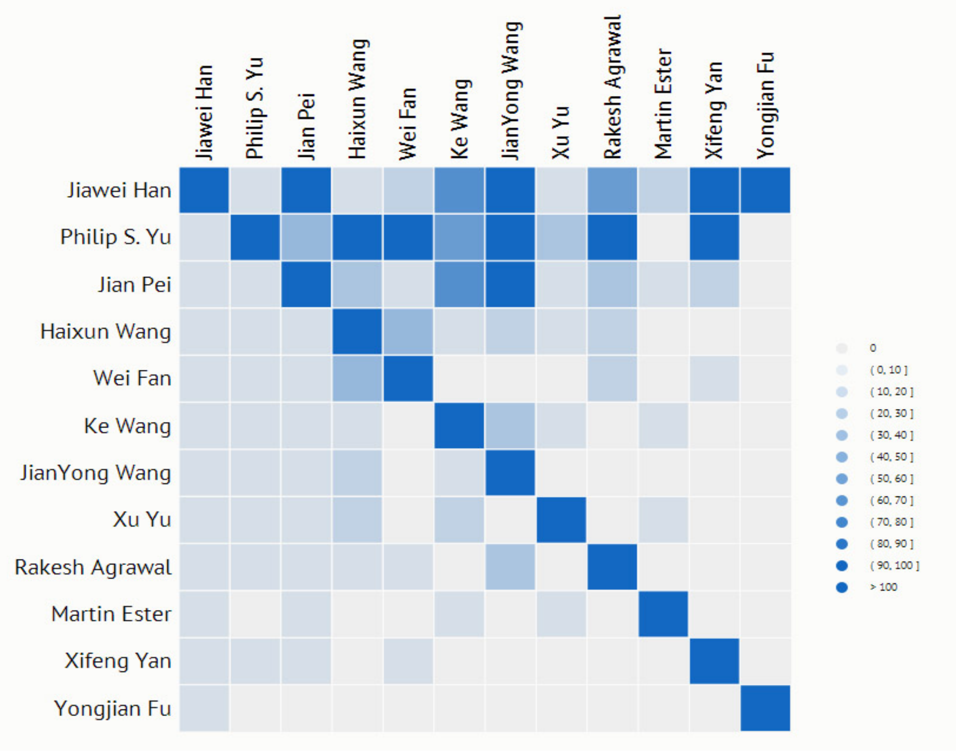
| Dataset | Topic | Representative Nodes |
|---|---|---|
| Author | Data Mining | Heikki Mannila, Philip S. Yu, Dimitrios Gunopulos, Jiawei Han, Christos Faloutsos, Bing Liu, Vipin Kumar, Tom M. Mitchell, Wei Wang, Qiang Yang, Xindong Wu, Jeffrey Xu Yu, Osmar R. Zaiane |
| | Machine Learning | Pat Langley, Alex Waibel, Trevor Darrell, C. Lee Giles, Terrence J. Sejnowski, Samy Bengio, Daphne Koller, Luc De Raedt, Vasant Honavar, Floriana Esposito, Bernhard Scholkopf |
| | Database System | Gerhard Weikum, John Mylopoulos, Michael Stonebraker, Barbara Pernici, Philip S. Yu, Sharad Mehrotra, Wei Sun, V. S. Subrahmanian, Alejandro P. Buchmann, Kian-Lee Tan, Jiawei Han |
| | Information Retrieval | Gerard Salton, W. Bruce Croft, Ricardo A. Baeza-Yates, James Allan, Yi Zhang, Mounia Lalmas, Zheng Chen, Ophir Frieder, Alan F. Smeaton, Rong Jin |
| | Web Services | Yan Wang, Liang-jie Zhang, Schahram Dustdar, Jian Yang, Fabio Casati, Wei Xu, Zakaria Maamar, Ying Li, Xin Zhang, Boualem Benatallah, Boualem Benatallah |
| | Semantic Web | Wolfgang Nejdl, Daniel Schwabe, Steffen Staab, Mark A. Musen, Andrew Tomkins, Juliana Freire, Carole A. Goble, James A. Hendler, Rudi Studer, Enrico Motta |
| | Bayesian Network | Daphne Koller, Paul R. Cohen, Floriana Esposito, Henri Prade, Michael I. Jordan, Didier Dubois, David Heckerman, Philippe Smets |
| Citation | Data Mining | Fast Algorithms for Mining Association Rules in Large Databases, Using Segmented Right-Deep Trees for the Execution of Pipelined Hash Joins, Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, Discovery of Multiple-Level Association Rules from Large Databases, Interleaving a Join Sequence with Semijoins in Distributed Query Processing |
| | Machine Learning | Object Recognition with Gradient-Based Learning, Correctness of Local Probability Propagation in Graphical Models with Loops, A Learning Theorem for Networks at Detailed Stochastic Equilibrium, The Power of Amnesia: Learning Probabilistic Automata with Variable Memory Length, A Unifying Review of Linear Gaussian Models |
| | Database System | Mediators in the Architecture of Future Information Systems, Database Techniques for the World-Wide Web: A Survey, The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles, Fast Algorithms for Mining Association Rules in Large Databases |
| | Web Services | The Web Service Modeling Framework WSMF, Interval Timed Coloured Petri Nets and their Analysis, The design and implementation of real-time schedulers in RED-linux, The Self-Serv Environment for Web Services Composition |
| | Web Mining | Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, Fast Algorithms for Mining Association Rules in Large Databases, The OO-Binary Relationship Model: A Truly Object Oriented Conceptual Model, Distributions of Surfers' Paths Through the World Wide Web: Empirical Characterizations, Improving Fault Tolerance and Supporting Partial Writes in Structured Coterie Protocols for Replicated Objects |
| | Semantic Web | FaCT and iFaCT, The GRAIL concept modelling language for medical terminology, Semantic Integration of Semistructured and Structured Data Sources, Description of the RACER System and its Applications, DL-Lite: Practical Reasoning for Rich Dls |

# Social Influence Sub-graph on "Data mining"

Table 4: Dynamic influence analysis for Dr. Jian Pei during 2000-2009. Due to space limitation, we only list coauthors who most influence on/by Dr. Pei in each time window.

| Year | Pairwise | Influence |
|---|---|---|
| 2000 - 2001 | Influence on Dr. Pei | Jiawei Han (0.4961) |
| | Influenced by Dr. Pei | Jiawei Han (0.0082) |
| 2002 - 2003 | Influence on Dr. Pei | Jiawei Han (0.4045), Ke Wang (0.0418), Jianyong Wang (0.019), Xifeng Yan (0.007), Shiwei Tang (0.0052) |
| | Influenced by Dr. Pei | Shiwei Tang (0.436), Hasan M.Jamil (0.4289), Xifeng Yan (0.2192), Jianyong Wang (0.1667), Ke Wang (0.0687) |
| 2004 - 2005 | Influence on Dr. Pei | Jiawei Han (0.2364), Ke Wang (0.0328), Wei Wang (0.0294), Jianyong Wang (0.0248), Philip S. Yu (0.0156) |
| | Influenced by Dr. Pei | Chun Tang (0.5929), Shiwei Tang (0.5426), Hasan M.Jamil (0.3318), Jianyong Wang (0.1609), Xifeng Yan (0.1458), Yan Huang (0.1054) |
| 2006 - 2007 | Influence on Dr. Pei | Jiawei Han (0.1201), Ke Wang (0.0351), Wei Wang (0.0226), Jianyong Wang (0.018), Ada Wai-Chee Fu (0.0125) |
| | Influenced by Jian Pei | Chun Tang (0.6095), Shiwei Tang (0.6067), Byung-Won On (0.4599), Hasan M.Jamil (0.3433), Jaewoo Kang (0.3386) |
| 2008 - 2009 | Influence on Dr. Pei | Jiawei Han (0.2202), Ke Wang (0.0234), Ada Wai-Chee Fu (0.0208), Wei Wang (0.011), Jianyong Wang (0.0095) |
| | Influenced by Dr. Pei | ZhaoHui Tang (0.654), Chun Tang (0.6494), Shiwei Tang (0.5923), Zhengzheng Xing (0.5549), Hasan M.Jamil (0.3333), Jaewoo Kang (0.3057) |

On "Data Mining" in 2009



# Scalability Performance

**Table 2: Scalability performance of different methods on real data sets. >10hr means that the algorithm did not terminate when the algorithm runs more than 10 hours.**

| Methods | Citation | Coauthor | Film |
|---|---|---|---|
| Sum-Product | N/A | >10hr | 1.8 hr |
| Basic TAP Learning | >10hr | 369s | **57s** |
| Distributed TAP Learning | **39.33m** | **104s** | 148s |

# Application—Expert Finding

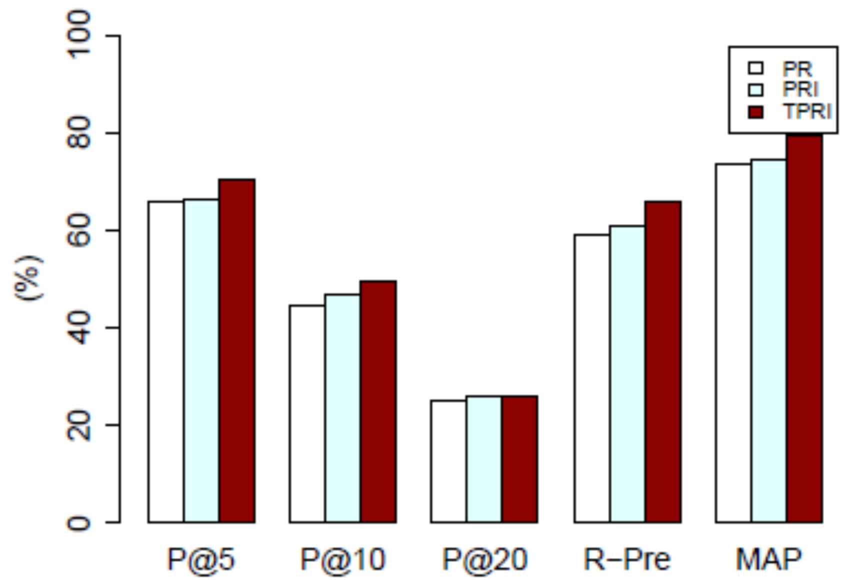



**Table 7: Performance of expert finding with different approaches.**

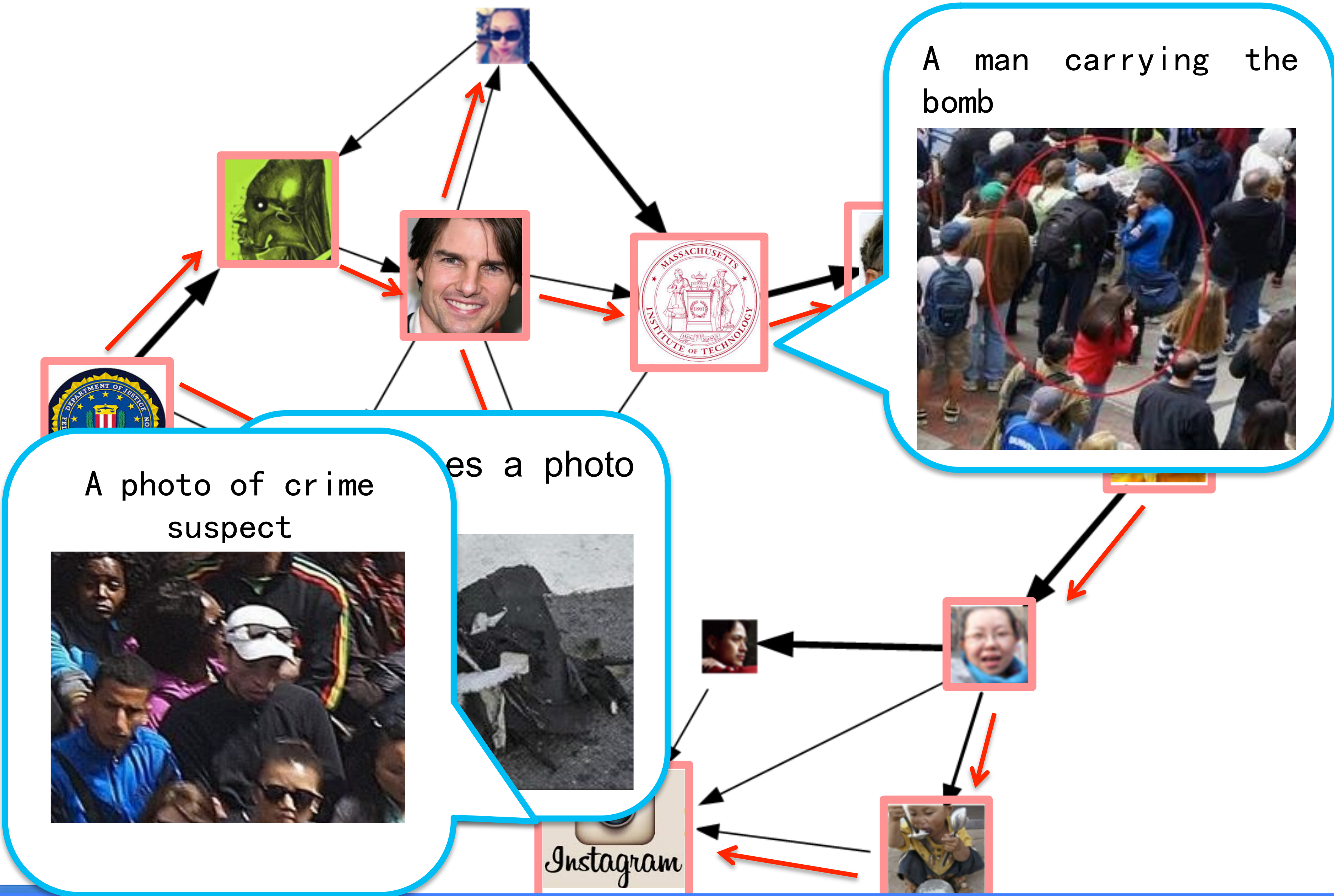
Expert finding data from (Tang, KDD08; ICDM08)

http://arnetminer.org/lab-datasets/expertfinding/

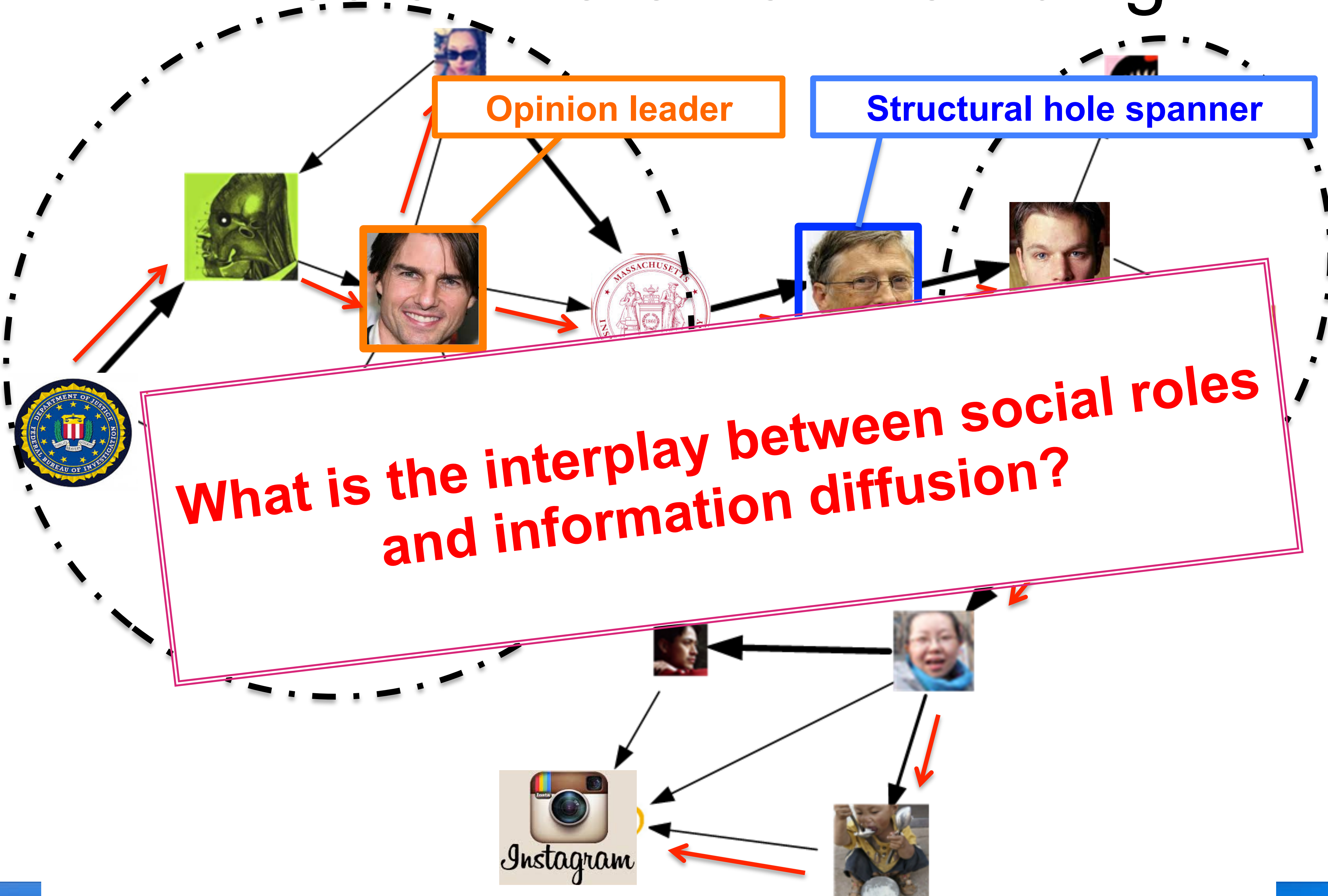
# Information Diffusion

- Information diffusion, also known as diffusion of innovations, is the study of how information propagates in or between networks.

