1 Degree Distribution

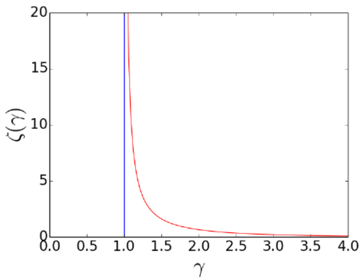
Poisson Distribution

Formula: $P(k) \sim \frac{\lambda^k e^{-\lambda}}{k!}$
Parameter: λ is the mean degree $\langle k \rangle$ **Examples** $G(n,p)$ model with large n and fixed $np \Rightarrow \lambda = np$

Zipf (Zeta) Distribution

It is one of the discrete power law probability distribution.
Formula $P(k) = \frac{k^{-\gamma}}{\zeta(\gamma)}$
Parameters
- $\zeta(\gamma) = \sum_i i^{-\gamma}, (\gamma > 1)$
- γ: increase, $\zeta(\gamma) \to 0$, the left (peak) of Zeta distribution goes higher and the tail goes thinner.

Examples scale-free network.
Moments of Zeta distribution $\langle k^m \rangle = \frac{\zeta(\gamma - m)}{\zeta(\gamma)}$
- $\langle k \rangle$(m=1): finite for $\gamma > 2$
- $\langle k^m \rangle = \frac{\zeta(\gamma - m)}{\zeta(\gamma)}$: finite for $\gamma > 1 + m$
- $\gamma \in (2,3)$: the generated scale-free network is super robust against random failure.

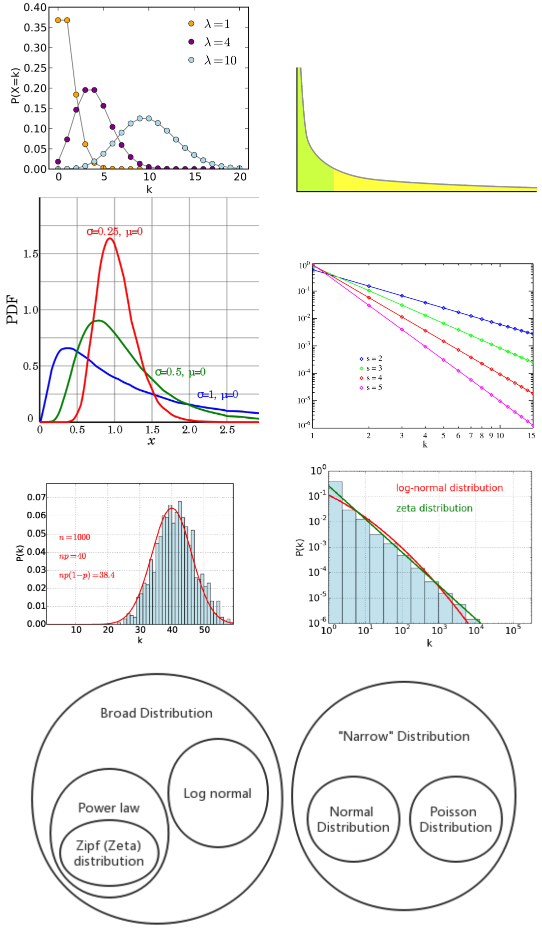


Broad Distribution

It is one of the heavy tail distribution. The heavy tail distribution means compared to exponential distribution(normal distribution), its tail is heavier.
Examples power-law distribution, log-normal distribution. (These two are very similar, but not exactly the same.)
Power law: scale-free network shows power law pattern.

Normal Distribution

Example ER model with large n and fixed p



(Order from the top-left to the bottom-right)1. Poisson. 2. Power-law. 3. Log-normal. 4. Zipf(Zeta). 5. Normal. 6. Comparison between log-normal and zipf in log-log plot. 7. Set relations among those distribution.

2 Models and Metrics Comparison

idea, algorithms, parameters, properties, applications

G(n,m) model

Parameters: n: the number of nodes. m: the number of edges
Algorithm:
1.generate n nodes.
2. randomly choose m edges from : $\frac{n(n-1)}{2}$ (without self-loop, undirected), or $\frac{n(n+1)}{2}$(with self-loop, undirected), or $n(n-1)$ (without self-loop, directed) or n^2(with self-loop, directed) edges.
Properties
- **Micro-canonical ensemble**: all the microstates have the same probability.

