

# ネットワーク分析の基礎と実践

津川 翔

筑波大学 システム情報系

# チュートリアルの構成

- 1章：はじめに
- 2章：ネットワークの基礎
- 3章：ネットワークの特徴量
- 4章：中心性解析
- 5章：コミュニティ抽出
- 6章：ネットワークのロバスト性
- 7章：ウィルス拡散
- 8章：Twitterのエゴネットワーク解析
- 9章：MIKAのネットワーク解析

基礎編

実践編

# チュートリアルのレベル感

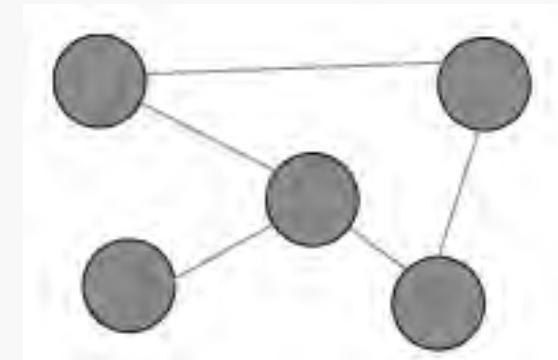
- ネットワーク分析/ネットワーク科学の**背景知識のない人**を対象
  - Python によるプログラミング経験や Unix のリテラシーがあるとよい
  - 学部レベルのグラフ理論の用語が分かっているとなおよい
- 少しでも**背景知識**がある人にとっては（特に前半）かなり退屈
  - 学生や後輩の指導になんらかの参考になれば...

# 第1章：はじめに

---

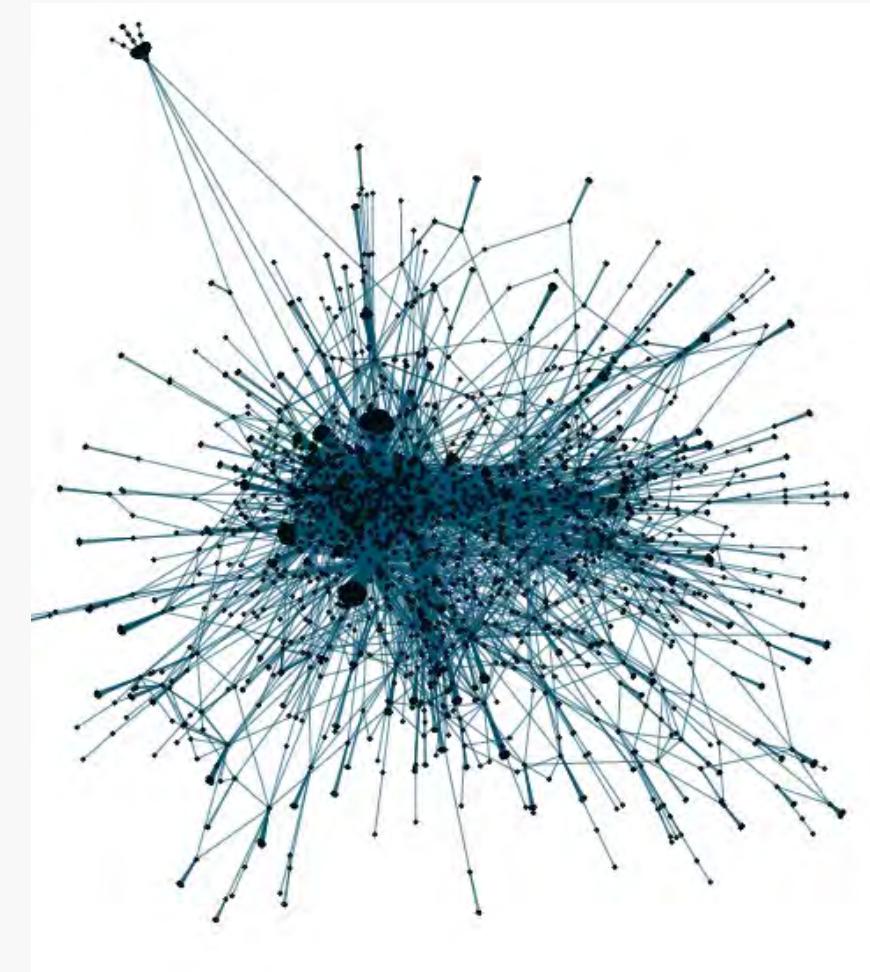
# ネットワークとは？

- なんらかの対象の間のつながりを点と線で表現したもの
- インターネット(通信網)もネットワークの1種
  - 本チュートリアルでは日常会話で意味する「ネットワーク」よりも広い対象をネットワークと呼ぶ