# Boston Marathon Bombing



# Boston Marathon Bombing

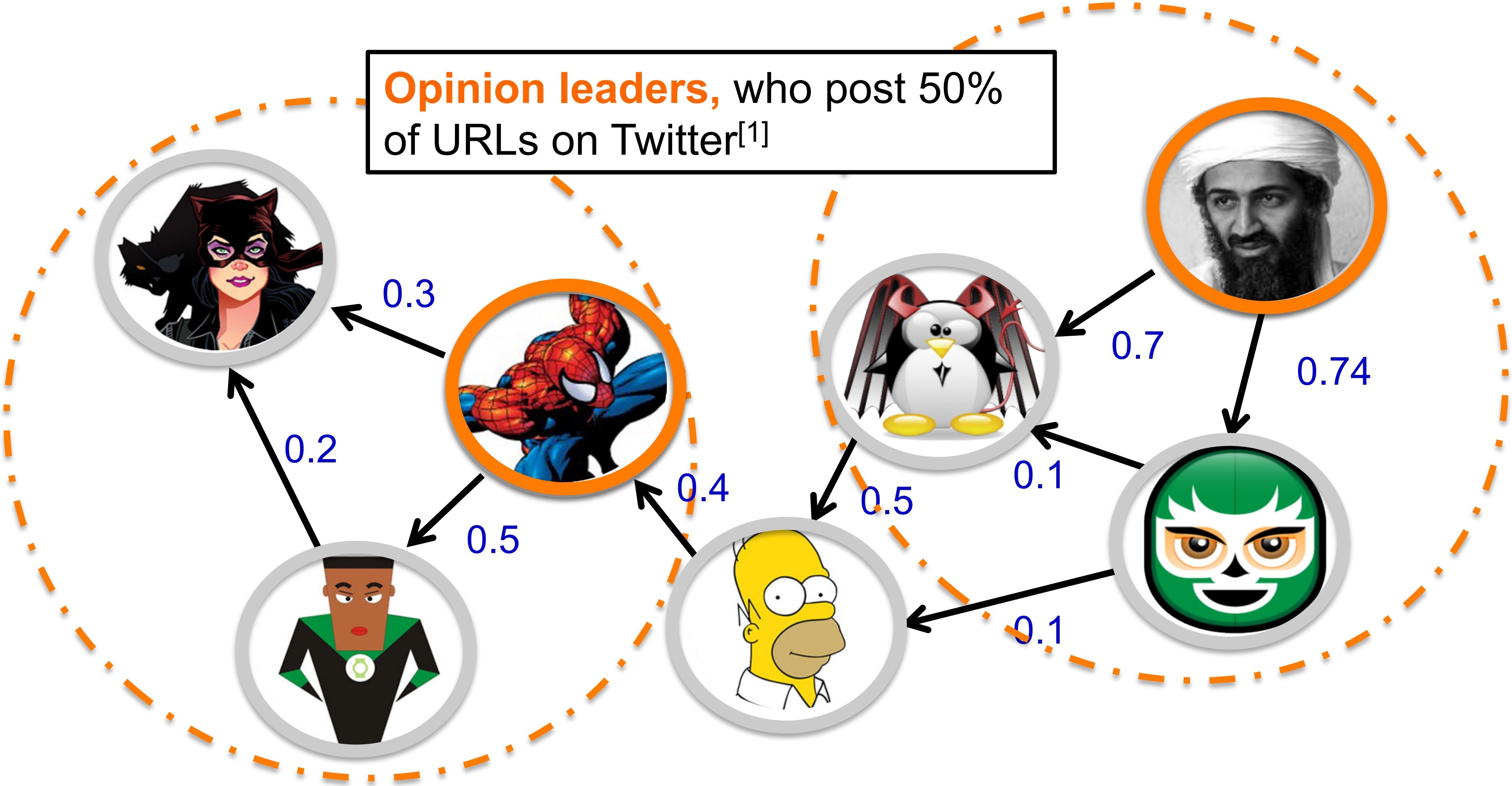


# Social-Role aware Information Diffusion

Yang Yang, Jie Tang, Cane Wing-Ki Leung, Yizhou Sun, Qicong Chen, Juanzi Li, and Qiang Yang. **RAIN: Social Role-Aware Information Diffusion.** AAAI'15, 2015.

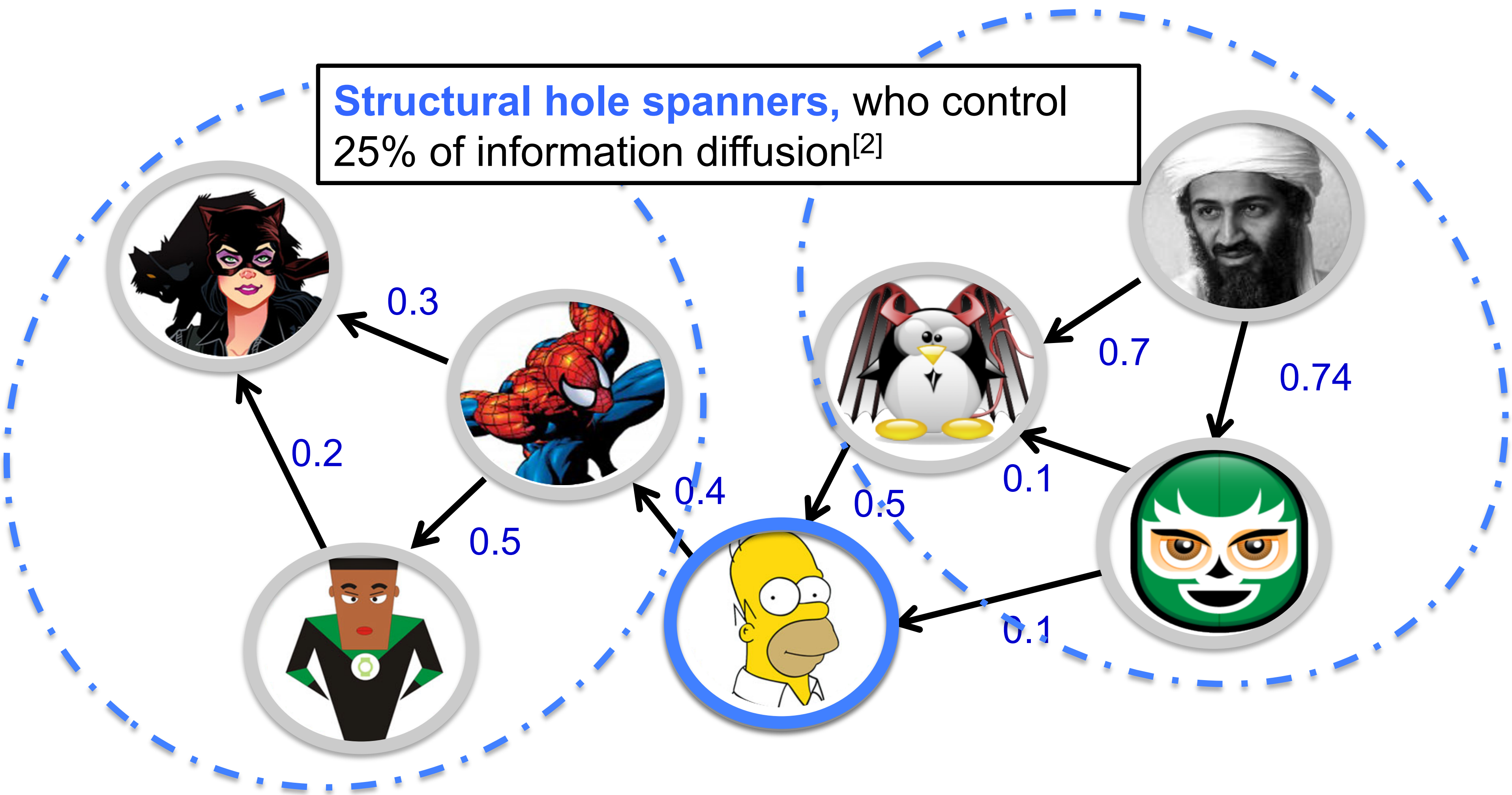
# Social Roles

**Opinion leaders,** who post 50% of URLs on Twitter[1]




[1] Wu, S.; Hofman, J. M.; Mason,W. A.; andWatts, D. J. 2011. Who says what to whom on twitter. In WWW'11 , 705–714.

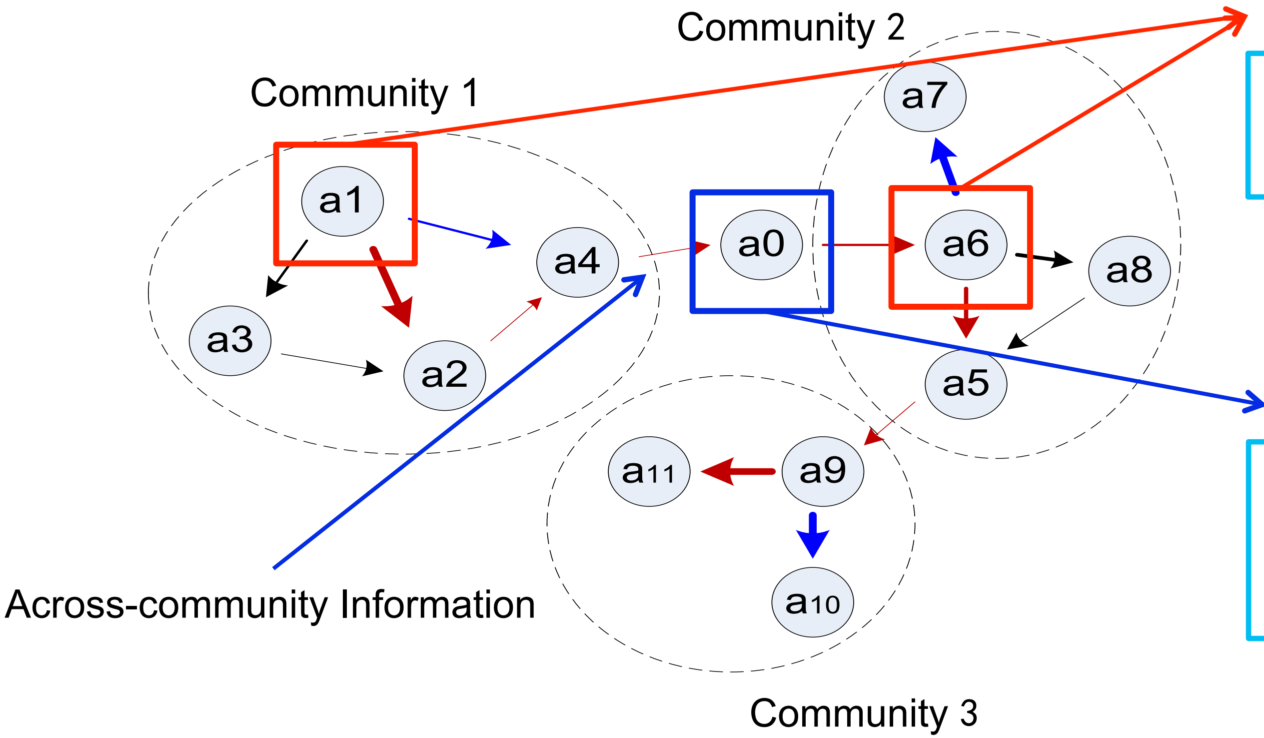
# Social Role




[2] Lou, T., and Tang, J. 2013. Mining structural hole spanners through information diffusion in social networks. In WWW'13 , 825–836

# Social Roles

>0.16 billion users
>0.17 billion posts
Complete data sets during Oct. 1st – Oct. 7$^{th}$, 2012.



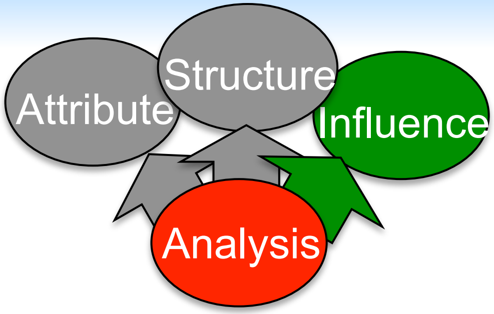

**Opinion Leader**

1% of users with largest PageRank scores

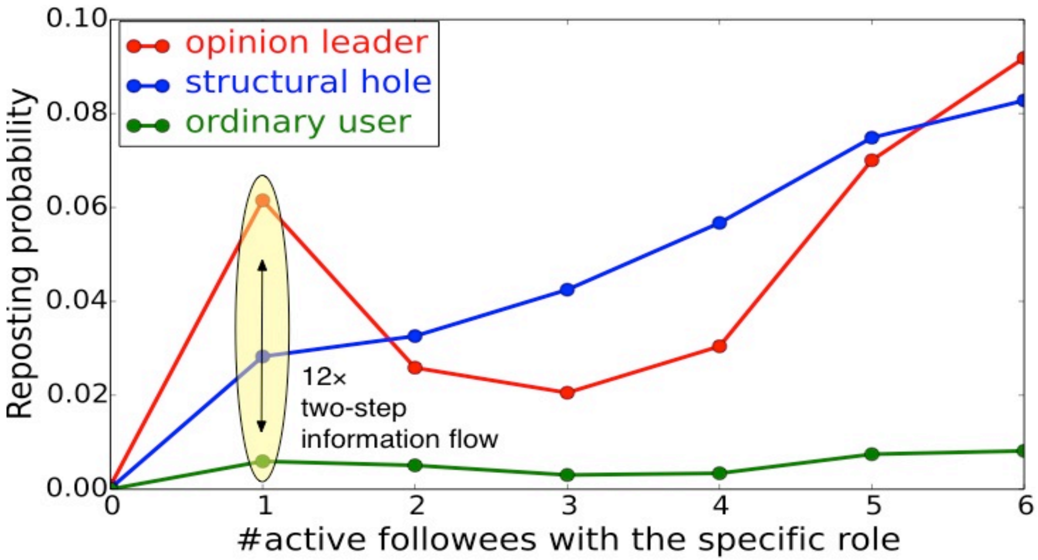
**Structural Hole**

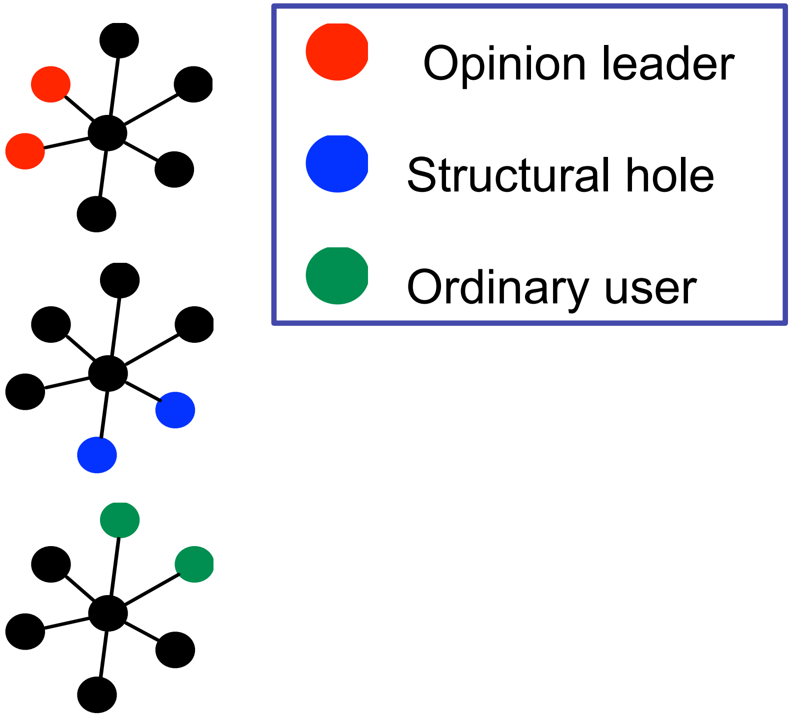
1% of users with smallest network constraint scores.


[1] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts. Who says what to whom on twitter. In **WWW'11**, pages 705–714, 2011.
[2] T. Lou and J. Tang. Mining Structural Hole Spanners Through Information Diffusion in Social Networks. In **WWW'13**. pp. 837-848.