G(n,p) model

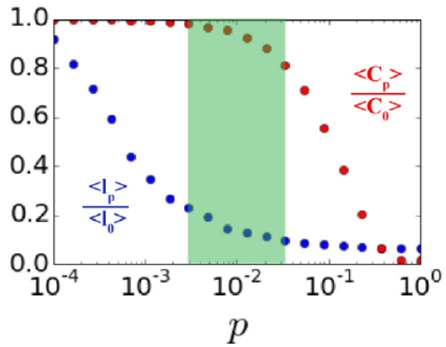
[1] **Parameters**: n: the number of nodes. p: the probability of one node linking to any node
Algorithm: (directed, with self-loop)
1.generate n nodes.
2.for each node i:
— 2.1 for $j = 1, \cdots, n$:
——- generate a uniformly random number q, if $q < p$, add an edge (i,j)
Properties
- **Limiting degree distribution**:when $n \to \infty$, if p is fixed, the degree distribution approximates to normal distribution, with $\mu = np, \sigma^2 = np(1-p)$
if np is fixed, it approximates to Poisson distribution, with $\lambda = np$.
- **Emergence of giant connected component** $2 = \frac{\langle k^2 \rangle}{\langle k \rangle} = 1 + np \Leftrightarrow np = 1 \Leftrightarrow p = \frac{1}{n}$
- **Critical failure rate** $np(1 - q_c) = 1 \Leftrightarrow q_c = 1 - \frac{1}{np}$

Ring lattice

Parameters: n: the number of nodes. d: dimension. s: distance, the llinked neighbors are $2s$.

Watts Strogatz

Parameters: n: the number of nodes. d: dimension. s: distance, the llinked neighbors are $2s$. p: rewire probability.
Algorithm:
1. construct a ring lattice (d,n,s)
2. rewire source of each link to random node with probability p.
Small world regime



Molloy Reed (Configuration Model)

Algorithm:
1. create empty network with n nodes
2 for each node i generate d_i "link stubs"
3 draw pairs of link stubs uniformly at random and connect them until no stubs are left.
Properties
- **Micro-canonical ensemble**

ERGM

Idea
- maximize entropy: this will be the system with the largest remaining uncertainty, and by choosing it you're making sure you're not adding any extra biases or uncalled for assumptions into your analysis. $\max_P H(\Omega, P) := -\sum_{g \in \Omega} P(g) \log(P(g))$
- preserve the expected properties. $\langle f_i(\Omega) \rangle := \sum_{g \in \Omega} f_i(g) P(g) = f(G_e)$

Parameters: θ: how big and which direction of the corresponding property's effect.
Algorithm to estimate parameters (MCMCMLE)
1. choose the initial value Θ_0
2. generate enough microstates by $P(\Theta_0)$ (Random walk)
3. compute the likelihood of the expected properties $P_{\Theta_0}(f_e)$
3. define $\Theta_1 = \Theta_0 + \epsilon$
4. generate microstates and compute $P_{\Theta_1}(f_3)$
5. compare P_{Θ_1} and P_{Θ_0} and take the one with bigger likelihood.
6. repeat for enough iterations.

Key Formulas $\frac{\partial}{\partial P(G)} \{H + \alpha(1 - \sum_G P(G)) + \sum_i \theta_i (\langle f_i \rangle - \sum_G f_i(G) P(G))\} = 0$
- $P(G) = \frac{\exp(-\sum_i \theta_i f_i(G))}{Z}$
- $Z = \sum_G \exp(-\sum_i \theta_i f_i(G))$
- $\langle f_i \rangle = -\frac{1}{Z} \frac{\partial Z}{\partial \theta_i}$

Stochastic Block Model

Parameters
- M: stochastic block property matrix. M_{kl}: the probability of edges between block k and block l (or within the block, if $k = l$).
- G: the network.
- z: the block assignment vector.
- E: the actual count matrix of edge numbers between two blocks (or within)
- N: the maximum number of edges between two blocks (or within one)

Usages
- using M and z, we can generate new G.
- using G and z, we can find the MLE M.

Formula:
$L(M, \vec{z}) = \sum_{1 \le k \le l \le B} M_{kl}^{E_{kl}} (1 - M_{kl})^{N_{kl} - E_{kl}}$
MLE $M := \hat{M}$, with $\hat{M}_{kl} = \frac{E_{kl}}{N_{kl}}$.
Microstate canonical version M, instead of being the probability matrix, becomes the

[1] For ER models, we use undirected, with self-loops by default

count matrix. M_{kl}: the number of edges between block k and l.
Their comparisons are listed in the table1.