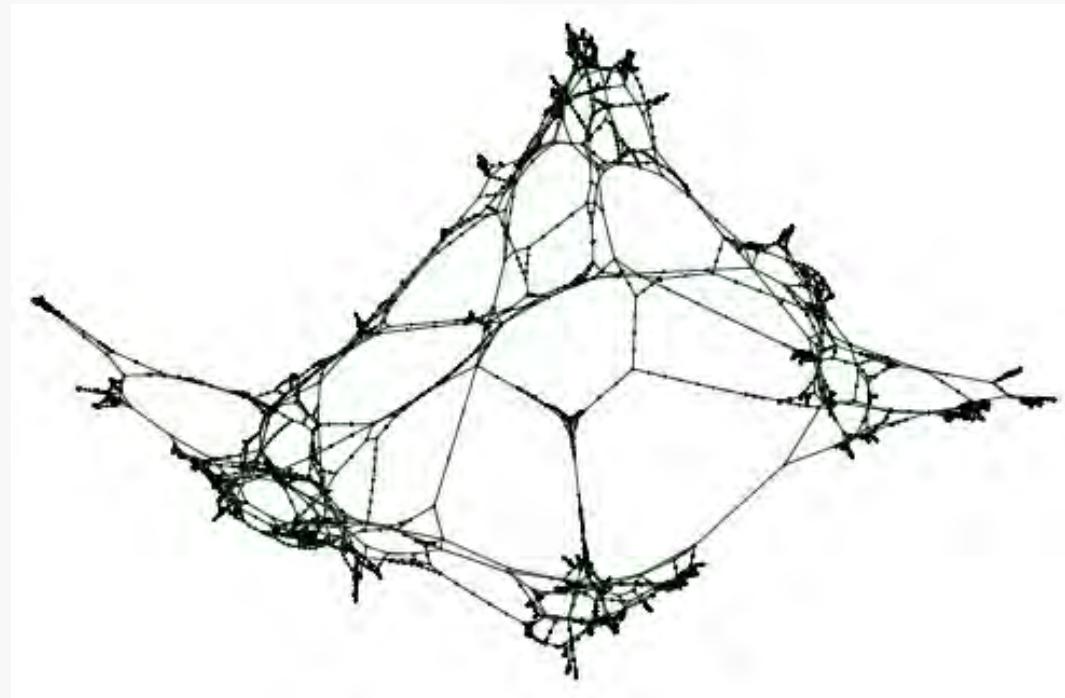
## インターネットのAS間の接続関係

- ノード：AS
- リンク：BGP のリンク



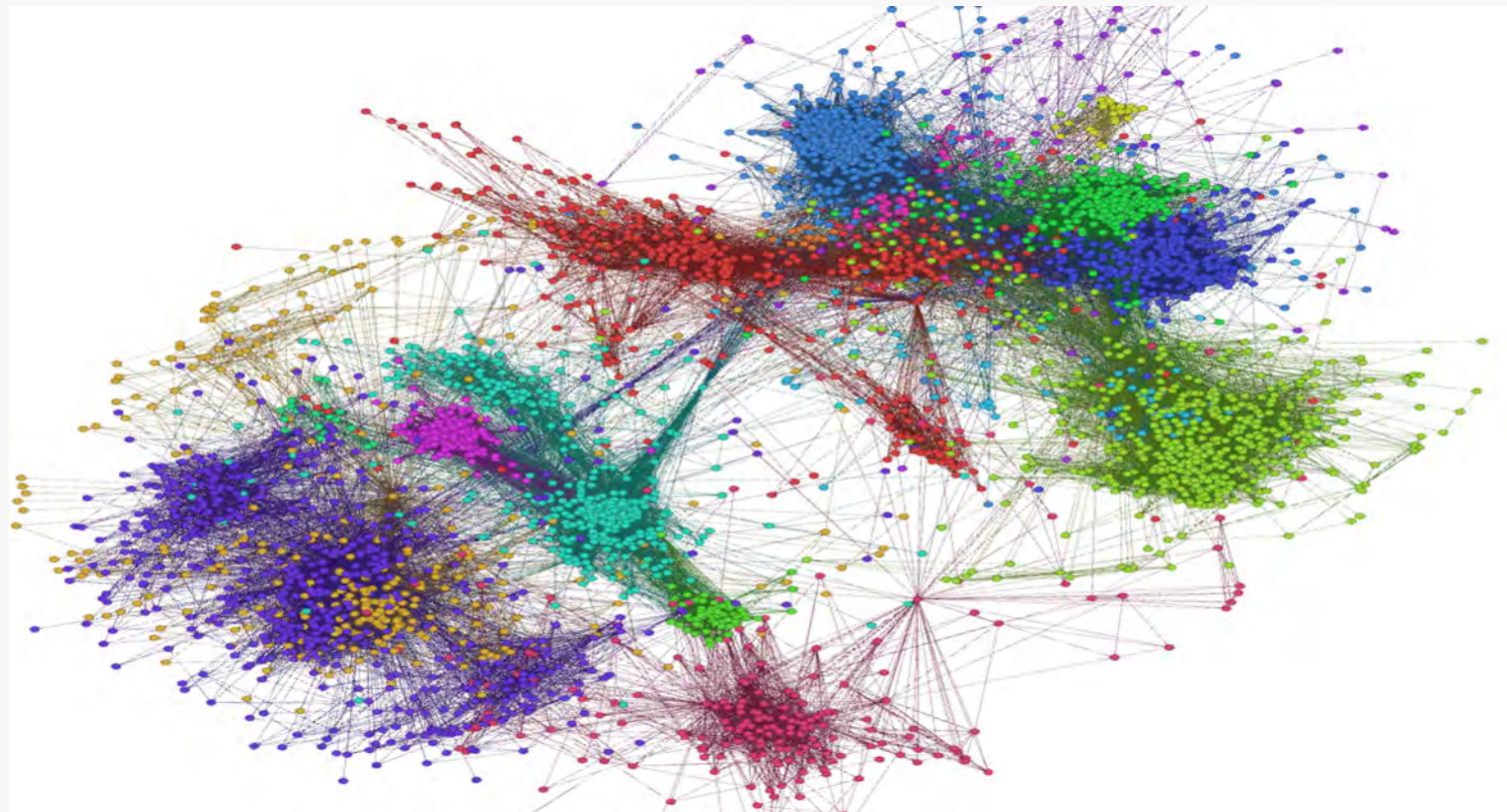
## 電力網

- ノード：発電所、変電所
- リンク：送電線



## SNS の友人関係

- ノード：SNSのユーザ
- リンク：友人関係



# なぜネットワークか？

- 複雑な系を「適度に」抽象化できる
- 数理的あるいは計算機を用いた解析手法を適用できる
- 人の社会での実際の情報拡散の特性：
  - 人の特性やその時々の状況によって決まる
  - 複雑で理解が難しい
- ネットワークとして抽象化した社会での確率的な情報拡散：
  - 解析が可能  
→ コミュニティやインフルエンサーが情報拡散に与える影響が明らかに

# 無線通信とネットワーク科学のギャップ

- ネットワーク科学の興味：  
ノード間の相互作用の結果としての系全体の性質やダイナミクス  
e.g., ネットワーク全体としての効率性・頑健性
- 無線通信の興味：  
ノード間の相互作用そのもの  
e.g., 電波伝播の特性

大規模で複雑なシステム (e.g., 社会、インターネット、CPS)  
の理解には両方の視点が不可欠

# ネットワークを扱う研究分野：ネットワーク科学

## ● 学際的な分野

- 情報科学、物理学、生物学、社会科学、経済学など様々な分野にまたがる
- 数理的な側面：物理分野を中心
- アルゴリズムや実応用の側面：情報分野を中心
- ネットワークを研究対象とする分野の研究者も参入

## ● 1990年代後半から急速に発展

- Wattsらの small world network
- Barabasiらの scale-free network
- Googleの PageRank

## ネットワーク分析の活用例 (pp.6--8)

- ネットワークの頑健性の解析 (6章)
- ウィルス拡散の特性の解析 (7章)
- テロ対策
- 脳機能の解析
- 情報検索
- 情報推薦
- バイラルマーケティング

# 本チュートリアルで扱うこと/扱わないこと

## ● 扱うこと：

- ネットワークの基礎的な概念と基本的な分析手法
- 実践的な（研究との距離の近い）分析例の紹介
- 分析例を自分でなぞるための方法

## ● 扱わないこと：

- 最先端の研究や現状の研究課題
  - サーベイ論文等を参照
- 数理的な解析手法
  - バラバシ本や増田・今野本を参照

# 準備 (1/2)

- GitHub からコード/データをダウンロード
    - <https://qr.paps.jp/LVdn>



s-tugawa / MIKA21\_tutorial Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main ▾ 1 branch 0 tags Go to file Add file ▾ Code ▾

	s-tugawa Update Readme.md	b5b8cb9 6 days ago	27 commits
	Sec3_basic Delete .gitkeep	6 days ago	
	Sec4_centrality Add files via upload	6 days ago	
	Sec5_community Add files via upload	6 days ago	
	Sec6_robustness Add files via upload	6 days ago	
	Sec7_epidemic Add files via upload	6 days ago	
	Sec8_twitter Update Readme.md	6 days ago	
	Sec9_collab Update Readme.md	6 days ago	
	README.md Update README.md	6 days ago	
☰	README.md		✎

MIKA21\_tutorial

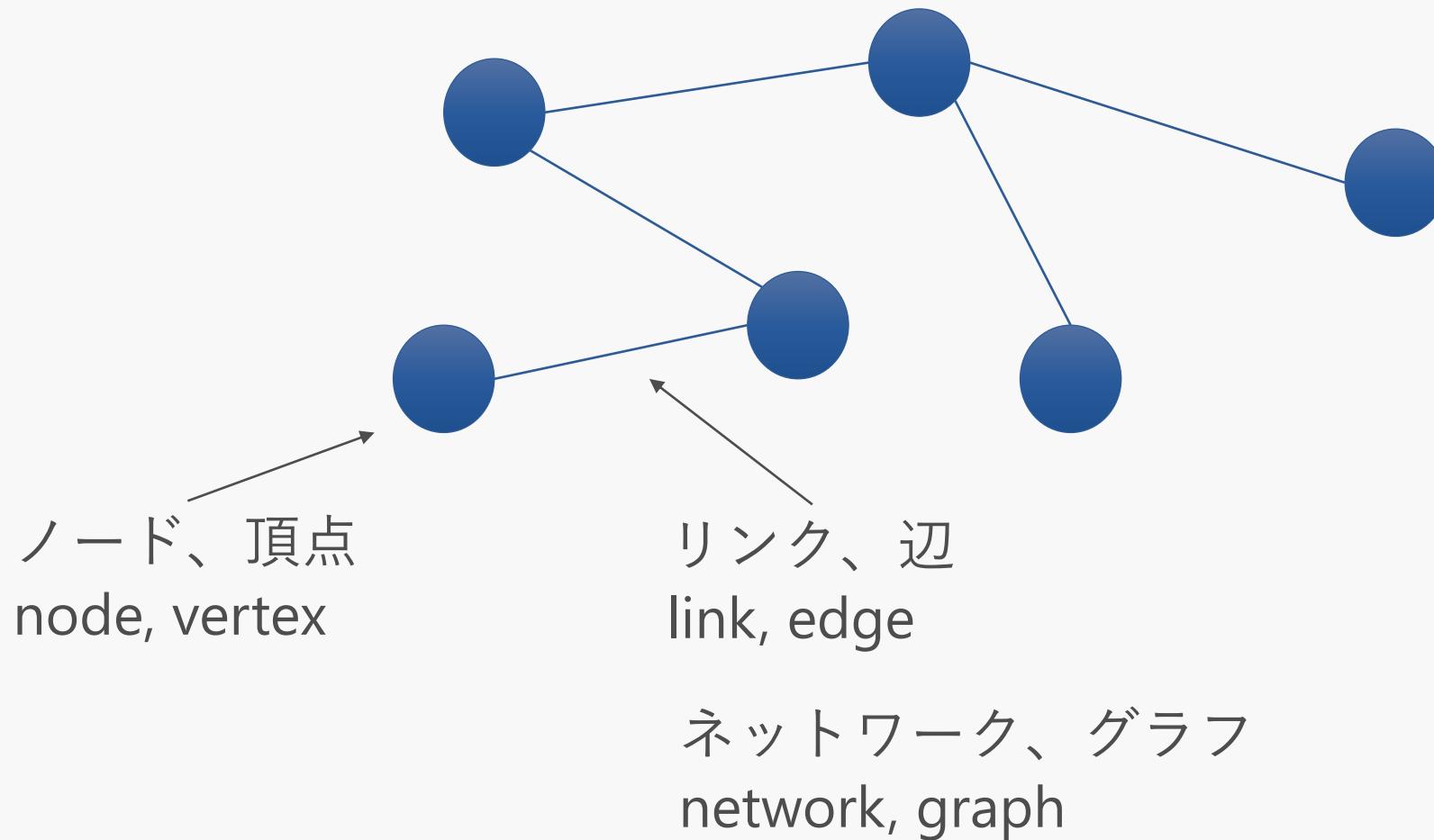
## 準備 (2/2)

- Python バージョン 3 系の実行できる環境を準備
- \$ pip3 install networkx pandas numpy searbor  
を実行

# 第2章：ネットワークの基礎

---

## 基本的な用語



# ネットワークとグラフ

## ● ネットワーク

- 現実に存在するものを指す場合が多い
- 例：www、social network
- ネットワークの用語：ネットワーク、ノード、リンク

## ● グラフ

- ネットワークを数学的に表現したもの指す場合が多い
- 例：web graph、social graph
- グラフの用語：グラフ、頂点、辺

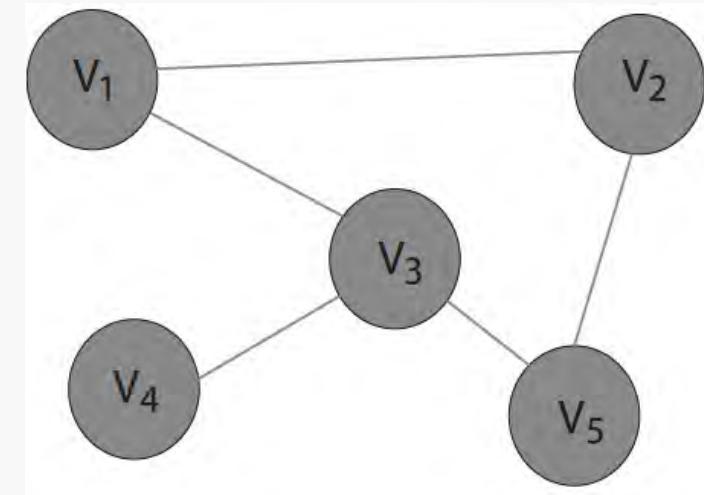
チュートリアルの中では、基本的にネットワークの用語を用いるが、  
グラフの用語を使う場合も同じ意味である