# Influence Strength

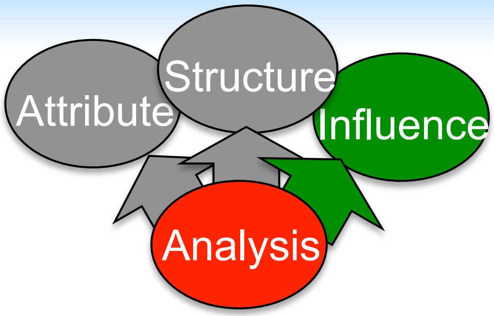


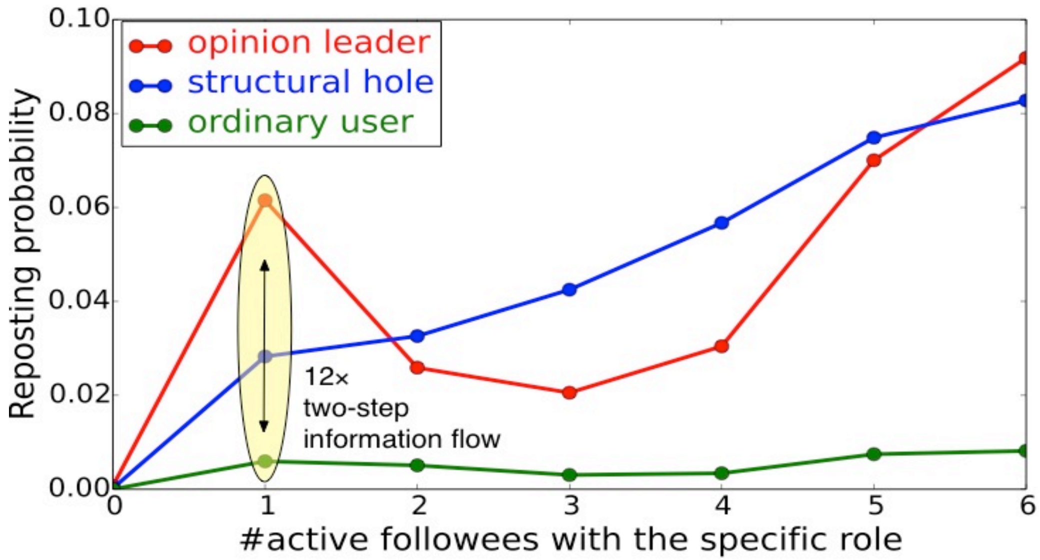

Opinion leader:

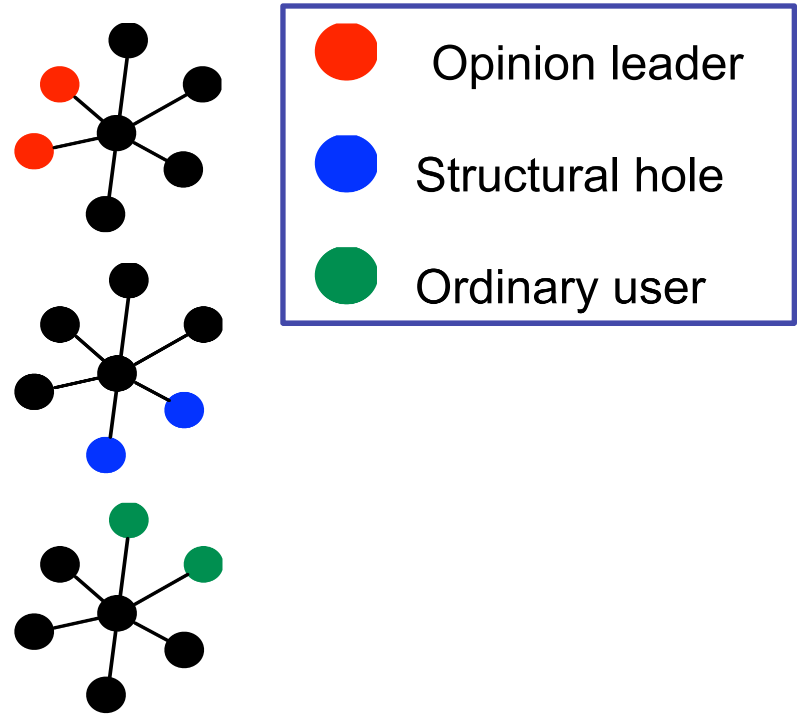
- Stage 1 - activation probability is12 times higher than ordinary user
- Stage 2 - information overload[1]: 2-3 opinion leaders are sufficient to spread a piece of information throughout a community.
- Stage 3 - information everywhere: spreading the information becomes a social norm to adopt.

[1] Lazarsfeld, P. F.; Berelson, B.; and Gaudet, H. 1944. The peoples choice: How the voter makes up his mind in a presidential election. New York: Duell, Sloan and Pearce .




# Influence Strength

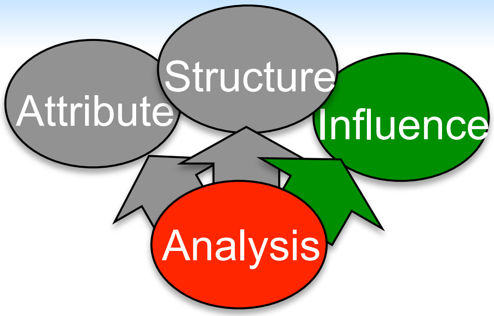


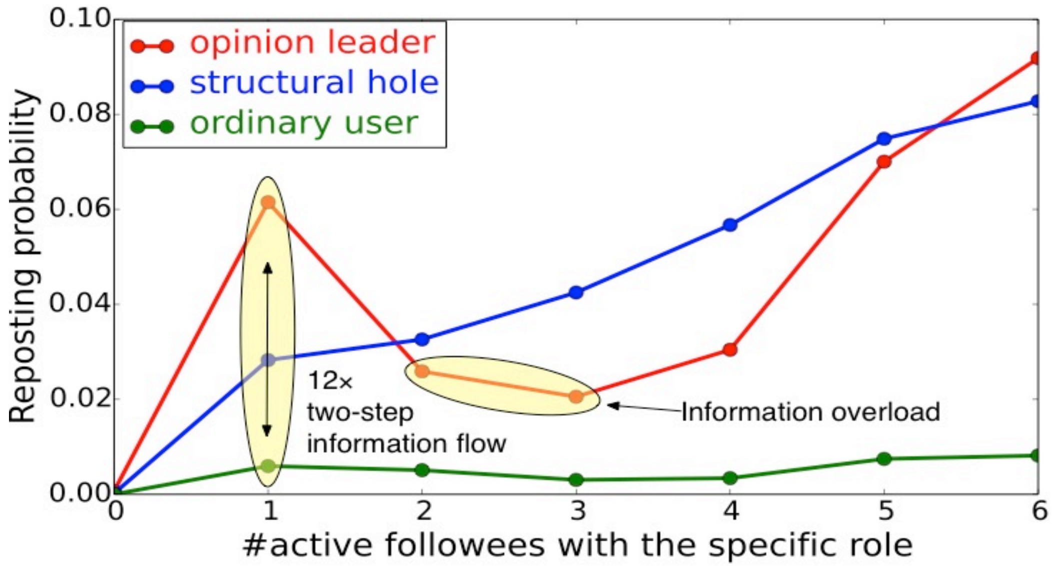

Opinion leader:

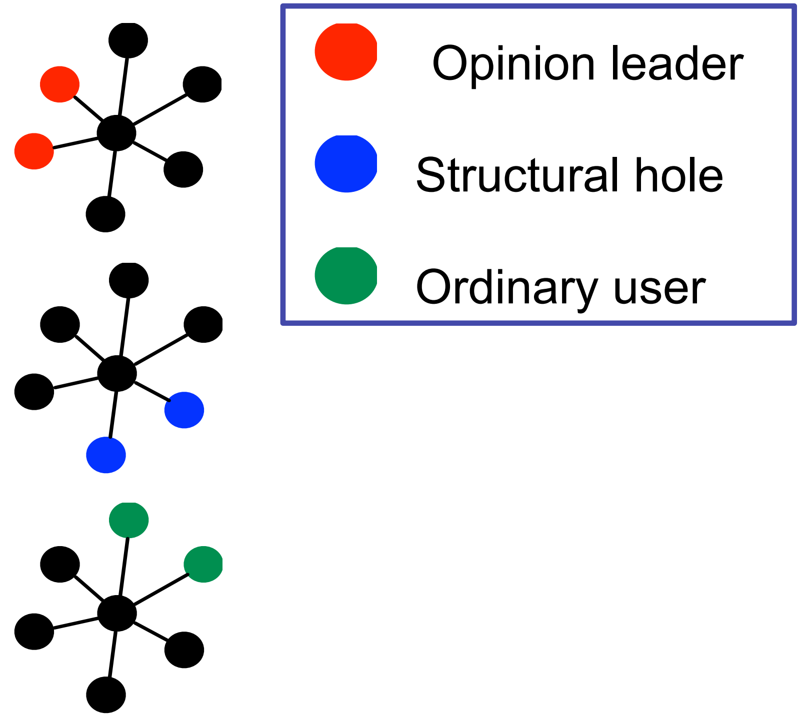
- Stage 1 - activation probability is 12 times higher than ordinary user
- Stage 2 - information overload[1]: 2-3 opinion leaders are sufficient to spread a piece of information throughout a community.
- Stage 3 - information everywhere: spreading the information becomes a social norm to adopt.

[1] Lazarsfeld, P. F.; Berelson, B.; and Gaudet, H. 1944. The peoples choice: How the voter makes up his mind in a presidential election. New York: Duell, Sloan and Pearce .




# Influence Strength

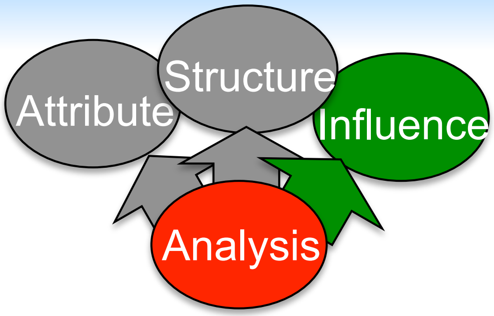


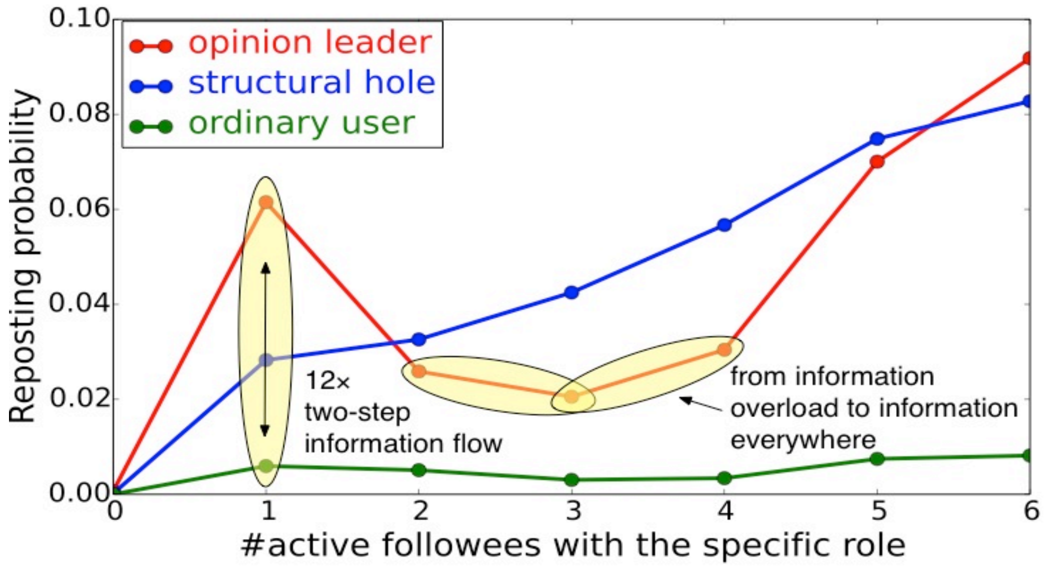

Opinion leader:

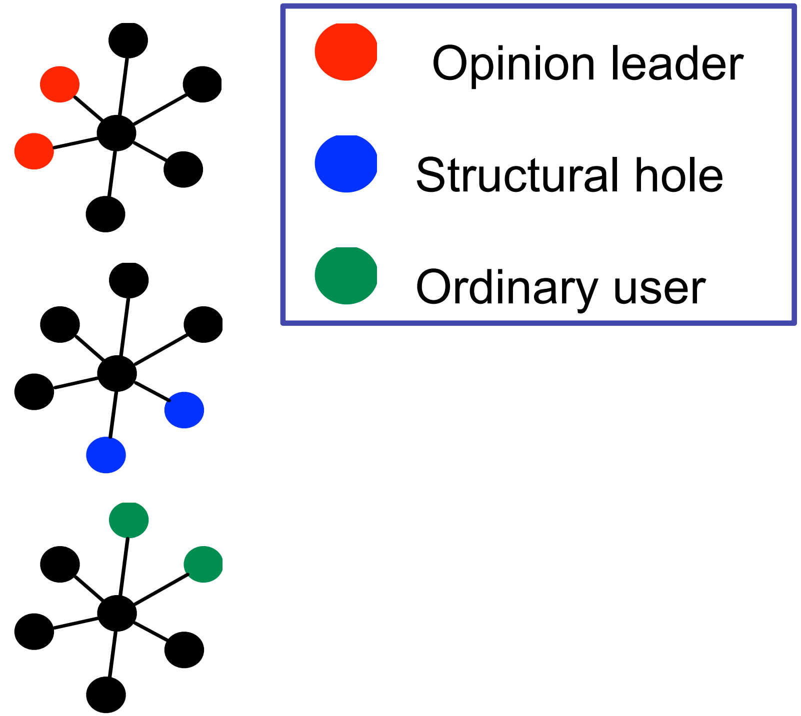
- Stage 1 - activation probability is 12 times higher than ordinary user
- Stage 2 - information overload[1]: 2-3 opinion leaders are sufficient to spread a piece of information throughout a community.
- Stage 3 - information everywhere: spreading the information becomes a social norm to adopt.

[1] Lazarsfeld, P. F.; Berelson, B.; and Gaudet, H. 1944. The peoples choice: How the voter makes up his mind in a presidential election. New York: Duell, Sloan and Pearce .




# Influence Strength

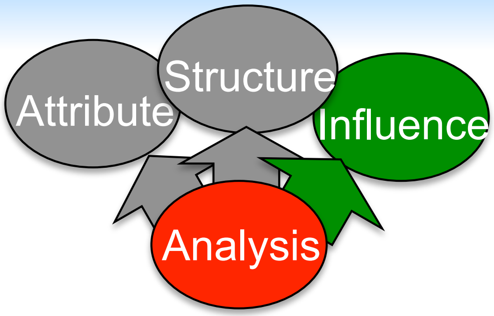


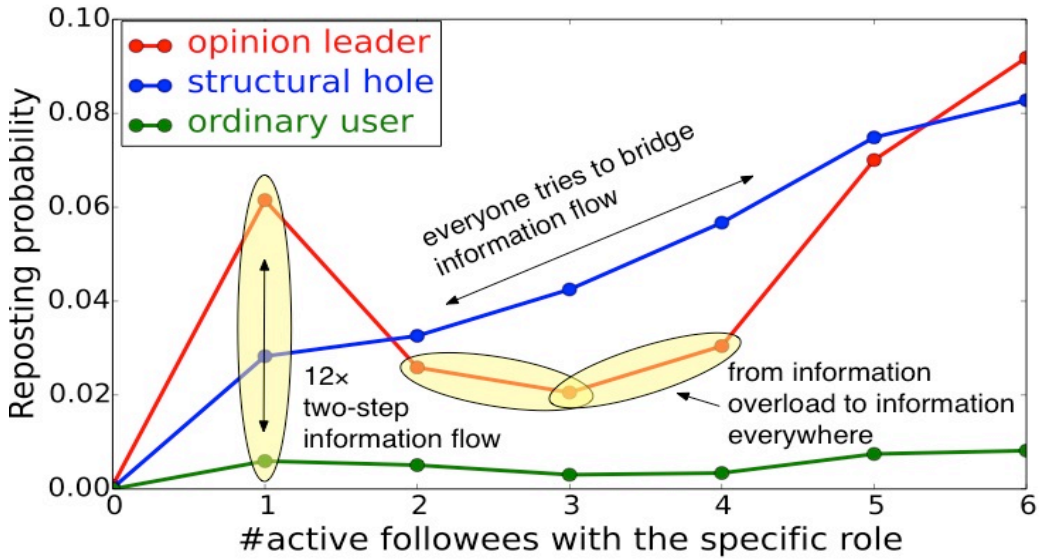

Opinion leader:

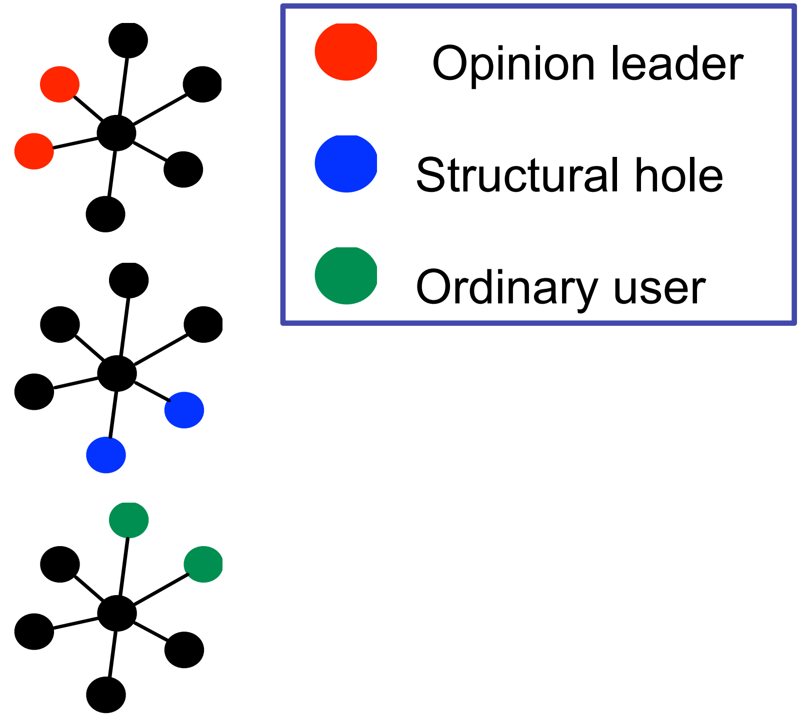
- Stage 1 - activation probability is 12 times higher than ordinary user
- Stage 2 - information overload[1]: 2-3 opinion leaders are sufficient to spread a piece of information throughout a community.
- Stage 3 - information everywhere: spreading the information becomes a social norm to adopt.