3 Minimal Discription length Principle VS MapEquation

MDL
This is a principle, which guides to find the optimal (number of) parameters. **Definition** $\lambda(\vec{z}) = H(\vec{z}) + \Delta\vec{z}$
However, this is just a general definition. It can have different concrete definitions under different frameworks.

MapEquation
Definition $L(\vec{z}_k) = Q_i H(Q_i) + \sum_{i=1}^{k} P_i H(P_i)$
This is a flow-based approach to realize the MDL principle. MapEquation tutorial

4 Matices and Diffusion

Adjacency Matrix: A
Definition $A_{ij}: d_{ij}$ the (weighted) number of edges from n_i to n_j.

Transition Matrix: T
Definition $T = A \cdot D^{-1}$, $T_{ij}: \frac{d_{ij}}{\sum_j d_{ij}}$, the normalized degree from n_i to n_j, by the outdegree.
Properties:
- $\sum_j t_{ij} = 1$

Laplacian Matrix: L
Definition $L = D - A$, D: a diagonal matrix with the $d_i i = d_{out}(i)$.
Meaning: It origins from Laplacian operator. $\frac{dx^{(t)}}{dt} = C(A-D)x^{(t)} = -CLx^{(t)}, C > 0$
Properties:
$\sum_j L_{ij} = 0$

Diffusion
Distribution
- total variation distance convergence time t: $t \propto \frac{\log\epsilon}{\log|\lambda_2|}$: the smaller, the larger $|\log\lambda_2|$, the smaller t is, the faster of the diffusion.
- spectral gap: $1-\lambda_2$: the larger the gap, the faster the diffusion.

Continuous time model (Laplacian matrix)
$\frac{dx^{(t)}}{dt} = -C \cdot L \cdot x^{(t)} \Rightarrow x^{(t)} = \sum_{i=1}^{n} a_i e^{-C\lambda_i}\vec{v_i} \simeq a_i e^{-C\lambda_2}\vec{v_2}$
- L determines the change rate. Larger λ_2, faster change, faster diffusion.
- $\lambda_2 = 0 \Leftrightarrow$ the network is disconnected. Capture how well connected the network is.

Summary No matter using transition matrix or laplacian matrix, there is one point in common: the larger eigenvalue gap $|\lambda_1 - \lambda_2|$ is, the faster the diffusion will be.
- in transition matrix: $|\lambda_1 - \lambda_2| = |1 - \lambda_2|$.
- in laplacian matrix: $|\lambda_1 - \lambda_2| = |\lambda_2|$.

5 Random Walk Model

Diffusion process
plus Markov property to generate the transition matrix.

Community detection: Flow Compression
using random walk to generate a series of walk history for code compression.

ERGM
Like application in diffusion process, generate the transition matrix and simulate numerous times to approximate the stationary distribution.

6 Key Concepts

Eigenvalue gap
: the gap between two largest eigenvalues of a matrix. Applied in Transition matrix, it is $1-\lambda_2$.

7 Formulas

network diameter

$$\mathrm{diam}(G) = \max_{(v,w)\in V\times V} \mathrm{dist}(v,w)$$

average shortest path length

$$\langle l \rangle = \frac{1}{|v|^2} \sum_{(v,w)\in V\times V} \mathrm{dist}(v,w)$$

giant connnected components

$$G' \Leftarrow \frac{|G'|}{|G|} \simeq 1$$

powers of adjacency matrix A_{ij}^k: the number of paths with length k between i and j.
partition quality

$$Q(G,C) = \frac{1}{2m} \sum_{(i,j)} (A_{ij} - \frac{d_i d_j}{2m}) \delta(C_i, C_j)$$

theoretical maximum of partition quality

$$Q_{\max}(G,C) = \frac{1}{2m} \sum_{(i,j)} (1 - \frac{d_i d_j}{2m}) \delta(C_i, C_j)$$

Modularity:

$$Q_{\mathrm{opt}}(G) = \max_C Q(G,C)$$

community assortativity coefficient:

$$\frac{Q_{\mathrm{opt}}(G)}{Q_{\max}(G,\tilde{C})}$$

$\tilde{C}$ is the one selected by modularity.