# ネットワークの数学的な表現

- $G = (V, E)$
- $V$ : ノード集合、 $E$ : リンク集合

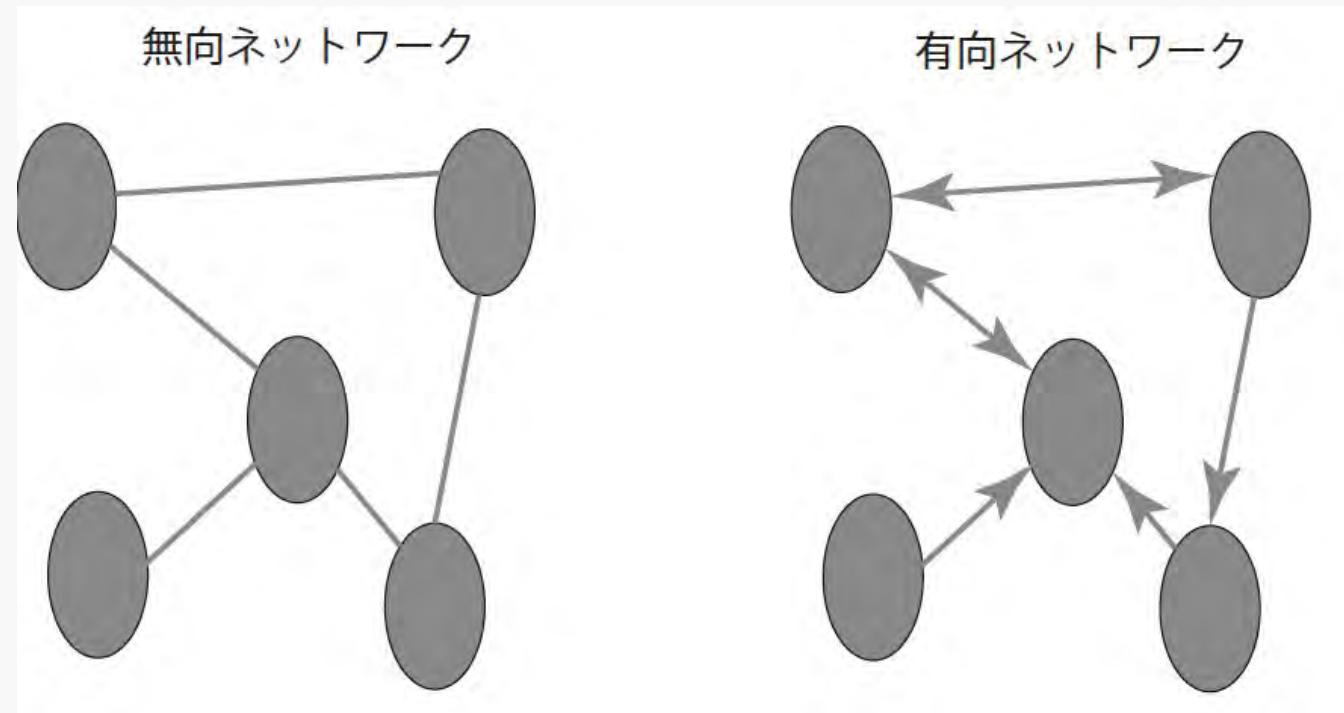
$$V = \{v_1, v_2, v_3, v_4, v_5\}$$

$$E = \{(v_1, v_2), (v_1, v_3), (v_2, v_5), (v_3, v_4), (v_3, v_5)\}$$



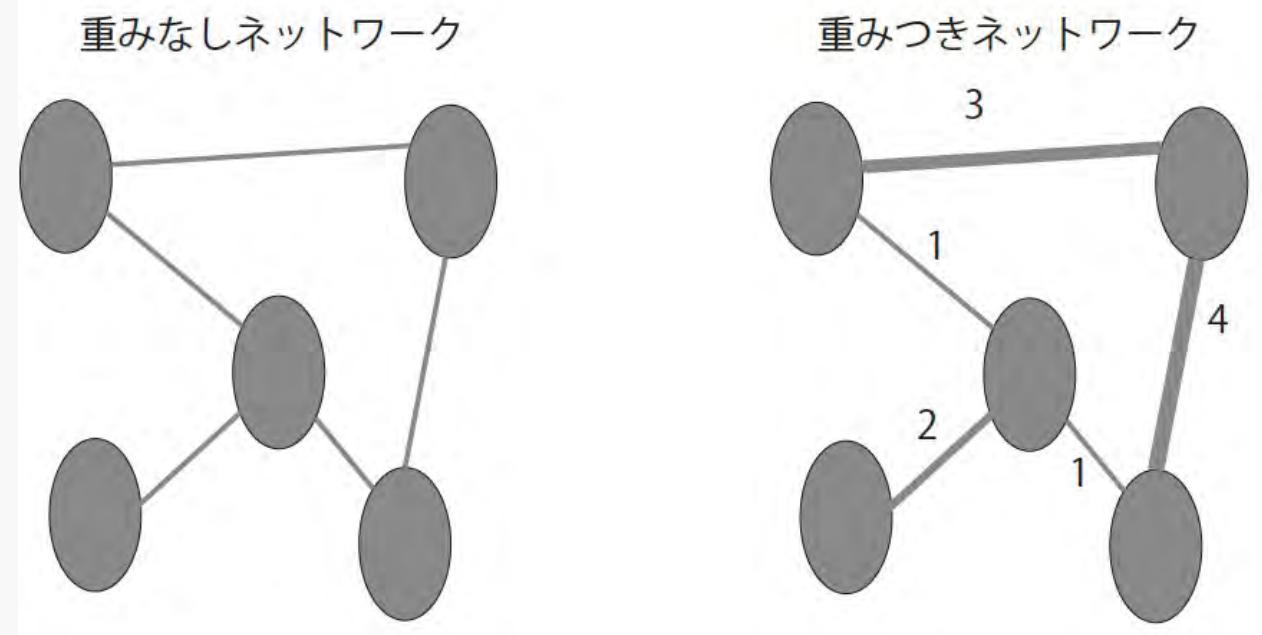
## 無向ネットワークと有向ネットワーク

- 無向リンクの例：共著関係、知人関係
- 有向リンクの例：ハイパーテーリンク、電子メールの送受信



## 重みつきネットワークと重みなしネットワーク

- 重みの例：メールの送信回数、電話の発信回数
- 重みなしの例：ハイパーテリンク、知人関係（知ってる/知らない）



## ネットワークとその表現方法の例

表 2.1: 分析対象のネットワークとその典型的な表現方法の例

ネットワーク	向き	重み
ルータ	無し	リンク容量
知人関係	無し	無し
メールの送受信関係	送信した人から受信した人へ	送信回数
WWW	リンクしたページからされたページへ	無し

## どのネットワーク表現を用いるか？

- データの性質から表現方法が一意に決まる場合もある
- 一般的にはデータを分析の目的に応じて適切な表現方法を用いる
- 例：Twitter のネットワークを分析する場合
  - Follow 関係を有向のリンクで表現
  - 相互 follow 関係を無向のリンクで表現
  - Mention 関係を有向重みつきリンクで表現どの表現方法が適切かは目的に依存する