[1] Lazarsfeld, P. F.; Berelson, B.; and Gaudet, H. 1944. The peoples choice: How the voter makes up his mind in a presidential election. New York: Duell, Sloan and Pearce .




# Influence Strength

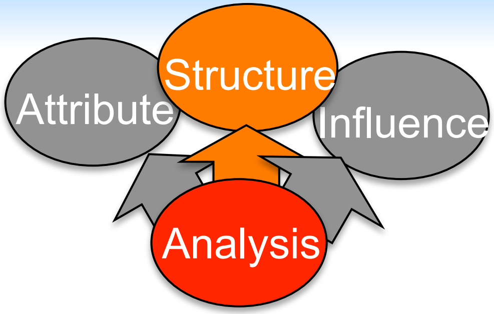


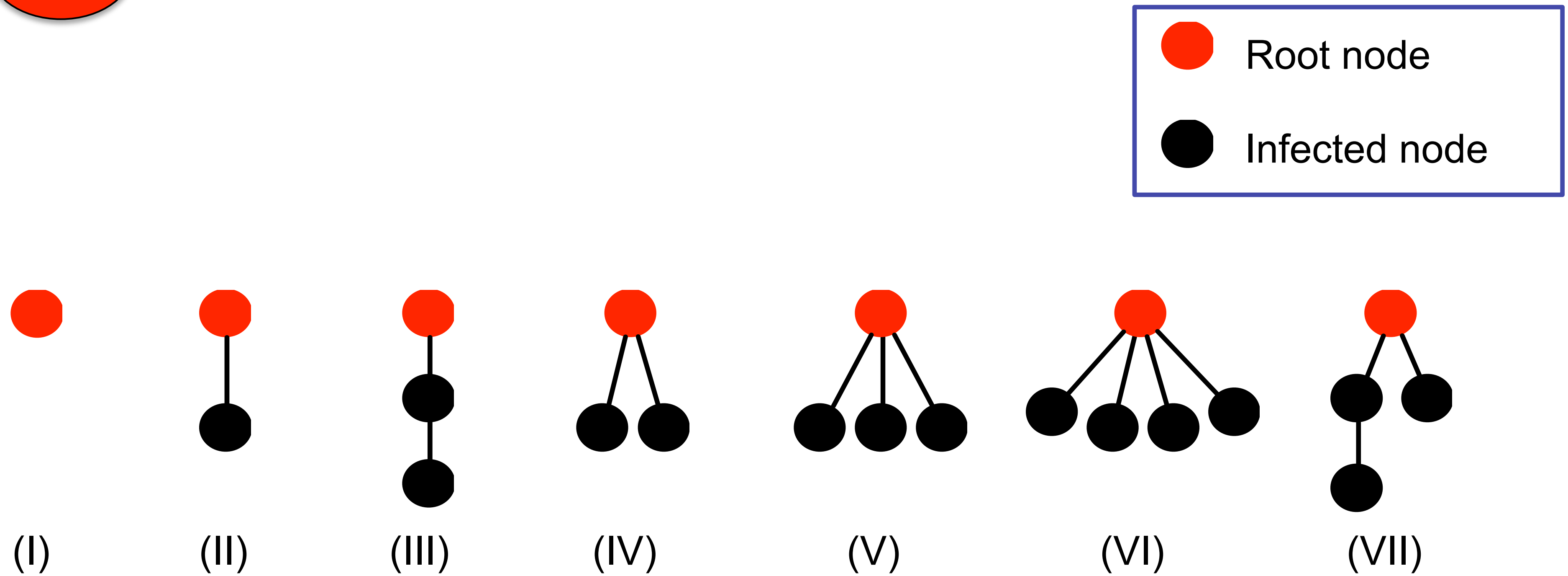

Structural hole spanners[2][3]:

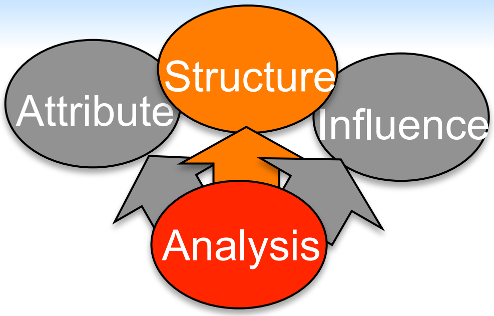
- SH tend to bring information that a certain community is rarely exposed to.
- Most users tries to bridge information flow between different groups.

[2] Burt, R. S. 2001. Structural holes versus network closure as social capital. Social capital: Theory and research 31–56.
[3] Burt, R. S. 2009. Structural holes: The social structure of competition . Harvard University Press.

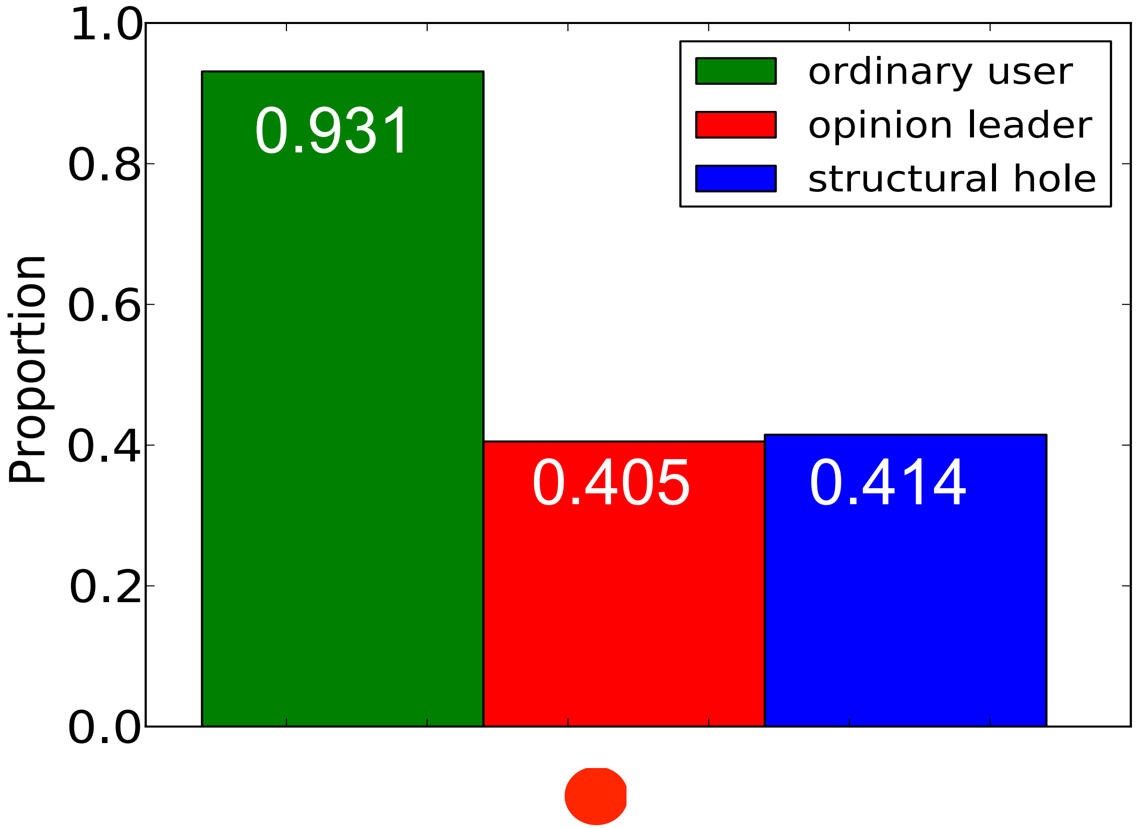



# Atomic Diffusion Structure

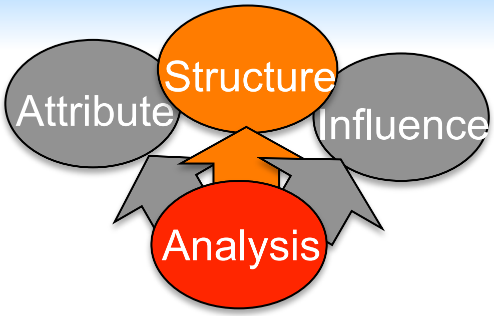




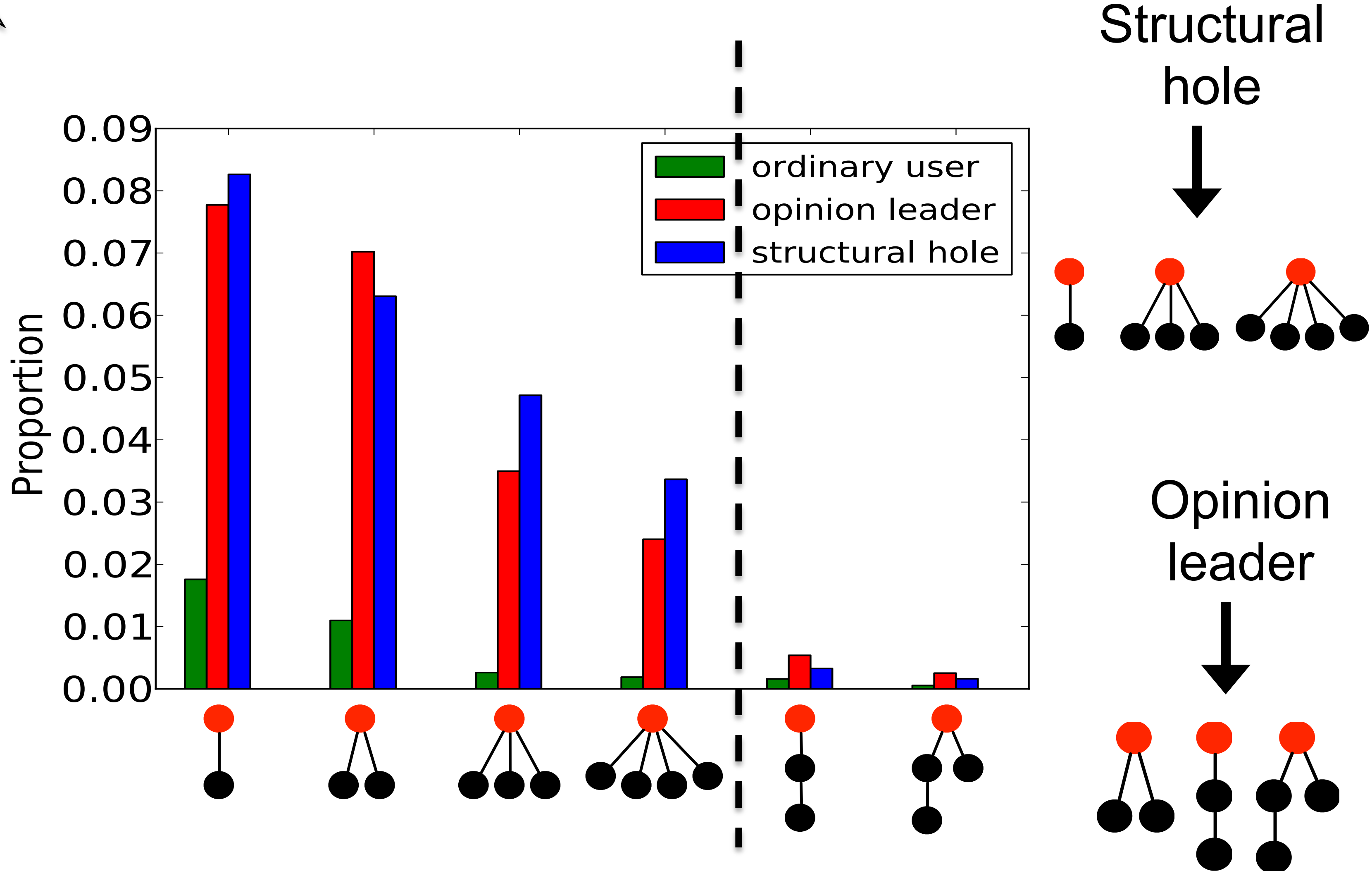

# Atomic Diffusion Structure



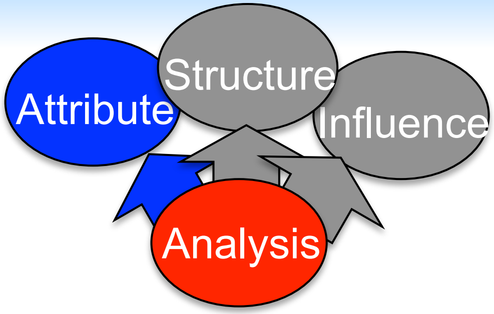

(I)