betweenness centrality:

$$C_B(i) = \sum_{s\neq i, t\neq i} n_i(s,t)$$

$n_i(s,t)$: the number of shortest path from s to t passing i or (normalized)

$$C_B(i) = \sum_{s\neq i, t\neq i} \frac{n_i(s,t)}{N_{s,t}}$$

$N_{s,t}$: the number of shortest path from s to t

closeness centrality:

$$C_C(i) = \frac{n-1}{\sum_j \mathrm{dist}(i,j)}$$

degree assortativity coefficient:

$$r = \frac{\sum_{i,j} A_{ij} d_i d_j - \frac{1}{2|E|} \sum_{i,j} (d_i d_j)^2}{\sum_i d_i^3 - \frac{1}{2|E|} \sum_{i,j} (d_i d_j)^2}$$

local clustering coefficient:

$$C_i = \frac{2 \cdot k(i)}{d_i(d_i - 1)} = \frac{k(i)}{\binom{d_i}{2}}$$

or for directed network:

$$C_i = \frac{k(i)}{d_{\mathrm{out}}(i)(d_{\mathrm{out}}(i) - 1)}$$

global clustering coefficient:

$$C = \frac{1}{n} \sum_i C_i$$

Generating functions:

$$[\frac{1}{k!} \frac{d^k}{dx^k} G_0(x)]_{x=0} = P(k)$$

$$G_0'(x)_{x=1} = \langle k \rangle$$

$$G_1'(x)_{x=1} = \frac{\langle k^2 \rangle}{\langle k \rangle} - 1$$

$$[(x\frac{d}{dx})^m G_0(x)]_{x=1} = \langle k^m \rangle$$

$G_z(x) = G_X(x) G_Y(x)$ generates the distribution of the sum of two random independent variable X and Y.
$xG_0(x)$ generates distribution of $P(x+1=k) = P(x=k-1)$
$\frac{1}{x}G_0(x)$ generates distribution of $P(x-1=k) = P(x=k+1)$
$[G_0(x)]^m$ generates distribution for the sum of m independent realizations of random variable X.
$G_0(G_1(X)) = \sum_k P(k)[G_1(x)]^k$ generates $P(\sum_{i=1}^{X} Y_i = k)$.

Friendship paradox:
The degree distribution of a node's neighbor (intuition: If a node has degree k, it has k times to be others' neighbor.):

$$P_1(k) \propto kP_0(k) = \frac{kP_0(k)}{\langle k \rangle}$$

$G_1(x)$ not including the "father"link.

$$G_1(x) = \frac{1}{x} \sum_k P_1(x) x^k = \frac{G_0'(x)}{G_0'(1)}$$

Generating function of neighbour degree distribution

$$x \cdot G_1(x)$$

Mean neighbor degree

$$\langle k_n \rangle = 1 + G_1'(1) = \frac{\langle k^2 \rangle}{\langle k \rangle} = \langle k \rangle + \frac{Var(X)}{\langle k \rangle}$$

Poisson distribution:

$$G_1(x) = \frac{G_0'(x)}{G_0'(1)} = G_0(x)$$

$$\Rightarrow \langle k_n \rangle = G_1'(1) + 1 = G_0'(1) + 1 = \langle k \rangle + 1$$

Component size:
$H_1(x)$: as the generating function of the component size that includes a node as some other node's neighbor.

$H_0(x)$: as the generating function of the component size that includes a random node.

$$H_1(x) = xG_1(H_1(x))$$

$$\Rightarrow H_1'(1) = \frac{1}{1-G_1'(1)}$$

$$H_0(x) = xG_0(H_1(x))$$

$$\langle s\rangle = H_0'(1) = 1 + G_0'(1)H_1'(1)$$

$$\Rightarrow \langle s\rangle = 1 + \frac{G_0'(1)}{1-G_1'(1)} = 1 + \frac{G_0'(1)}{2-\frac{\langle k^2\rangle}{\langle k\rangle}}$$