## 隣接行列とエッジリスト

- 隣接行列：リンク  $(v_i, v_j)$  を行列の  $i$  行、 $j$  列の成分で表現

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

- エッジリスト：全てのリンクを列挙したリスト

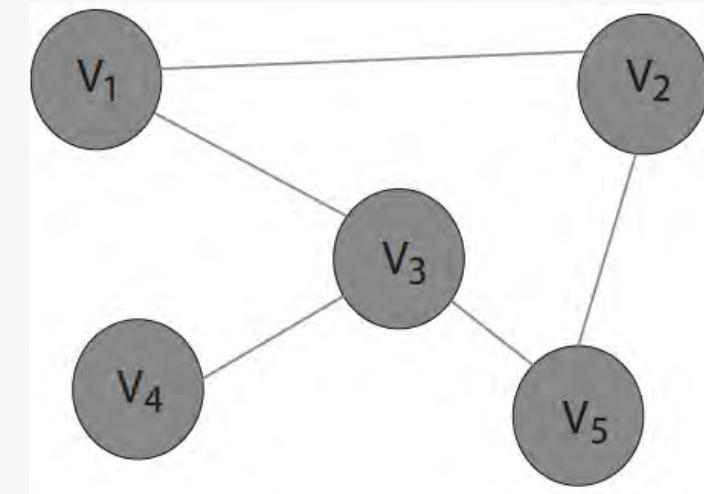
1 2

1 3

2 5

3 4

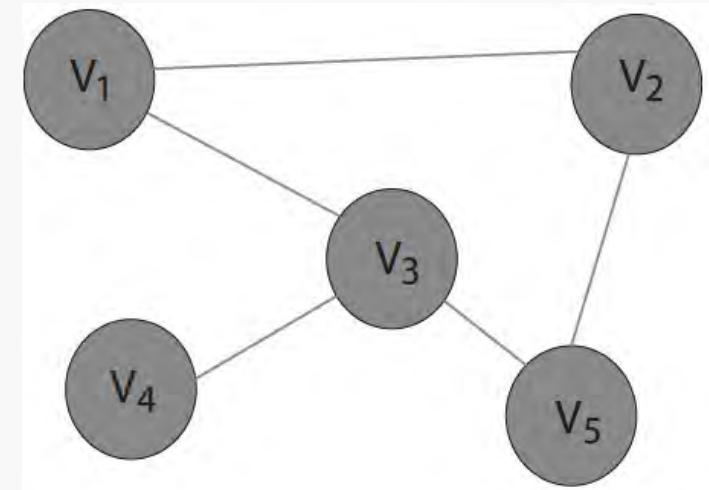
3 5



## 次数 (degree)

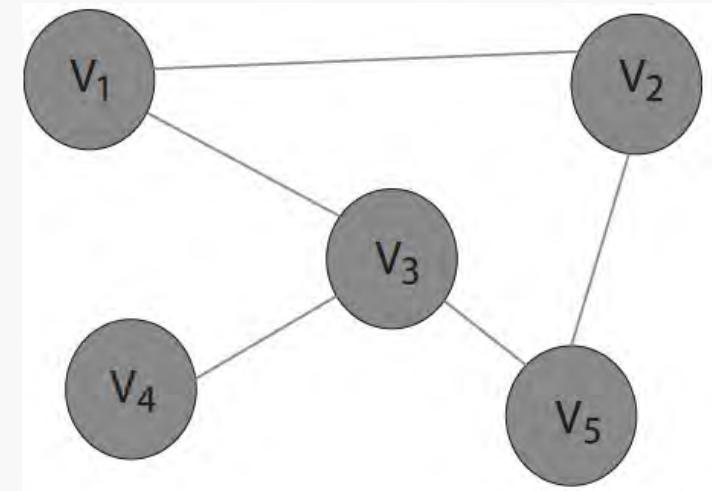
- ノードの有するリンクの数
- 有向ネットワークでは入次数、出次数が定義可能

$$k_1 = 2, k_2 = 2, k_3 = 3, k_4 = 1, k_5 = 2$$



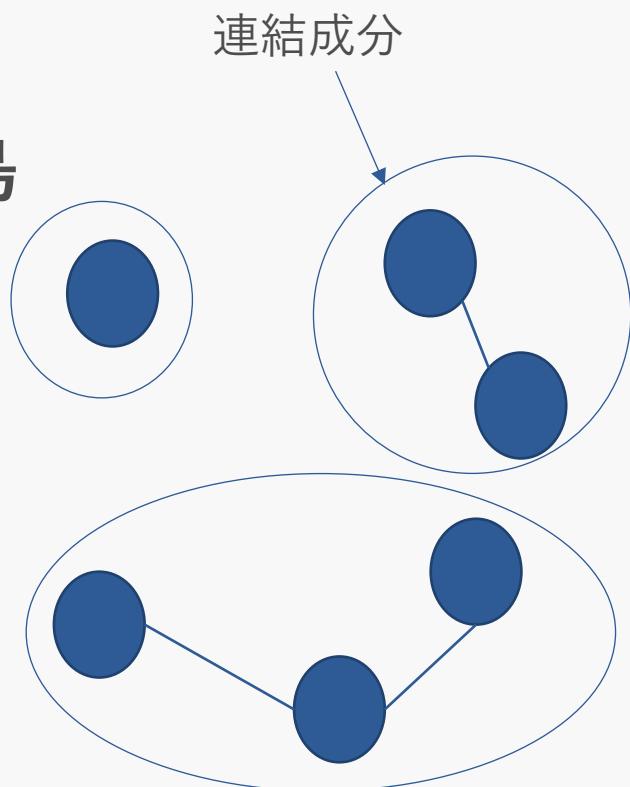
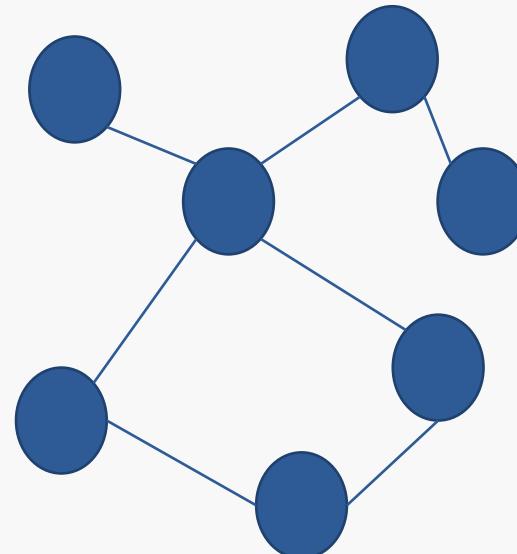
## 経路 (path)

- リンクを辿ってあるノード  $u$  から別のノード  $v$  まで到達できる時、辿ったリンクの列を経路と呼ぶ
- 経路長：経路に出現するリンク数
- 最短経路：ノード  $u, v$  間の経路のうち  
経路長が最短のもの
- 最短経路長（距離）：最短経路の経路長

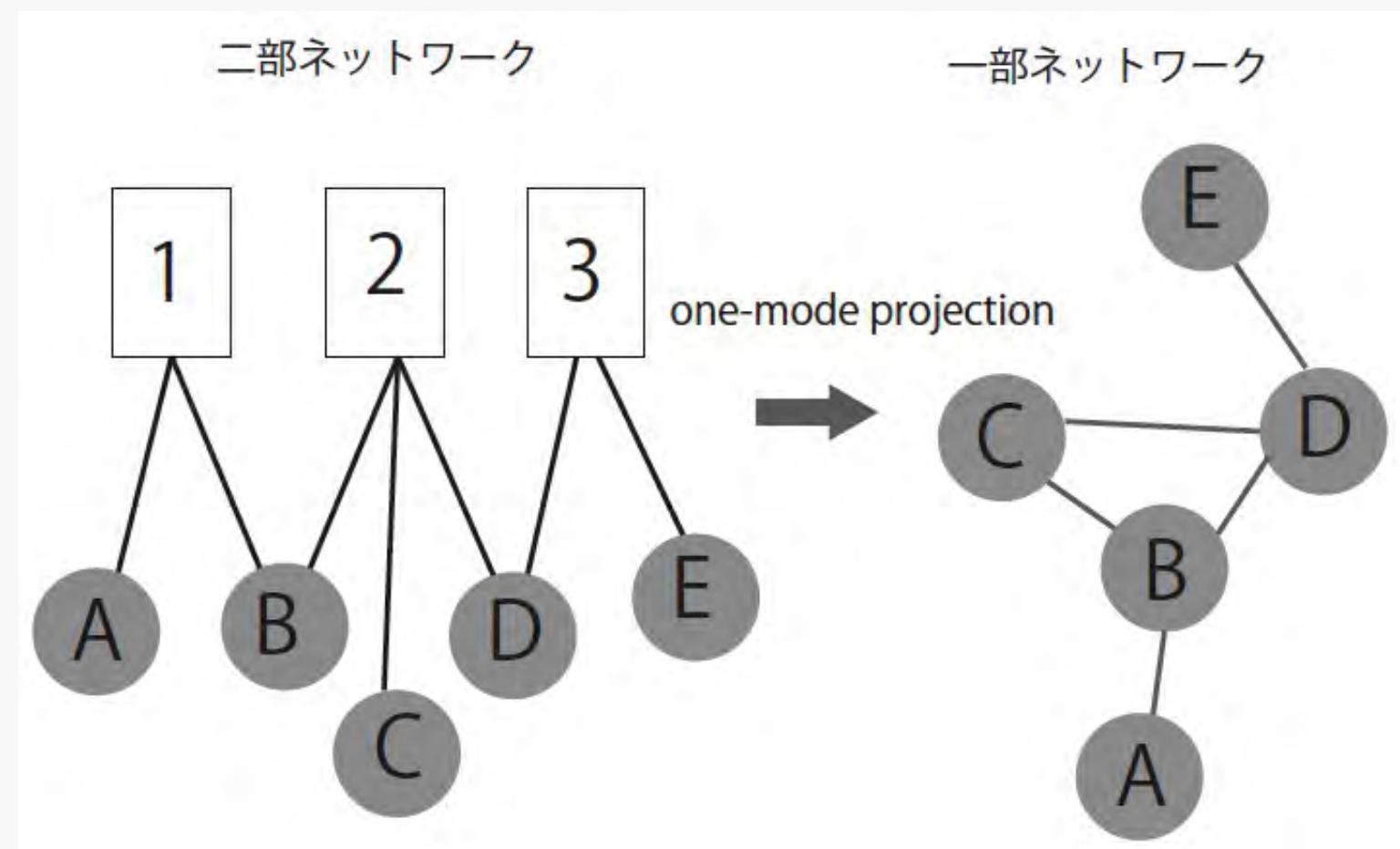


# 連結 (connected)

- 連結：全ノード間に1つ以上の経路が存在
- 非連結：連結でない
- 連結成分：連結でないネットワークのそれぞれの島  
(極大部分ネットワーク)

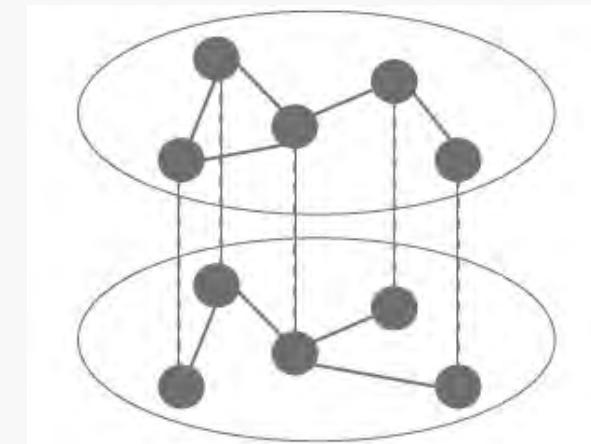
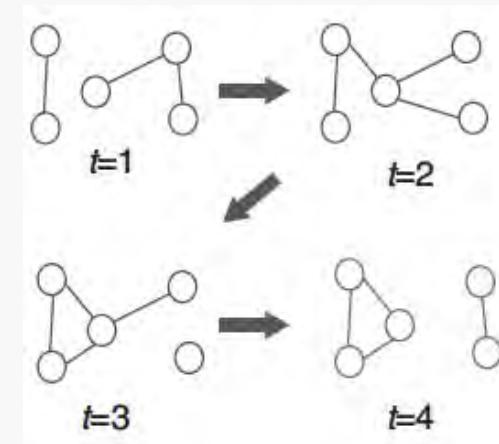