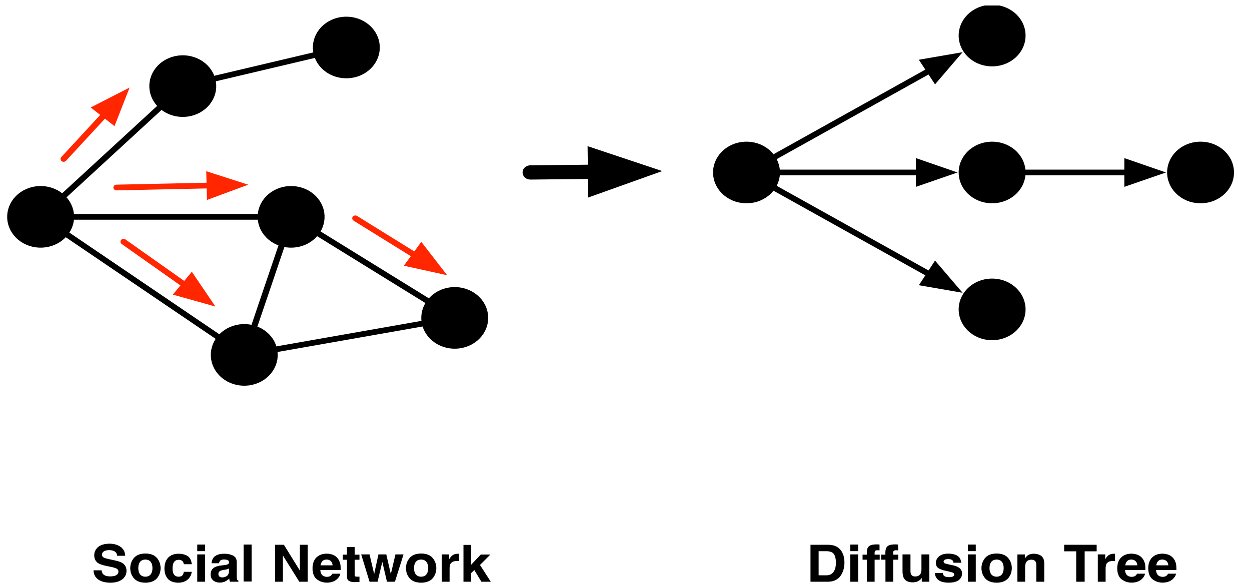

# Atomic Diffusion Structure



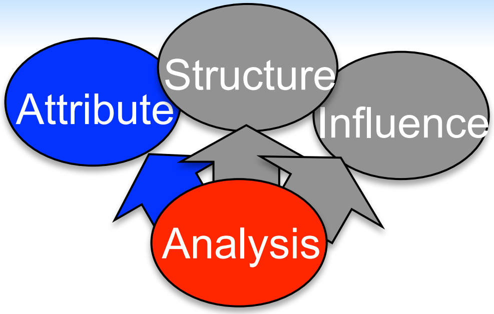

**Diffusion structures tend to be wide, and not too deep**



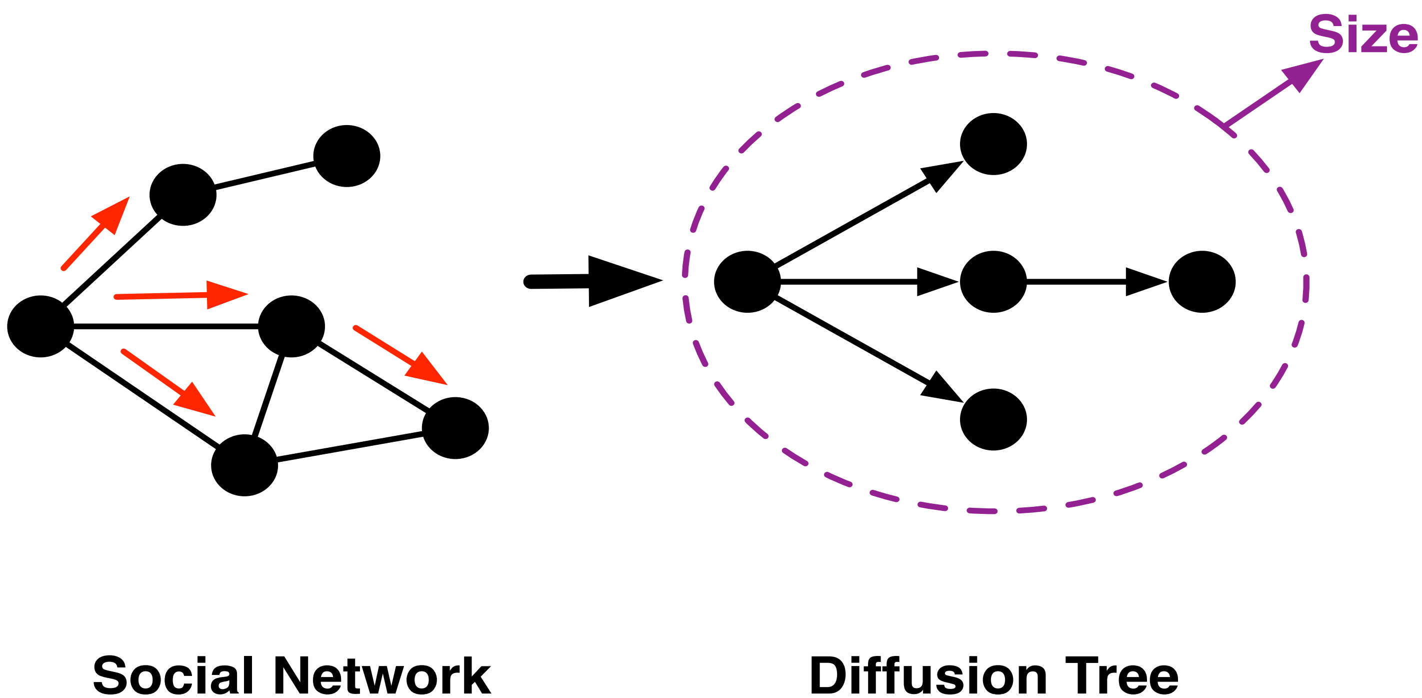

# Formulation



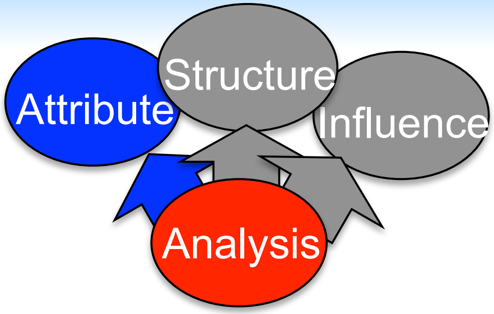

*Definition 1.* **Diffusion Tree.** In a given $G$, a diffusion tree of a message $i$ comprises a set of 4-tuples: $\{(v', v, i, t)\}$, where each tuple $(v', v, i, t)$ indicates that user $v$ retweeted $i$ from $v'$ at time $t$. In a given tuple, $v' = -1$ *iff* $v$ is the user who first posted $i$. In such case, the corresponding tuple is called the root of the diffusion tree.