Molloy-Reed criterion: the critical point of the emergence of giant connected component:

$$\frac{\langle k^2\rangle}{\langle k\rangle} > 2$$

Surviving network:

$$F_0(x) = (1-q)G_0(x)$$

$$F_1(x) = (1-q)G_1(x)$$

$$H_1(x) = q + xF_1(H_1(x)) = q + (1-q)xG_1(H_1(x))$$

$$H_0(x) = 1 + xF_0(H_1(x)) = q + (1-q)xG_0(H_1(x))$$

$$\langle s\rangle = H_0'(1) = (1-q)[1 + \frac{(1-q)\langle k\rangle}{1-(1-q)G_1'(1)}]$$

$$(1-q)G_1'(1) \to 1 \Leftrightarrow q_c = 1 - (\frac{\langle k^2\rangle}{\langle k\rangle} - 1)^{-1}$$

Random walk:
visitation probability:

$$\pi^{(t)} = \pi^{(0)} \cdot T^t$$

stationary distribution (Feedback centrality):

$$\pi = \pi \cdot T$$

Eigenvector centrality:

$$\alpha x = x \cdot A$$

PageRank:

$$\alpha x = x \cdot T + \vec{1}$$

total variation distance:

$$\delta(\pi, \pi') = \frac{1}{2}\sum_i |\pi_i - \pi_i'| \simeq \frac{1}{2}\|\lambda_2 a_2 \vec{v_2}\|_1$$

Approximating diffusion speed:

$$t \propto \frac{\log(\epsilon)}{\log(|\lambda_2|)}$$

Laplacian matrix of a network:

$$\frac{dx^{(t)}}{dt} = -C \cdot L \cdot x^{(t)}$$

$$\Rightarrow x^{(t)} = \sum_{i=1}^{n} a_i e^{-C\lambda_i} \vec{v_i} \simeq a_i e^{-C\lambda_2} \vec{v_2}$$

Shannon Entropy:

$$H(X) = -\sum_i P(X=i)\log P(X=i)$$

MapEquation:

$$L(\hat{z}) = qH(Q) + \sum_{i=1}^{k} p_i H(P_i), q + \sum_{i=1}^{k} p_i = 1$$

EGRM:

$$\frac{\partial}{\partial P(G)}\{H + \alpha(1 - \sum_G P(G)) + \sum_i \theta_i(\langle f_i\rangle - \sum_G f_i(G)P(G))\} = 0$$

$$P(G) = \frac{\exp(-\sum_i \theta_i f_i(G))}{Z}$$

$$Z = \sum_G \exp(-\sum_i \theta_i f_i(G))$$

$$\langle f_i\rangle = -\frac{1}{Z}\frac{\partial Z}{\partial \theta_i}$$

Master equation for time-dependent degree:

$$P(k, s, t+1) = \frac{1}{t}P(k-1, s, t) + (1 - \frac{1}{t})P(k, s, t)$$

- initial condition: $P(k, 1, 1) = \delta_{k,0}$
- boundary condition: $P(k, t, t) = \delta_{k,1}$

Time-dependent degree distribution:

$$P(k, t) = \frac{1}{t}\sum_{s=1}^{t} P(k, s, t)$$

$$\Rightarrow (t+1)P(k, t+1) - (t-1)P(k, t) = P(k-1, t) + \delta_{k,1}$$

Limiting (stationary) degree distribution:

$$P(k) = \frac{1}{2}[P(k-1) + \delta_{k,1}] \Rightarrow P(k) = 2^{-k}$$

Preferential attachment:

$$P(v_i) = \frac{k_i(t)}{\sum_j k_j(t)}$$

$$\frac{dk_i(t)}{dt} = \frac{k_i(t)}{2t}$$

$$\Rightarrow k_i(t) = m(\frac{t}{t_i})^{\frac{1}{2}}$$

Stationary degree distribution:

$$P(k) = 2m^2 k^{-3} \propto k^{-3}$$

Maximum time difference δ:

$$\forall i = 0, 1, \cdots, l |t_{i+1} - t_i| \leq \delta$$

Betweenness preference of node v

$$I^v(S; D) = \sum_{s,d} P_{sd}^v \log_2(\frac{P_{sd}^v}{P^v(S=s)P^v(D=d)})$$

- P_{sd}^v: the joint probabilities of S and D.
- $P^v(S)$, $P^v(D)$: the marginal probability of source and destination.
- null model:

$$\hat{P}_{sd}^v = \frac{w(s,v)}{\sum_{s'} w(s',v)} \cdot \frac{w(v,d)}{\sum_{d'} w(v,d')}$$