## 二部ネットワーク



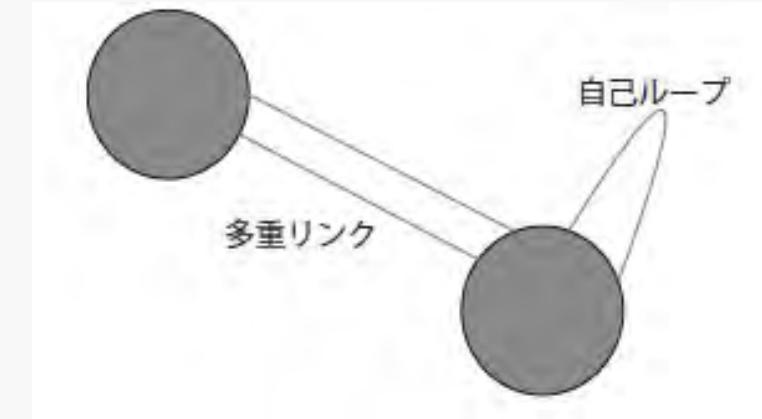
## 様々なネットワークの表現方法

- テンポラルネットワーク
  - 多層ネットワーク
  - 属性付きネットワーク
- など



# 以降で対象とするネットワーク

- 重みなし無向の単純ネットワーク
  - 連結かつ多重リンクおよび自己ループなし
- 一番基本的なケースを扱う
- 重みつき有向には多くの場合拡張可能



## まとめ

- 対象とするシステムをどのようなネットワークで表現するかを決めるのがネットワーク分析の最初の一歩
- ネットワークの表現方法にはリンクの向きの有無、重みの有無などの選択肢がある
- どの表現を用いるかの選択は分析者にある程度自由度があり、その選択によって使える分析手法や分析結果が変わる