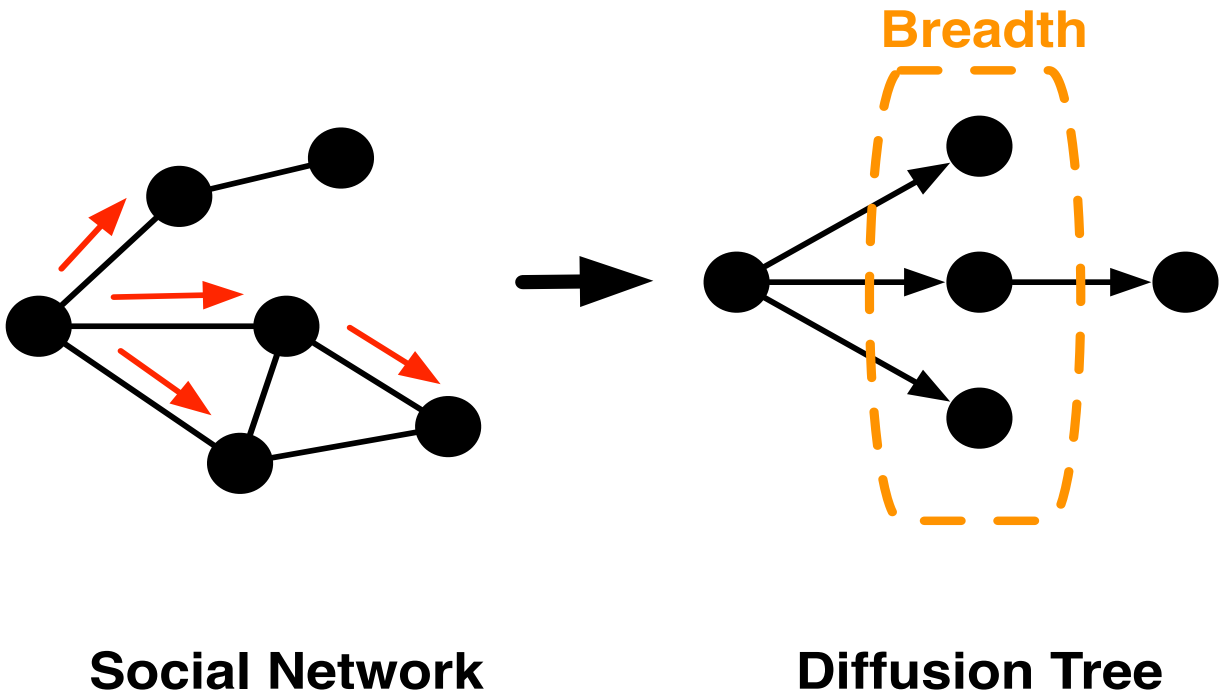

# Formulation



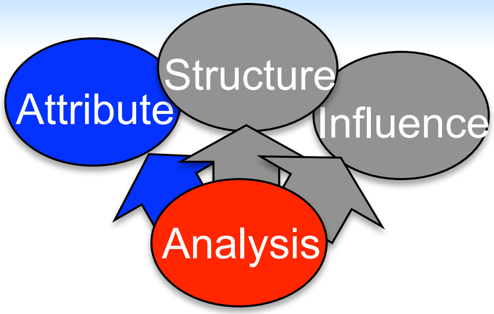

**Diffusion size: how many users will receive the information**



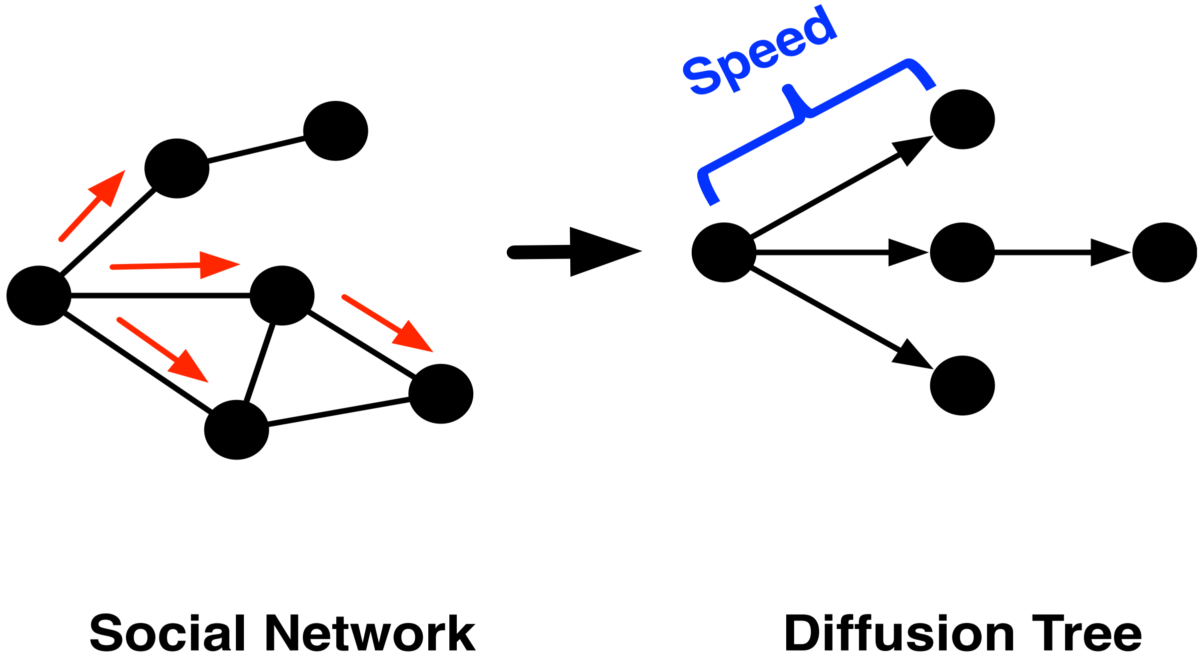

# Formulation



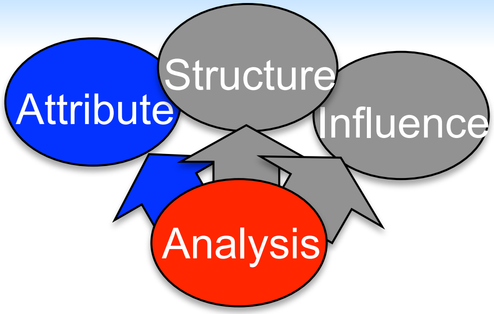

**Diffusion breadth: how widely the information will propagate**



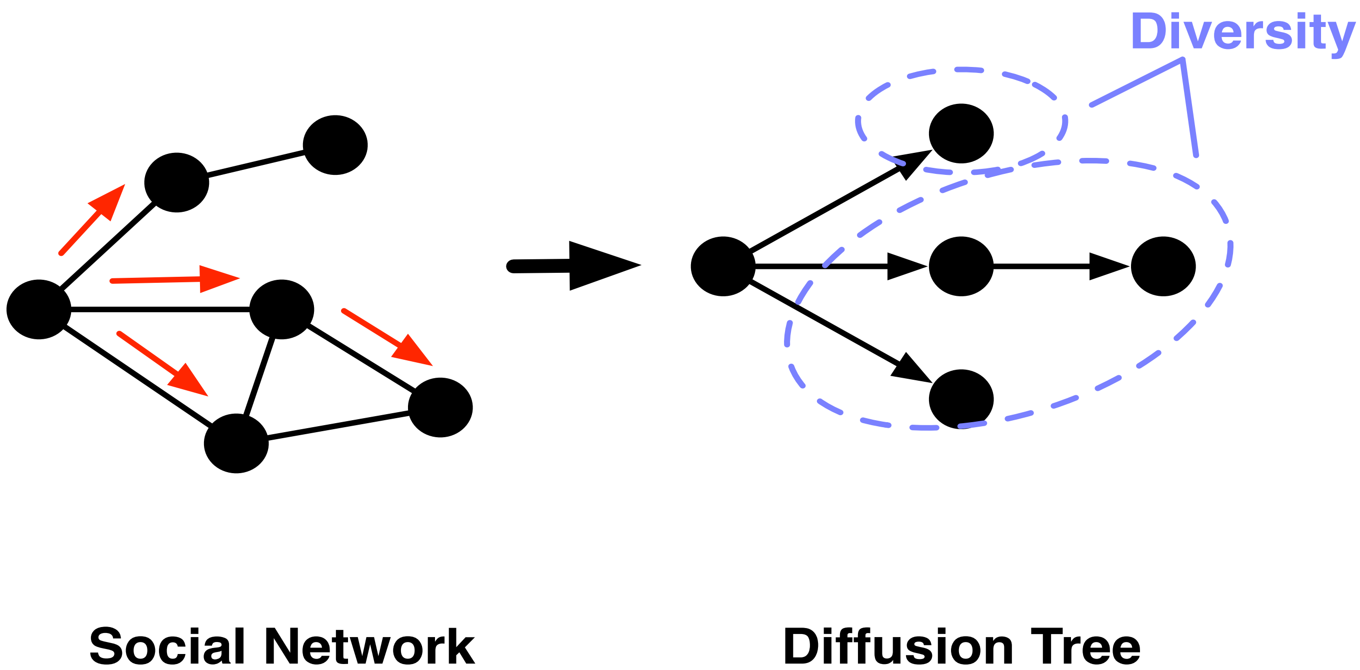

# Formulation




**Diffusion speed: how fast the information will propagate**

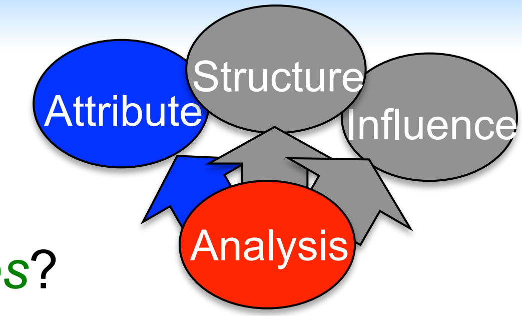



# Formulation

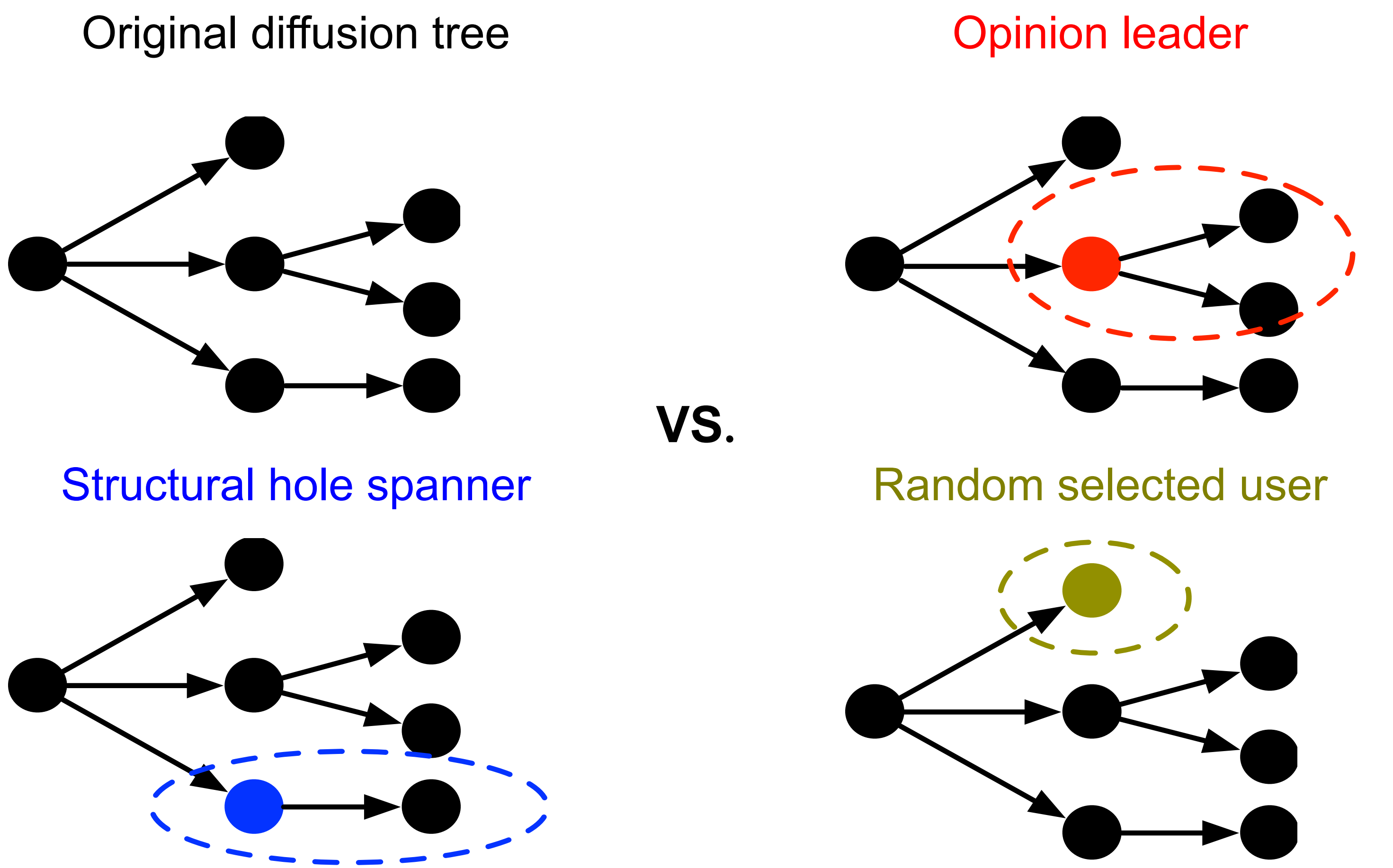


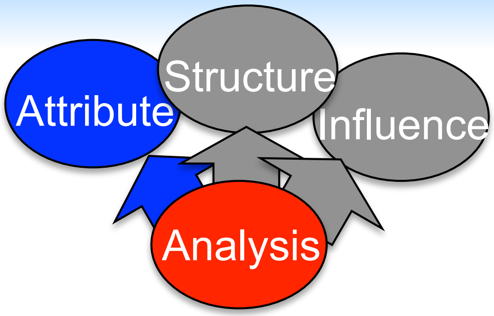

**Diffusion diversity: how many communities will receive the information**

# Analysis Setup

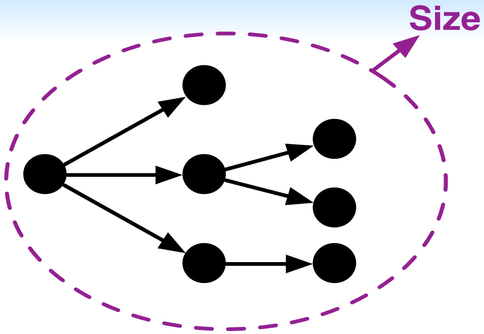


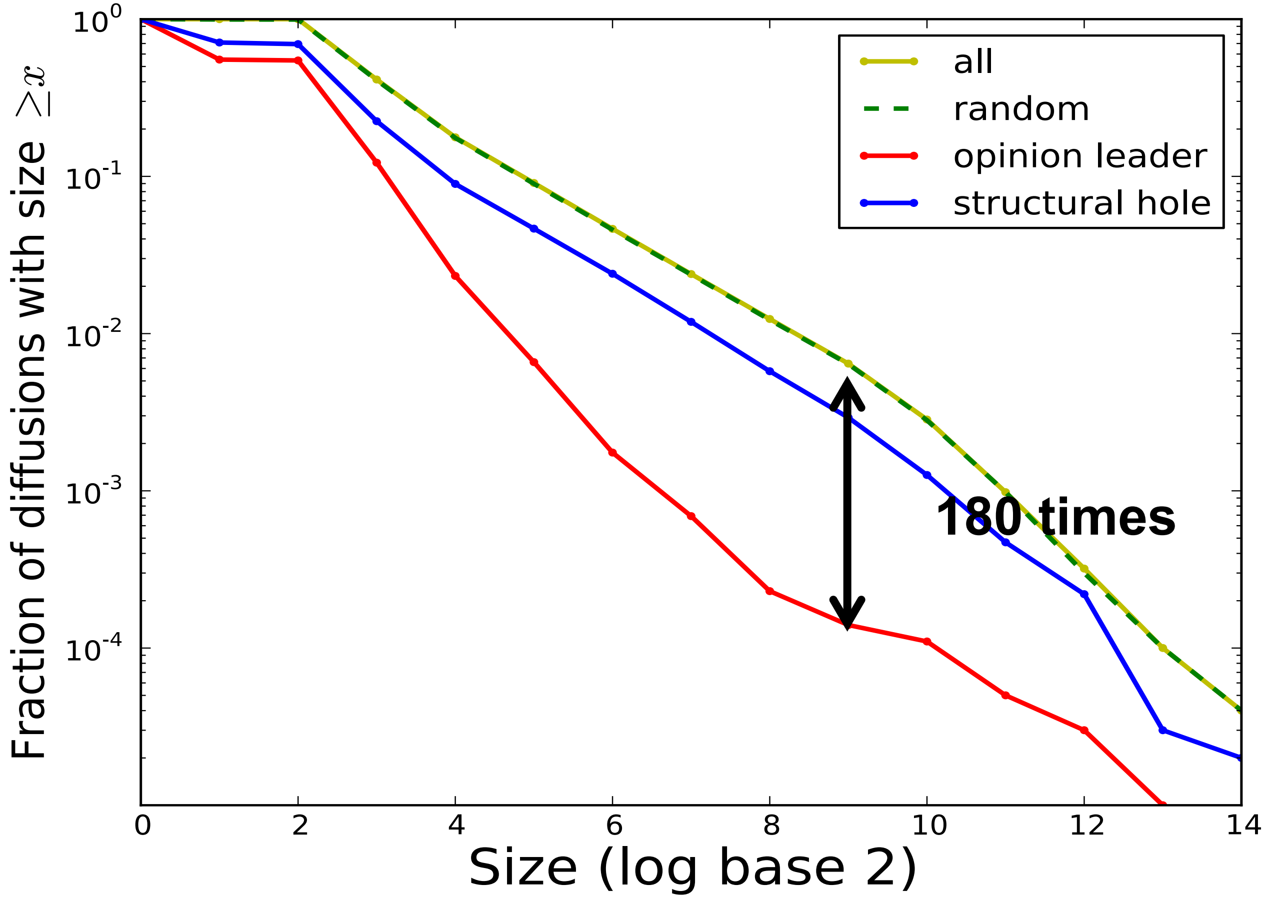

How different *social roles* influence different *diffusion attributes*?