8 Algorithms

Analyzing a social network using ERGMs. Lecture 12

9 Examples

Lecture 07

	G(n,p)*	G(n,m)*	Watts-strogatz	Ring lattice	Scale free	Stochastic Block Model
|V|	n	n	n			
|E|	$p \cdot \binom{n+1}{2}$	m	sn			
Number of microstates	$2^{\binom{n+1}{2}}$	$\binom{\binom{n+1}{2}}{m}$				
Probability of microstates	With edges m_G : $p^{m_G} \cdot (1-p)^{\binom{n+1}{2}-m_G}$	$1/\binom{\binom{n+1}{2}}{m}$				
Diameter $D \simeq \frac{\log n}{\log\langle k\rangle}$		$\frac{n}{2s}$	$\frac{\log n}{\log(np)}$		$D \simeq \frac{\log\log(n)}{\log(\gamma-2)}$	
Degree distribution	$P(k) = \binom{n}{k} p^k (1-p)^{n-k}$		$(P(k=2s)=1)$			
Average degree	np	$\frac{2m}{n}$	$2s$			
Variance of degree	$np(1-p)$					
Average shortest path length						
Clustering coefficient	$(n \to \infty)$ with p fixed:$C \simeq p$, with np fixed, $C \to 0$			$\frac{3s-3}{4s-2}$		
Diffusion speed $t \propto \frac{\log(\epsilon)}{\log\|\lambda_2\|}$						
MLE Estimate	$\hat{p} = \frac{\langle k\rangle}{n}$					$\hat{M}_{kl} = \frac{E_{kl}}{N_{kl}}$
Entropy	$-\sum_{m=0}^{\binom{n+1}{2}} \binom{\binom{n+1}{2}}{m} P(n,m) \log P(n,m)$, where $P(n,m) = p^m \cdot (1-p)^{\binom{n+1}{2}-m}$	$\log\binom{\binom{n+1}{2}}{m} = \log Z(n,m)$				$\sum_{k\leq l} \log\binom{N_{kl}}{M_{kl}}$
Generating function	$G_1(x) = G_0(x)$					
Molloy-Reed criterion $\frac{\langle k^2\rangle}{\langle k\rangle} > 2$	$np > 1$				$\frac{\zeta(\gamma-2)}{\zeta(\gamma-1)} > 2$	
Critical failure ratio $q_c = 1-(\frac{\langle k^2\rangle}{\langle k\rangle}-1)^{-1} = 1-\frac{1}{G_1'(1)}$	$1-\frac{1}{np}$			$q_c = 1-\frac{1}{2s-1}$	$q_c = 1-(\frac{\zeta(\gamma-2)}{\zeta(\gamma-1)}-1)^{-1}$, $q_c \to 1$	

Tabelle 1: Model and metrics comparisons

Matrix	A: adjacency	T: transition	L: Laplacian
Applicable types of network	directed/undirected/weighted	directed/undirected/weighted	undirected, directed: more complicated
Eigenvalues		$1 = \lvert\lambda_1\rvert \geq \lvert\lambda_2\rvert \geq \cdots \geq \lvert\lambda_n\rvert$ Proof. If the T is aperiodic and irreducible, $\forall i > 1, \lvert\lambda_i\rvert < 1$ spectral gap: $1 - \lambda_2$	$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ Algebraic connectivity: λ_2. $\lambda_2 = 0 \Leftrightarrow$ network is disconnected.
Eigenvalue Sort Ordering	Descending	Descending	Ascending
Eigenvectors		The eigenvector of λ_1: stationary distribution	Fiedler vector: the eigenvector of λ_2: bisect the clusters. Detect communities by iterations.
Centrality	Eigenvector $\alpha x = x \cdot A$, usually not necessarily take the eigenvector of the largest eigenvalue.	Feedback $\pi = \pi \cdot T$ PageRank: 1-unit bonus. $\alpha x = x \cdot T + \vec{1}$	
Applications	D: $D_{ii} = d_{\text{out}}(i)$	Stationary distribution (Feedback centrality) Total variation distance: $\delta(\pi, \pi^{(t)}) \simeq \frac{1}{2}\sum_j \lvert\lambda_2 a_2 \vec{v_2}_j\rvert$, that is $t \propto \frac{\log(\epsilon)}{\log\lvert\lambda_2\rvert}$	Algebraic connectivity, Fiedler vector.

Tabelle 2: Matrices Comparison