# 第3章：ネットワークの特徴量

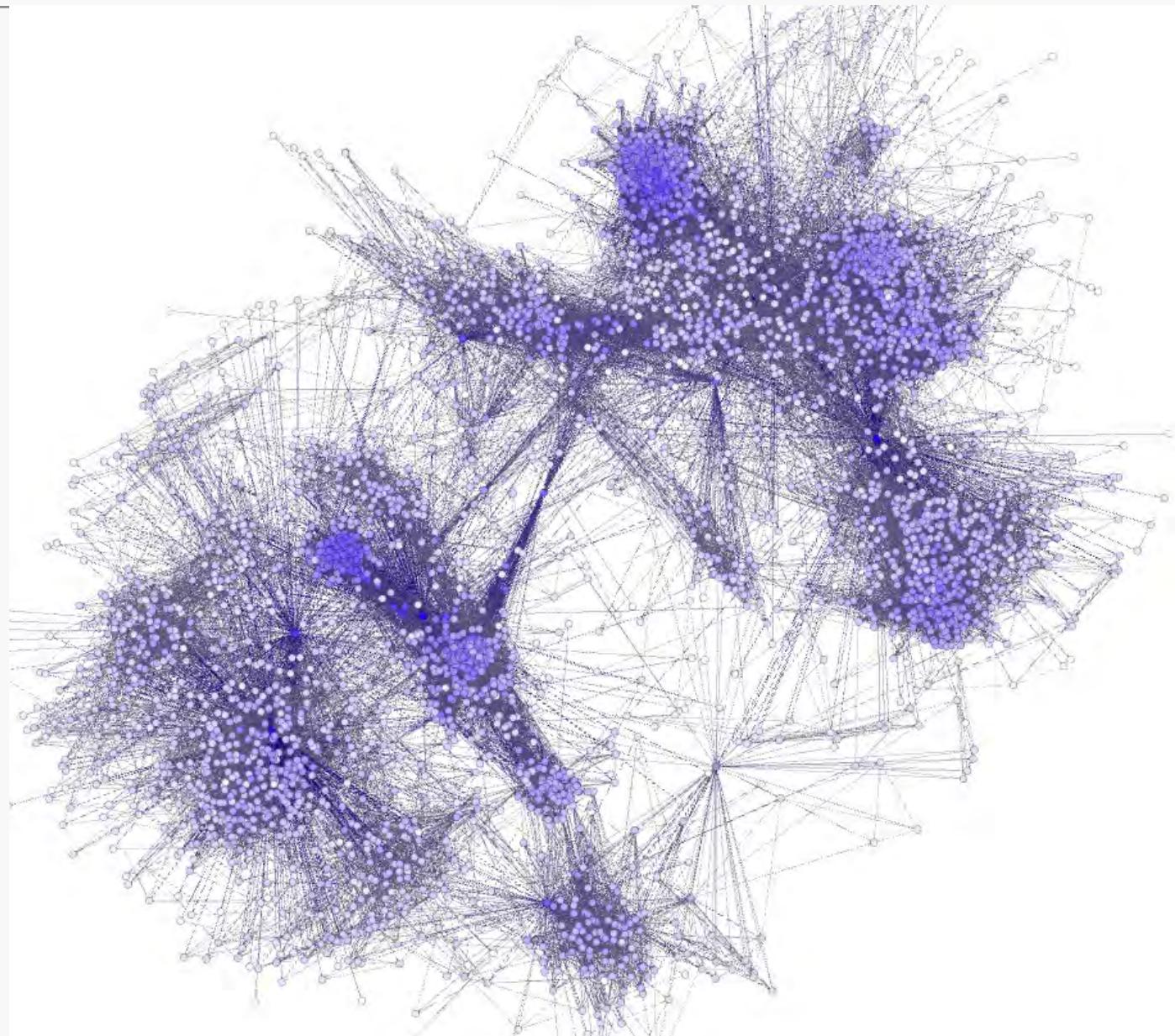
---

## 対象のネットワークがどんな形をしているか？

- 分析対象のネットワークにどんな特徴があるかを調べたい
- ネットワークの特徴を定量化する指標が様々存在する
- しかし、その指標を計算する前に…

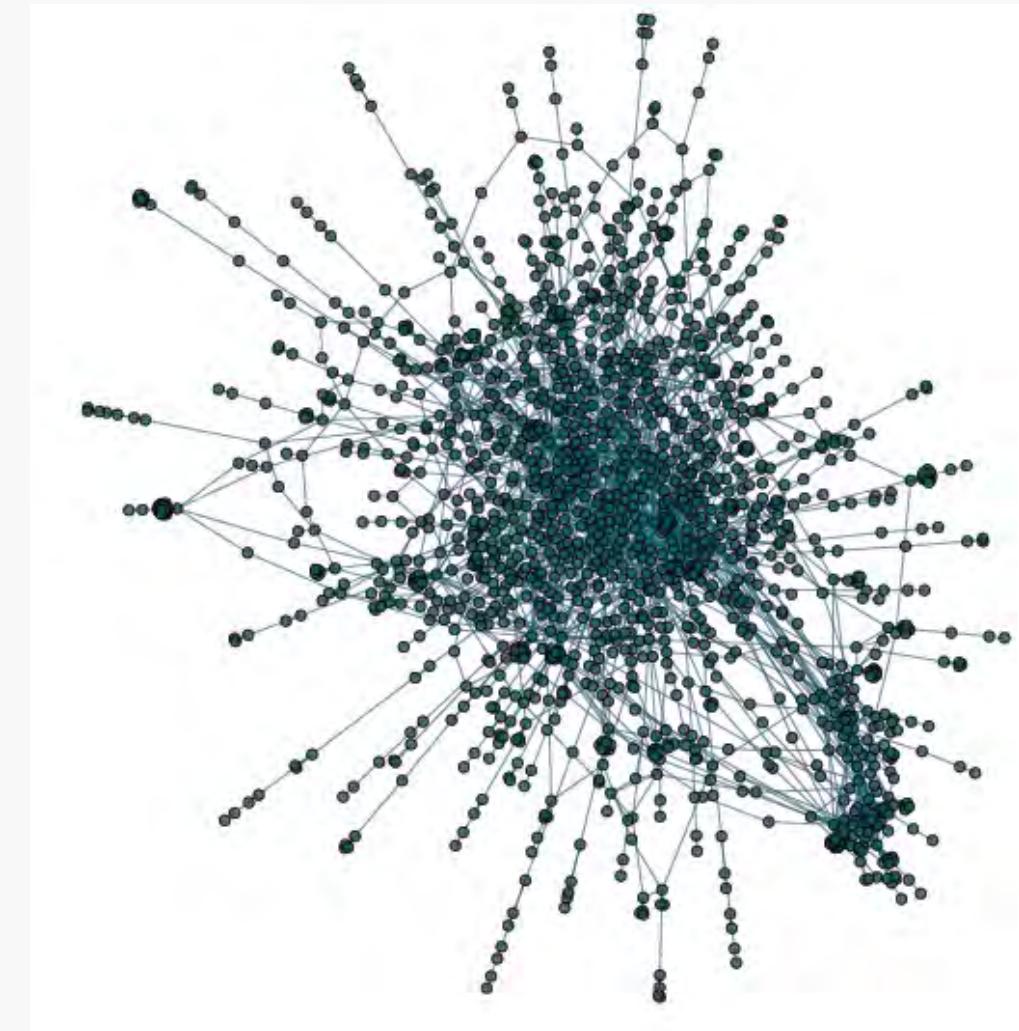
# ネットワークの可視化

- Facebook



# ネットワークの可視化

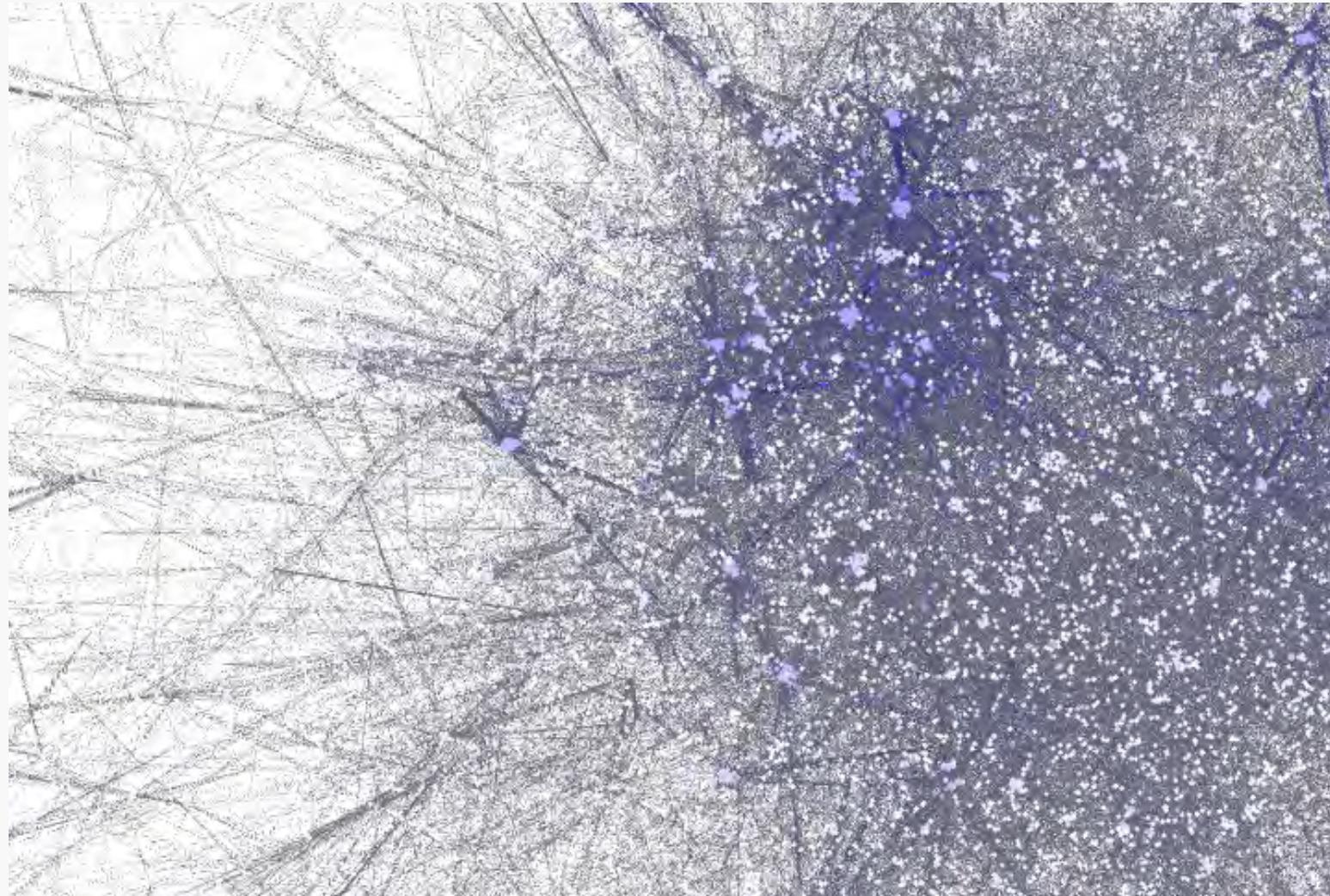
- タンパク質



# ネットワークの可視化

- 論文共著関係
  - 3.7万ノード
  - 23万リンク

大きすぎるとよく  
分からぬ...

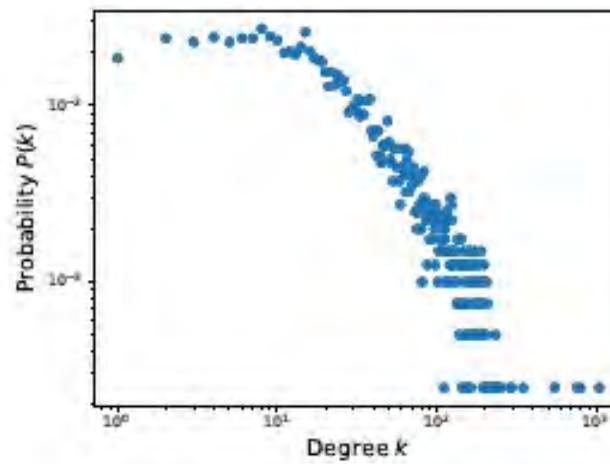


## 次数分布

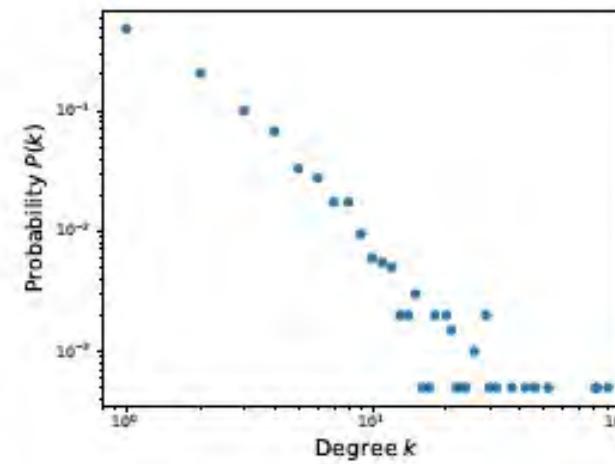
- 次数  $k$  を有するノードの数をノード数で割った値の列

$$\{p(k)\} \equiv \{p(0), p(1), p(2), \dots, p(N - 1)\}$$

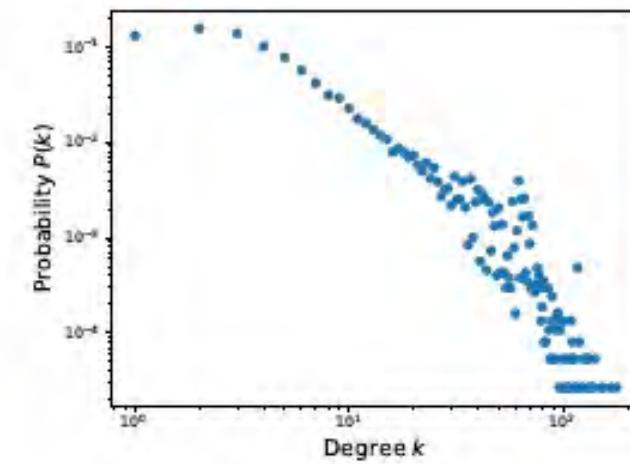
- 現実のネットワークは  $p(k) \propto k^{-\gamma}$  (べき分布) に近いことが多い



(d) Facebook (対数)



(e) タンパク質 (対数)



(f) 共著関係 (対数)

## 平均次数と密度

- 平均次数：次数のネットワーク全体での平均

$$\langle k \rangle = \frac{1}{N} \sum_{i=1}^N k_i = \frac{2M}{N}$$

- 密度：

$$\frac{2M}{N(N - 1)}$$

## 平均最短経路長（平均距離）と直径

- 平均距離：全ノードペアの最短経路長の平均

$$L = \frac{2}{N(N-1)} \sum_{1 \leq i < j \leq N} d(v_i, v_j)$$

- 直径：最短経路長の最大値
- 現実のネットワークは平均距離も直径も短い
  - c.f., 6次の隔り (six degree of separation)、small world

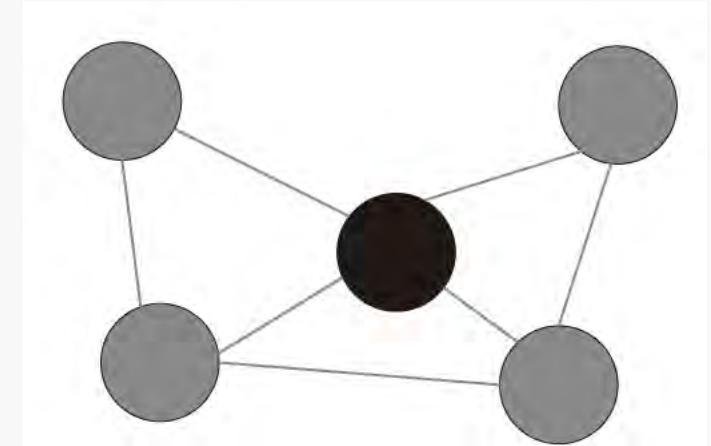
## クラスタリング係数

- ネットワーク中の三角形の多さを表す
- ノード $v_i$  のクラスタリング係数：

$$C_{v_i} = \frac{\sum_{j,k \in \Gamma(v_i)} A_{j,k}/2}{k_i(k_i - 1)/2} = \frac{\sum_{j,k \in \Gamma(v_i)} A_{j,k}}{k_i(k_i - 1)}$$

- ネットワークのクラスタリング係数：

$$C = \frac{1}{N} \sum_{i=1}^N C_{v_i}$$



# ネットワークの特徴量の計算

- \$ python3 calc\_feature.py hoge.txt

```
import networkx as nx

#平均次数
deg_seq = list(dict(G.degree()).values())
avedeg=np.average(deg_seq)
#ノード数
len(G.nodes())
#密度
nx.density(G)
#クラスタリング係数
nx.average_clustering(G)
#平均経路長
nx.average_shortest_path_length(G)
#直径
nx.diameter(G)
```