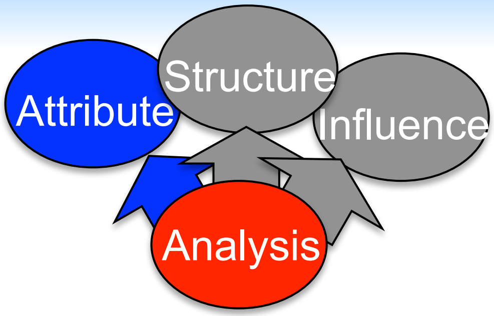




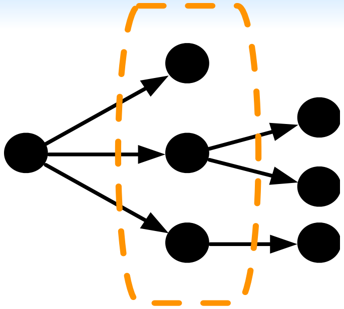

# Diffusion Size

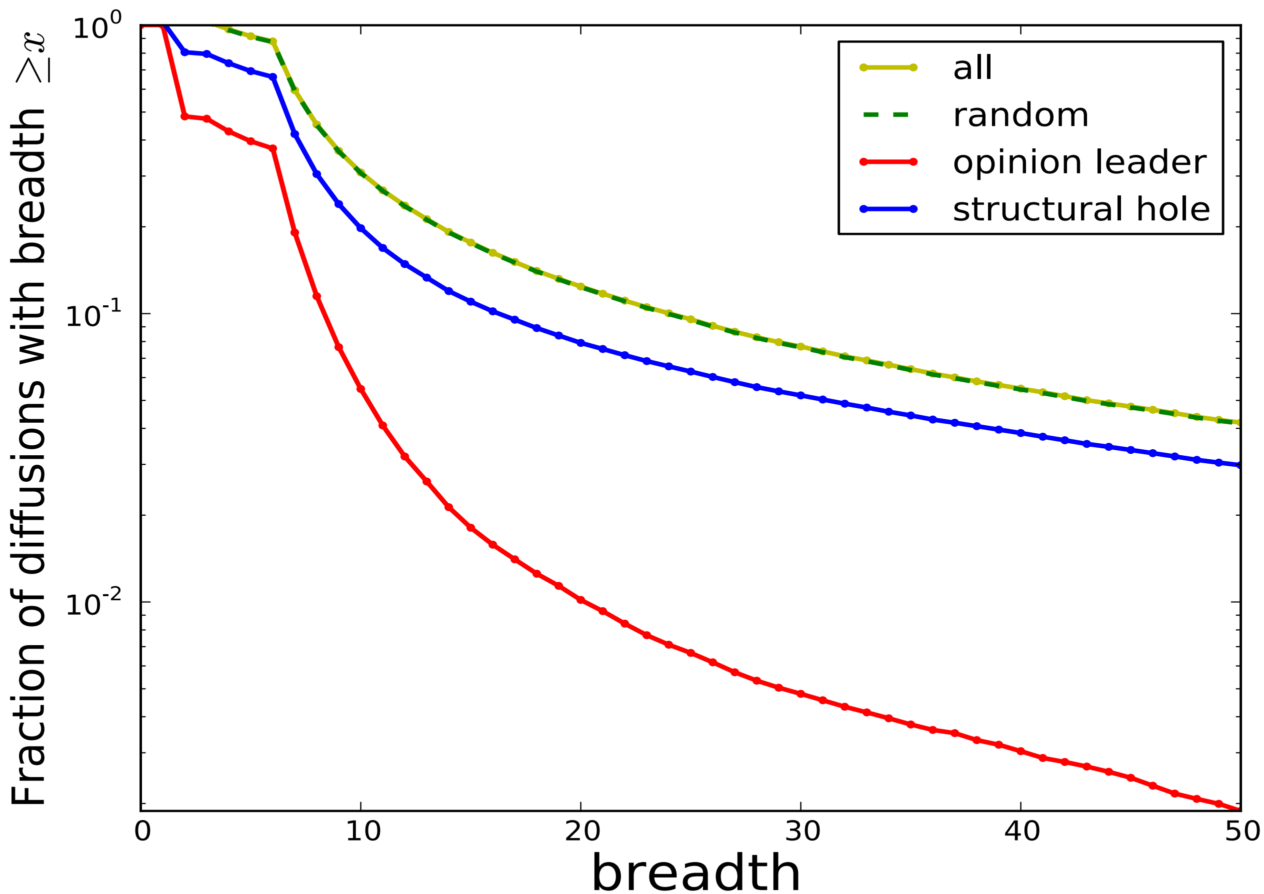


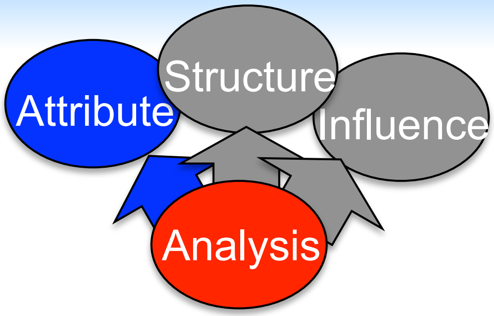





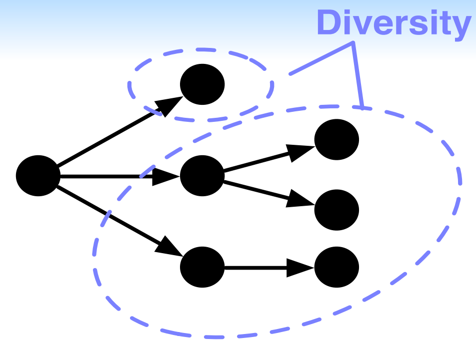

# Diffusion Breadth

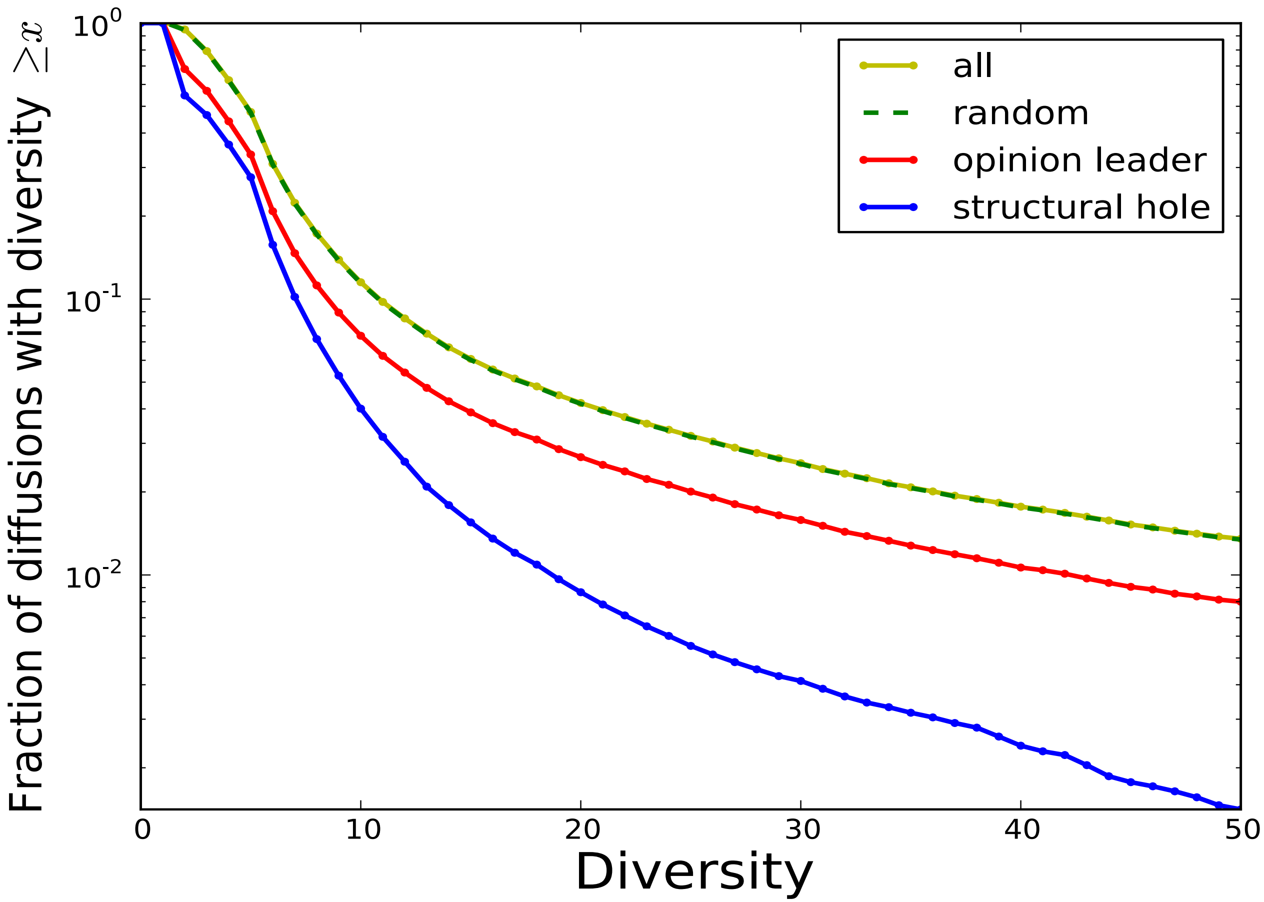


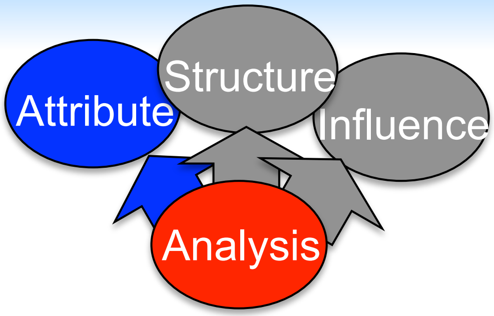





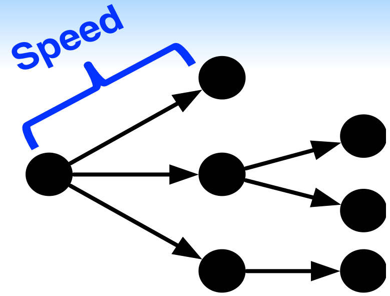

# Diffusion Diversity

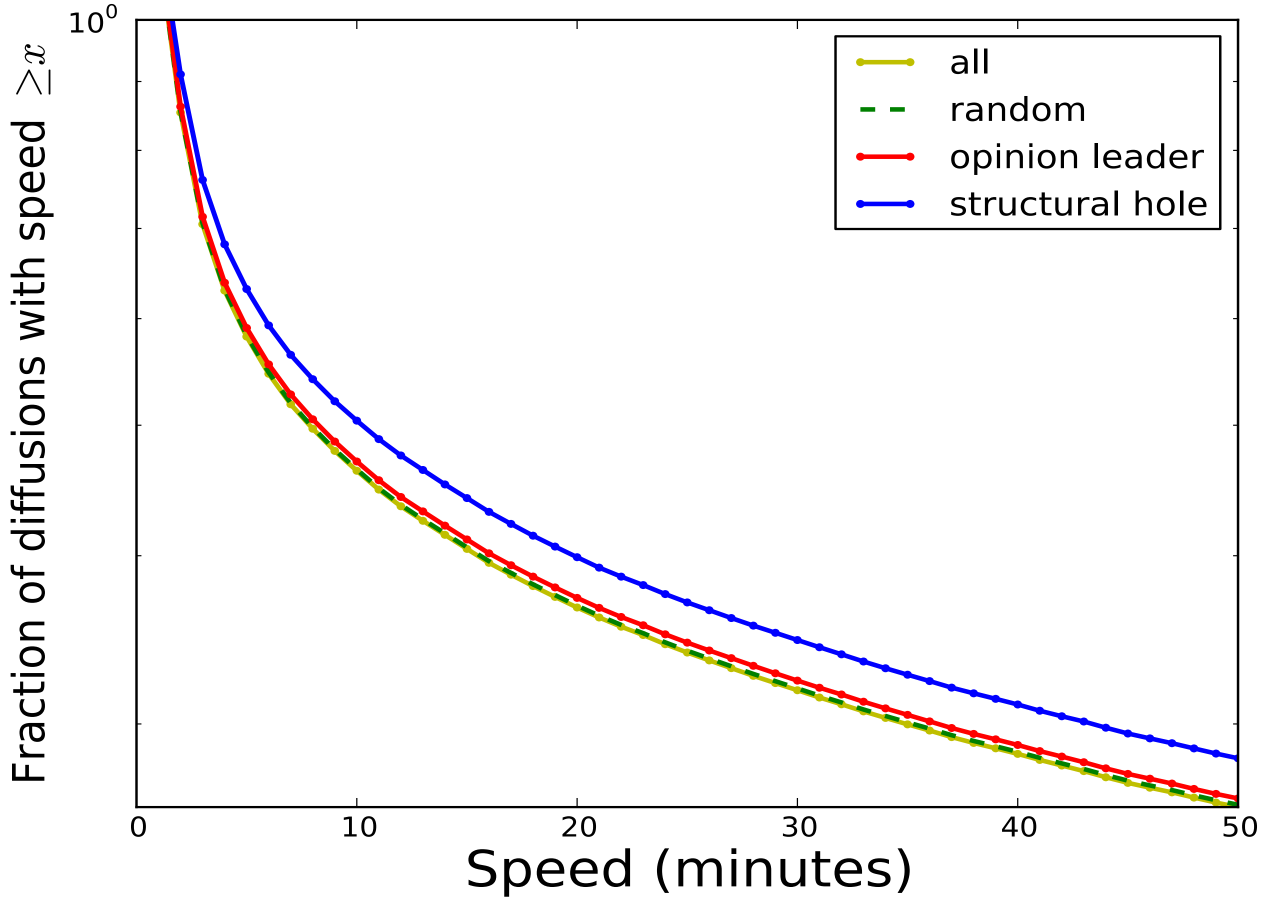


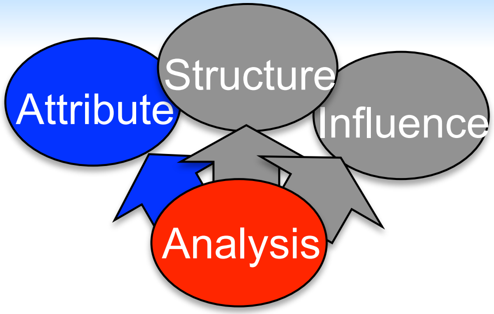





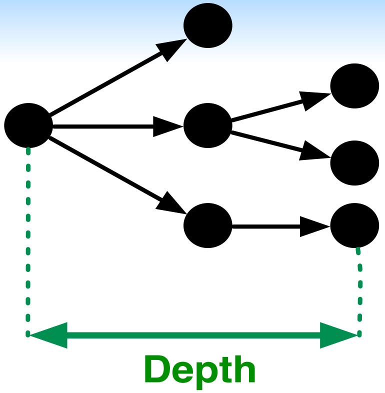

# Diffusion Speed

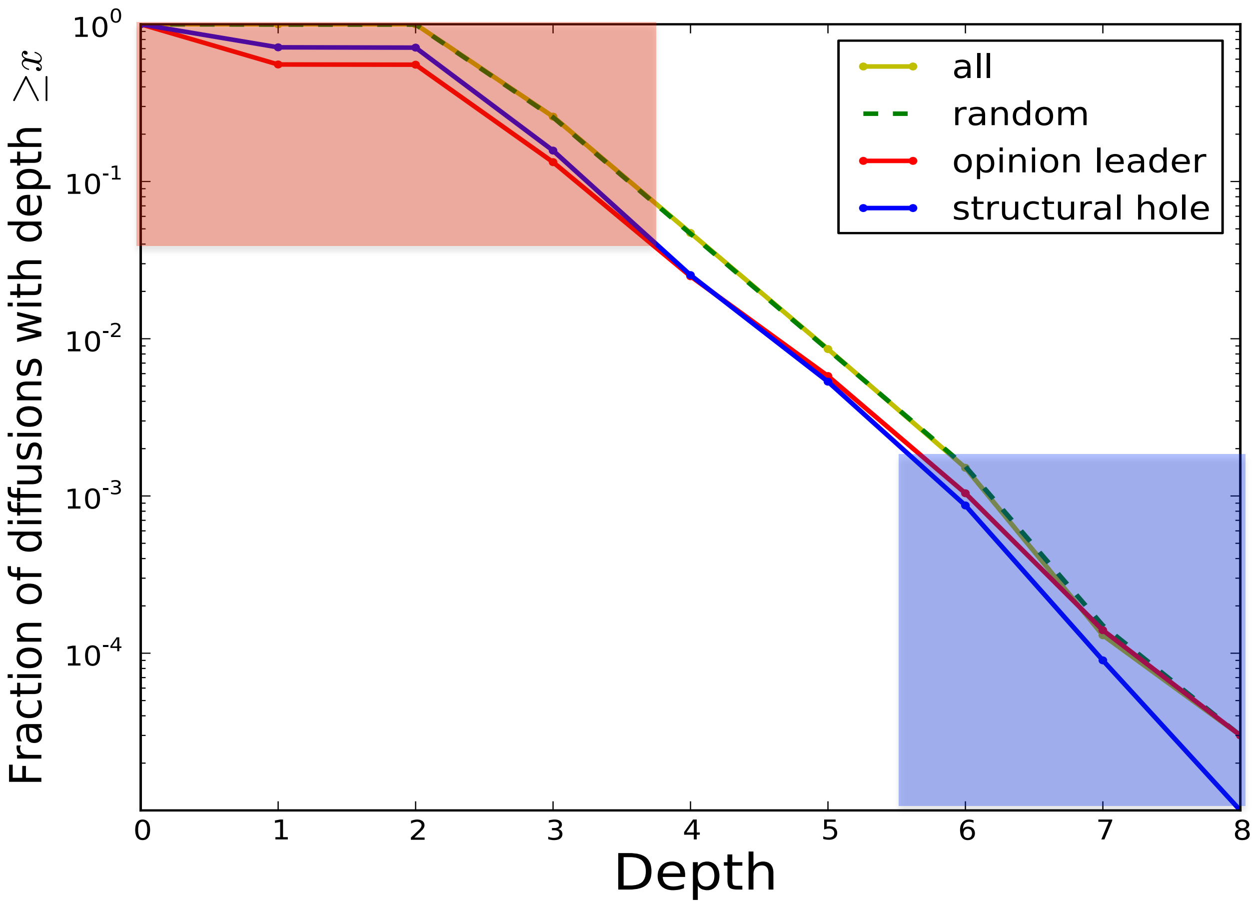


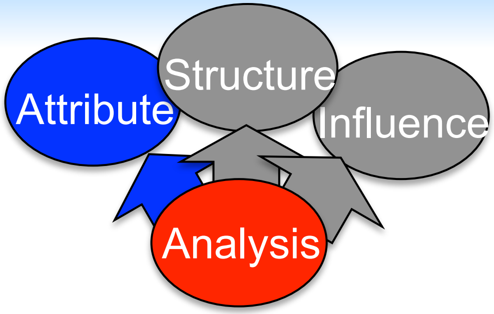






# Diffusion Depth






# Conclusion

- ***Opinion leaders*** are more influential on diffusion size & breadth;
- ***Structural hole spanners*** have more influence on diffusion diversity & speed;
- Diffusion depth is not sensitive to both opinion leaders and structural hole spanners.

# How to better model information diffusion by leveraging social role information?

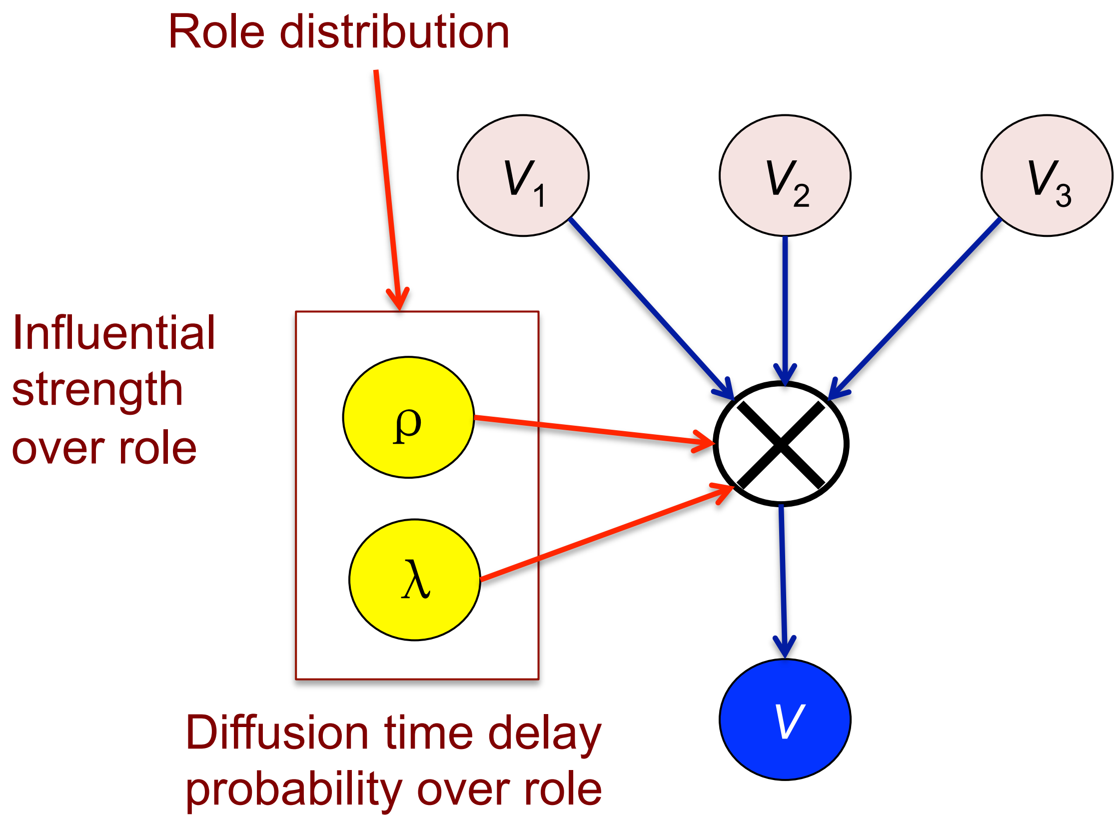
**Given:**

1. **A social network;**
2. **A set of historical diffusion trees.**

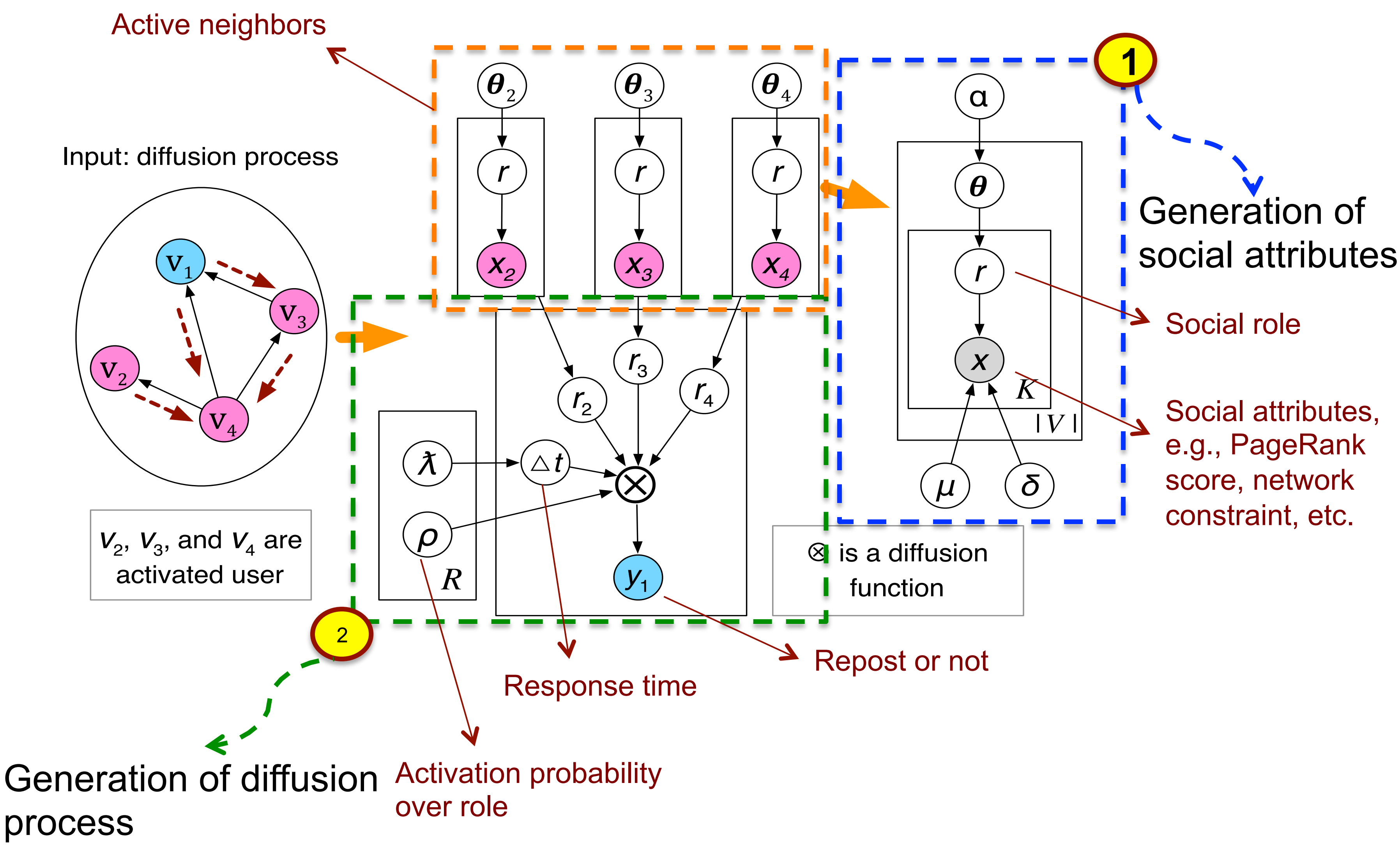
**Goal:**

1. **Model the diffusion process in future;**
2. **Infer social roles distributions of users.**

# Model: General Idea



# RAIN (Role Aware Information diffusioN)



# RAIN: Objective Function

- Likelihood:

$$L = \prod_{i=1}^{I} \prod_{t=1}^{T} \prod_{v \in A_{it}} P(v \in A_{it}) \times \prod_{i=1}^{I} \prod_{v \notin D_{iT}} P(v \notin D_{iT})$$

$$\times \prod_{u \in V} \prod_{k=1}^{K} P(x_{uk}) \times \prod_{u \in V} \prod_{r=1}^{R} P(\theta_{ur}|\alpha)$$

$$\times \prod_{r=1}^{R} \{P(\rho_r|\beta) + P(\lambda_r|\gamma)\} \times \prod_{r=1}^{R} \prod_{k=1}^{K} P(\mu_{rk}, \delta_{rk}|\tau)$$

The probability of user *v* adopting the information *i* at time *t*

$$P(v \in A_{it}) = \boxed{\sum_{\mathbf{z}_{i*v}^t} P(\mathbf{z}_{i*v})} - \prod_{u \in B(v) \cap D_{it-1}} \boxed{P(z_{iuv}^t = 0)}$$

$$= \prod_{u \in B(v) \cap D_{it-1}} (\varphi_{iuv}^t + \varepsilon_{iuv}^t) - \prod_{u \in B(v) \cap D_{it-1}} \varepsilon_{iuv}^t.$$

All adoptions

Failed adoptions

The probability of user *v* never adopts the information *i*

$$P(v \notin D_{iT}) = \prod_{u \in B(v) \cap D_{iT}} \sum_r (1 - \rho_r)\theta_{ur}.$$

Assumption here:
T >> the last observed timestamp

The probability of user *v* with the social attributes $x_{vk}$

$$P(x_{uk}) = \sum_r \sqrt{\frac{\delta_{rk}}{2\pi}} \exp\{-\frac{\delta_{rk}(x_{uk} - \mu_{rk})^2}{2}\}\theta_{ur}.$$

A mixture of Gaussian

Priors to model parameters

# Model Learning

Gibbs Sampling:

- Sample latent role *r* for user u's each social attribute

$$P(r_{uk}|\mathbf{r}_{\neg uk},\mathbf{x}) = \frac{P(\mathbf{x},\mathbf{r})}{P(\mathbf{x}_{\neg uk},\mathbf{r}_{\neg uk})} = \frac{n_{ur_{uk}}^{\neg uk}+\alpha}{\sum_r (n_{ur}^{\neg uk}+\alpha)} \frac{\Gamma(\tau_2+\frac{n_{r_{uk}k}}{2})}{\Gamma(\tau_2+\frac{n_{r_{uk}k}^{\neg uk}}{2})} \times \frac{\sqrt{(\tau_1+n_{r_{uk}k}^{\neg uk})}\eta(n_{r_{uk}k}^{\neg uk},\bar{x}_{r_{uk}k}^{\neg uk},s_{r_{uk}}^{\neg uk})}{\sqrt{(\tau_1+n_{r_{uk}k})}\eta(n_{r_{uk}k},\bar{x}_{r_{uk}k},s_{r_{uk}})},$$

- Sample role *r*, time delay *t*, and activation result *z* for each adoption

$$\begin{aligned}
&P(r_{iuv},\Delta t_{iuv},z_{iuv}|\mathbf{r}_{\neg iuv},\Delta\mathbf{t}_{\neg iuv},\mathbf{z}_{\neg iuv},\mathbf{y}) \\
&= \frac{P(\mathbf{r},\Delta\mathbf{t},\mathbf{z},\mathbf{y})}{P(\mathbf{r}_{\neg iuv},\Delta\mathbf{t}_{\neg iuv},\mathbf{z}_{\neg iuv},\mathbf{y}_{\neg iuv})} \\
&= \frac{n_{ur_{iuv}}^{\neg iuv}+\alpha}{\sum_r (n_{ur}^{\neg iuv}+\alpha)} \times \frac{n_{z_{iuv}r_{iuv}}^{\neg iuv}+\beta_1^{z_{iuv}}\beta_0^{1-z_{iuv}}}{n_{1r_{iuv}}^{\neg iuv}+\beta_1+n_{0r_{iuv}}^{\neg iuv}+\beta_0} \\
&\times \frac{(n_{r_{iuv}}^{\neg iuv}+\gamma_1)\prod_{t=0}^{\Delta t-2}(s_{r_{iuv}}^{\neg iuv}-n_{r_{iuv}}^{\neg iuv}+\gamma_0+t)}{\prod_{t=0}^{\Delta t-1}(\gamma_1+s_{r_{iuv}}^{\neg iuv}+\gamma_0+t)} \times \Phi,
\end{aligned}$$

- Update model parameters according to sampling results

**Input**: the hyper-parameters $\alpha$, $\beta$, $\gamma$, and $\tau$, the number of social roles $R$, a social network $G$ along with each user's social attribute $\mathbf{x}_v$, and a set of diffusion trees.

```
foreach user u ∈ V do
    Initialize θ_u randomly;
end
for r = 1 to R do
    Initialize ρ_r and λ_r randomly;
end
repeat
    % sampling process;
    foreach user u ∈ V do
        for k = 1 to K do
            Draw a latent variable r, which is associated
            with x_uk, according to P(r_uk | r_¬uk, x) (Eq. 7);
        end
    end
    foreach 4-tuple (u, v, i, t) in each diffusion tree do
        Draw latent variables (t, r, z) according to
        P(r_iuv, Δt_iuv, z_iuv | r_¬iuv, Δt_¬iuv, z_¬iuv, y) (Eq. 9);
    end
    % parameter update;
    for r = 1 to R do
        Update λ_r and ρ_r according to Eq. 10;
        foreach user u ∈ V do
            Update θ_ur according to Eq. 10;
        end
        for k = 1 to K do
            Update μ_rk and δ_rk according to Eq. 11
        end
    end
until Convergence;
```

# Retweet Prediction

Table 2: Performance of repost prediction on several topics.

| Topic | Method | P@10 | P@50 | P@100 | MAP |
|---|---|---|---|---|---|
| Campus | C | | | | |
| | S | | | | |
| | IC | | | | |
| | R | | | | |
| Horoscope | C | | | | |
| | S | | | | |
| | IC | | | | |
| | R | | | | |
| Movie | C | | | | |
| | S | | | | |
| | IC | | | | |
| | R | | | | |
| History | C | | | | |
| | S | | | | |
| | IC | | | | |
| | R | | | | |
| Society | C | | | | |
| | S | | | | |
| | IC | | | | |
| | R | | | | |
| Health | C | | | | |
| | S | | | | |
| | IC | | | | |
| | R | | | | |
| Political | C | | | | |
| | S | | | | |
| | IC Model | [illegible] | [illegible] | [illegible] | [illegible] |
| | RAIN | **0.216** | **0.164** | **0.130** | **0.239** |
| Travel | Count | 0.142 | 0.056 | 0.031 | 0.103 |
| | SVM | 0.094 | 0.048 | 0.032 | 0.128 |
| | IC Model | **0.206** | 0.120 | 0.098 | 0.254 |
| | RAIN | 0.194 | **0.159** | **0.126** | **0.260** |

**Goal**: predict whether a user will repost a particular post

**Data**: a complete Tencent Weibo data on Nov. 1-3, 2012

- Posts are categorized based on topics: ***campus, constellation, movie, history, society, health, political, and travel***
- Posts on Nov.1-2 as train data, Nov. 3 as test data

# Retweet Prediction

Table 2: Performance of repost prediction on several topics.

| Topic | Method | P@10 | P@50 | P@100 | MAP |
|---|---|---|---|---|---|
| Campus | Count | 0.028 | 0.010 | 0.006 | 0.068 |
| | SVM | | | | |
| | IC Model | | | | |
| | RAIN | | | | |
| Horoscope | Count | | | | |
| | SVM | | | | |
| | IC Model | | | | |
| | RAIN | | | | |
| Movie | Count | | | | |
| | SVM | | | | |
| | IC Model | | | | |
| | RAIN | | | | |
| History | Count | | | | |
| | SVM | | | | |
| | IC Model | | | | |
| | RAIN | | | | |
| Society | Count | | | | |
| | SVM | | | | |
| | IC Model | | | | |
| | RAIN | | | | |
| Health | Count | | | | |
| | SVM | | | | |
| | IC Model | | | | |
| | RAIN | | | | |
| Political | Count | | | | |
| | SVM | | | | |
| | IC Model | | | | |
| | RAIN | | | | |
| Travel | Count | | | | |
| | SVM | | | | |
| | IC Model | | | | |
| | RAIN | | | | |

## Baselines:

Count: ranks users by the number of active followees

SVM: Support Vector Machine, majorly considers features as

- *#active followers*
- *#active followees*
- *#whether the user have reposted similar messages*

IC Model: traditional IC model with fitted parameters[1]

**RAIN**: **R**ole **A**ware **IN**formation diffusion

## Evaluation Metrics:

Precision@K (K=10, 50, 100)

Mean Average Precision (MAP)

[1] Kimura, M.; Saito, K.; Ohara, K.; and Motoda, H. 2011. Learning information diffusion model in a social network for predicting influence of nodes. Intelligent Data Analysis 15(4):633–652.

# Retweet Prediction

Table 2: Performance of repost prediction on several topics.

| Topic | Method | P@10 | P@50 | P@100 | MAP |
|---|---|---|---|---|---|
| Campus | Count | 0.028 | 0.010 | 0.006 | 0.068 |
| | SVM | 0.098 | 0.045 | 0.032 | 0.127 |
| | IC Model | **0.231** | 0.142 | 0.102 | 0.259 |
| | RAIN | 0.228 | **0.145** | **0.106** | **0.263** |
| Horoscope | Count | 0.019 | 0.010 | 0.006 | 0.005 |
| | SVM | 0.124 | **0.162** | 0.088 | **0.263** |
| | IC Model | 0.149 | 0.111 | 0.098 | 0.125 |
| | RAIN | **0.171** | 0.121 | **0.102** | 0.130 |
| Movie | Count | 0.015 | 0.007 | 0.004 | 0.009 |
| | SVM | 0.094 | 0.111 | 0.060 | 0.199 |
| | IC Model | 0.227 | 0.147 | **0.147** | 0.236 |
| | RAIN | **0.229** | **0.173** | 0.144 | **0.238** |
| History | Count | 0.191 | 0.056 | 0.033 | 0.096 |
| | SVM | 0.154 | 0.051 | 0.030 | 0.221 |
| | IC Model | 0.206 | 0.134 | **0.135** | 0.230 |
| | RAIN | **0.225** | **0.171** | 0.134 | **0.262** |
| Society | Count | 0.245 | 0.058 | 0.029 | 0.156 |
| | SVM | 0.100 | 0.023 | 0.012 | 0.122 |
| | IC Model | 0.171 | 0.131 | **0.109** | 0.198 |
| | RAIN | **0.176** | **0.140** | 0.106 | **0.204** |
| Health | Count | 0.041 | 0.008 | 0.005 | 0.035 |
| | SVM | 0.164 | 0.064 | 0.039 | **0.197** |
| | IC Model | 0.169 | 0.113 | 0.096 | 0.162 |
| | RAIN | **0.175** | **0.134** | **0.115** | 0.185 |
| Political | Count | 0.019 | 0.005 | 0.003 | 0.007 |
| | SVM | 0.104 | 0.077 | 0.039 | 0.176 |
| | IC Model | 0.209 | 0.132 | 0.102 | 0.224 |
| | RAIN | **0.216** | **0.164** | **0.130** | **0.239** |
| Travel | Count | 0.142 | 0.056 | 0.031 | 0.103 |
| | SVM | 0.094 | 0.048 | 0.032 | 0.128 |
| | IC Model | **0.206** | 0.120 | 0.098 | 0.254 |
| | RAIN | 0.194 | **0.159** | **0.126** | **0.260** |

**Comparison Results:**

- **Count: performs worst due to the lack of supervised information.**
- **SVM: performs well on *local topics* but falls short on *global topics*.**
- **IC Model: suffers from *model complexity*.**
- **RAIN: improves the performance *+32.6%* in terms of MAP by reducing model complexity.**

# Diffusion Scale Prediction

- We predict the ***scale*** of a diffusion process
  - X-axis: the number of reposts
  - Y-axis: the proportion of original posts with particular number of reposts

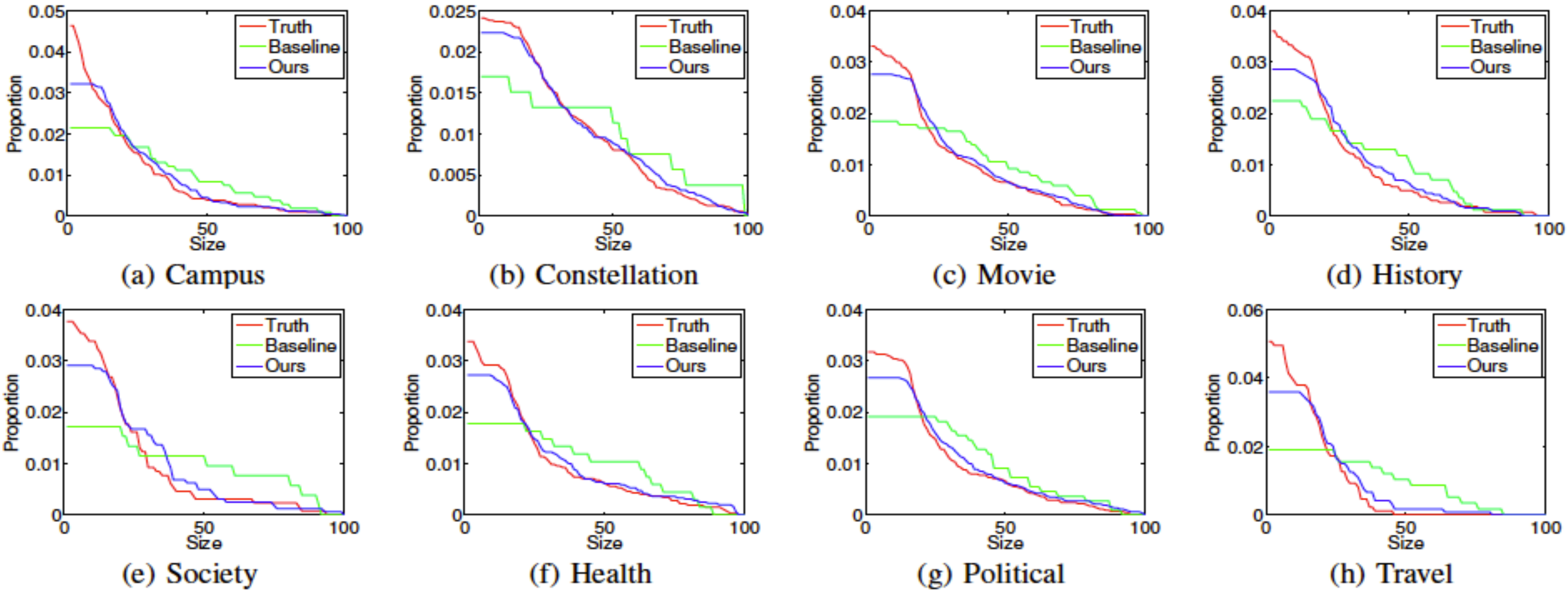



**Figure 8: Diffusion scale distributions of the different topics in the test set.**

# Diffusion Duration Prediction

- We predict the ***duration*** of a diffusion process
  - X-axis: the time interval between the first and last posts
  - Y-axis: the proportion of original posts with particular time interval

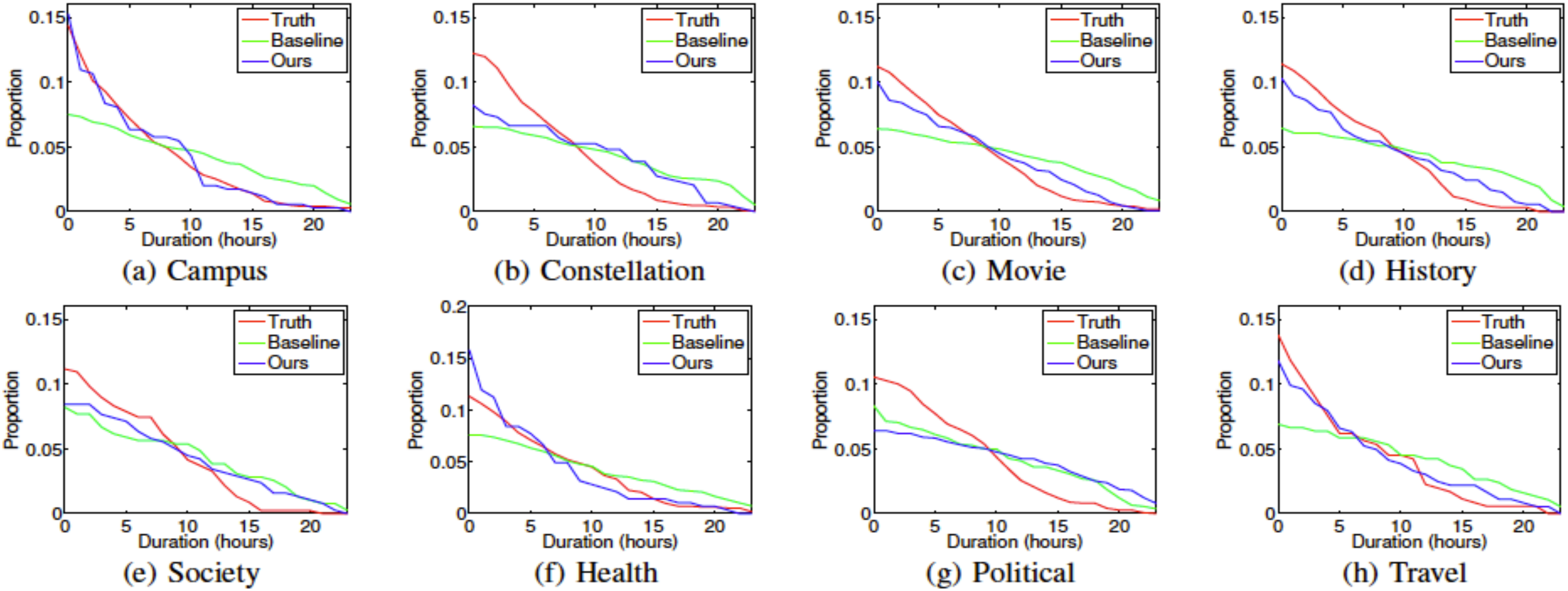



**Figure 9: Diffusion duration distributions of the different topics in the test set.**