# 次数分布の例

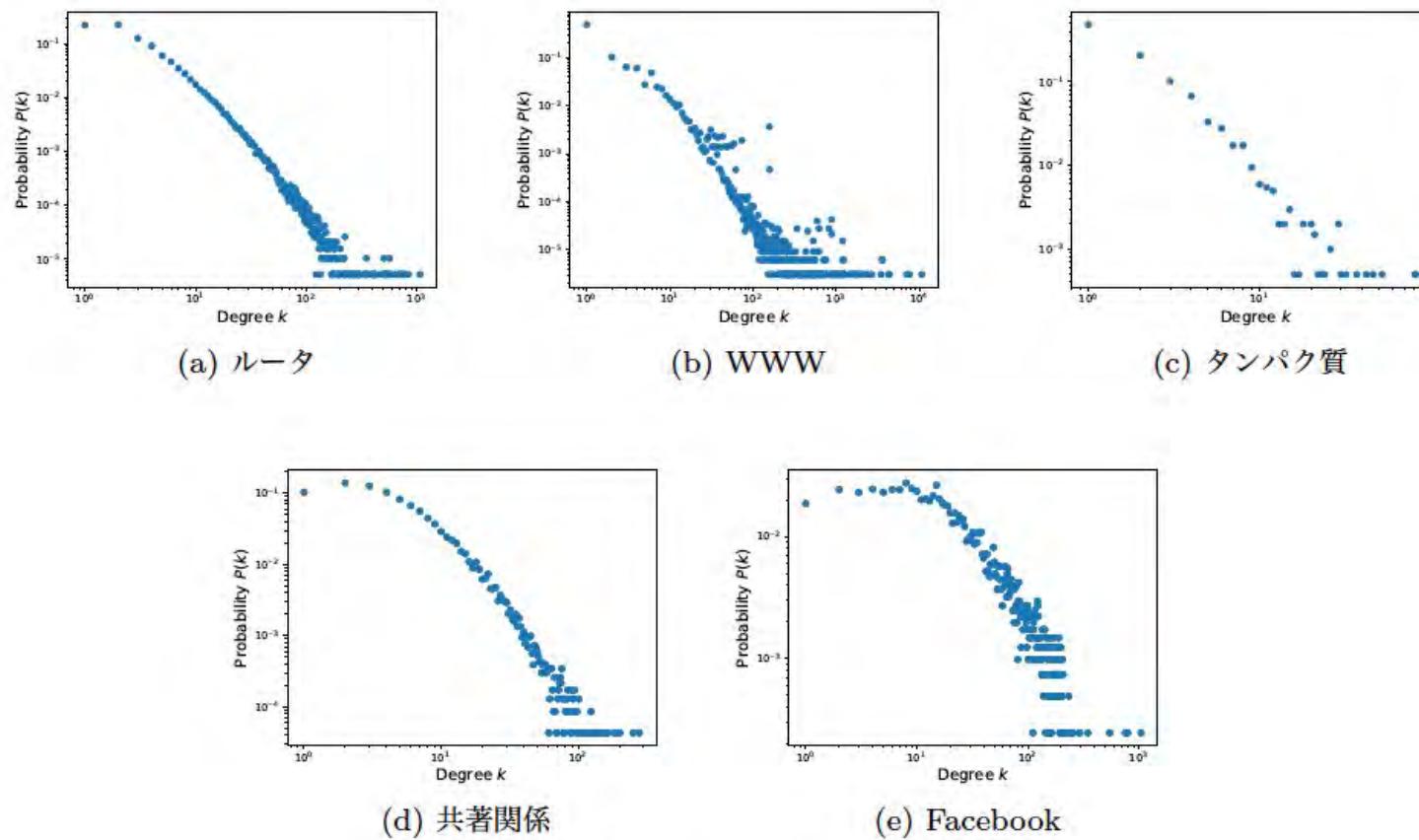


図 3.5: 様々な種類のネットワークの次数分布

## 特徴量の例

	ルータ	WWW	タンパク質	共著関係	Facebook
ノード数	192,244	325,728	2,018	23,133	4,039
平均次数	6.33	6.86	2.90	8.07	43.7
クラスタリング係数	0.16	0.23	0.05	0.63	0.61
連結成分の数	308	1	185	567	1
最大連結成分の割合	0.99	1.0	0.81	0.92	1.0
平均最短経路長	6.98	7.17	5.61	5.35	3.69

## まとめ

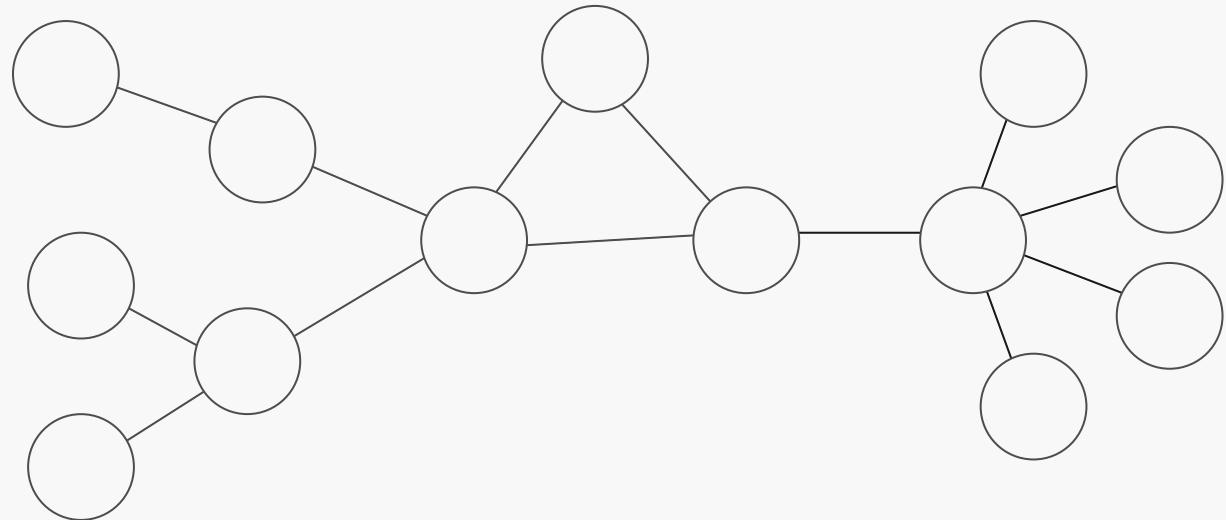
- 分析対象のネットワークの特徴を知るにはまずは可視化が有効
- 大規模なネットワークや分析対象のネットワークの定量比較にはネットワークの特徴量の計算が有用
- 紹介したネットワークの特徴量はネットワーク上のダイナミクスにも影響する (e.g., ウィルス拡散)
- ネットワーク分析の研究ではお作法的に次数分布やその他の特徴量を計算して論文に載せることが多い

# 第4章：中心性解析

---

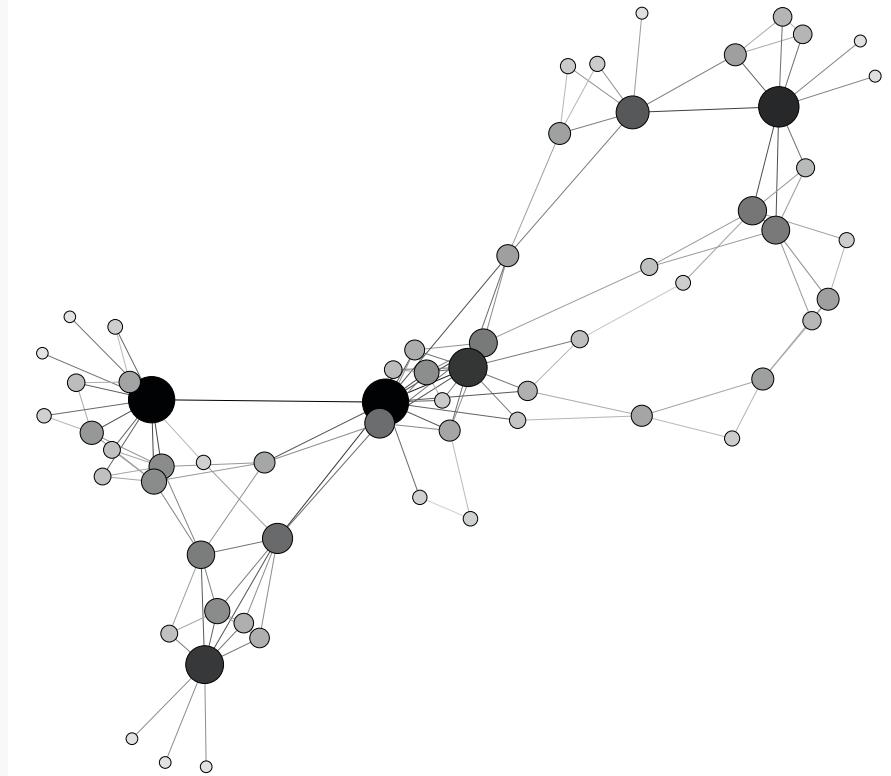
# ネットワークの中心を見つける

- Freeman 1979 の時代
  - ソーシャルネットワークの重要人物の特定
- Brin and Page 1998の時代
  - 重要な Web ページを見つける (Google PageRank)



# 中心性解析の応用例

- バイラルマーケティング
- 感染症対策
- テロ対策
- ネットワークのrobust性向上



# 本チュートリアルで紹介する中心性

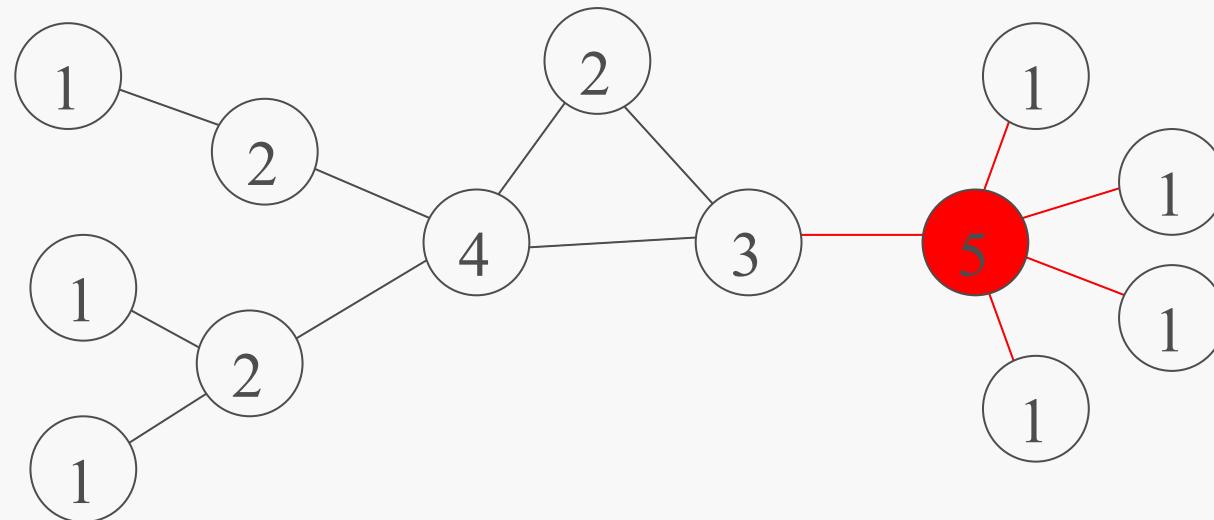
- 次数中心性
- 近接中心性
- 媒介中心性
- 固有ベクトル中心性
- PageRank
- コアネス (k-コア指標)

# 次数中心性

- ノードの次数で定義される

$$C_{\text{deg}}(i) = k_i$$

- 多くのつながりを有するノードが中心であるという考え方に基づく指標

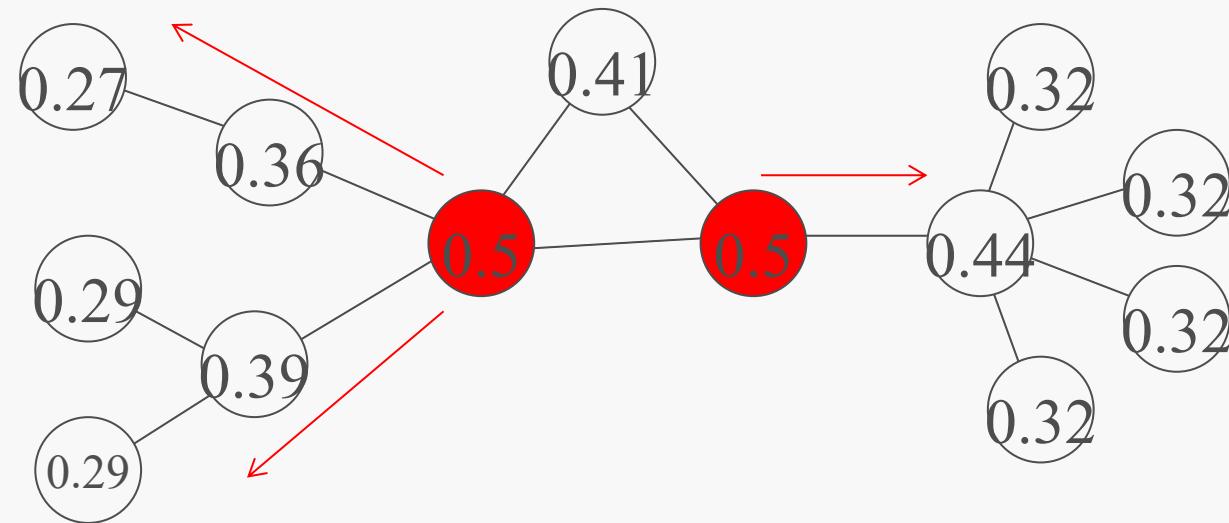


# 近接中心性

- ノード*i*の近接中心性は次式で定義される

$$C_{\text{clo}}(i) = \frac{1}{\sum_j d_{i,j}}$$

- ただし  $d_{i,j}$  はノード  $i$  から  $j$  への最短経路長
  - 短い距離で各ノードに到達できるノードが中心であるという考え方に基づく指標



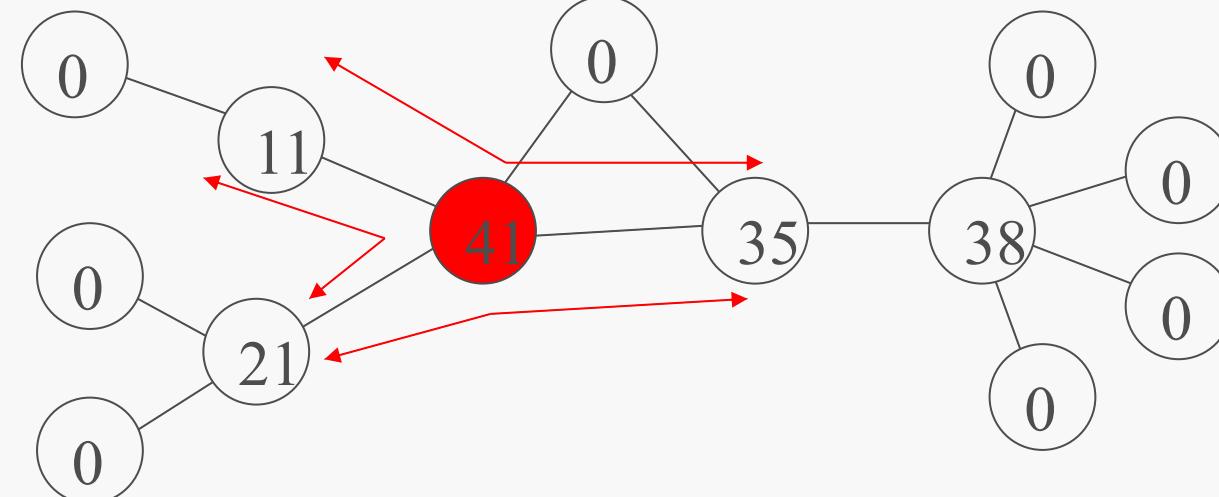
# 媒介中心性

- ノード  $i$  の媒介中心性は次式で定義される

$$C_{\text{bet}}(i) = \frac{\sum_{s,t \in V, s \neq t \neq i} g_{st}(i)}{\sum_{s,t \in V, s \neq t} g_{st}}$$

- 分母はノード  $s$  から  $t$  への最短経路の数
- 分子はノード  $s$  から  $t$  への最短経路のうち  $i$  を通る経路の数

- ノード間の関係を媒介しているノードが中心であるという考え方に基づく指標



# 固有ベクトル中心性

- 中心性の高いノードに隣接しているノードは中心であるという考え方に基づく

$$C_{\text{eig}}(i) = \frac{1}{\lambda} \sum_j^N A_{ij} C_{\text{eig}}(j)$$

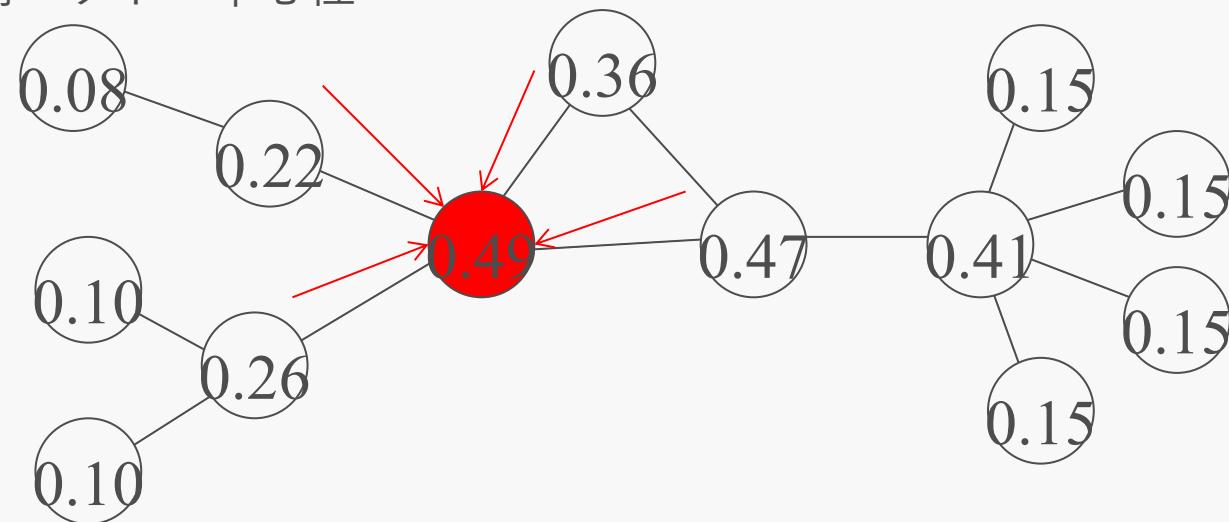
↑ 隣接行列  
↑ ノード i の固有ベクトル中心性

$$\mathbf{C} = \frac{1}{\lambda} \mathbf{A} \mathbf{C}$$

ベクトル、行列形式

$$\lambda \mathbf{C} = \mathbf{A} \mathbf{C}$$

A の固有値が  $\lambda$ 、  
C が固有ベクトルになっている

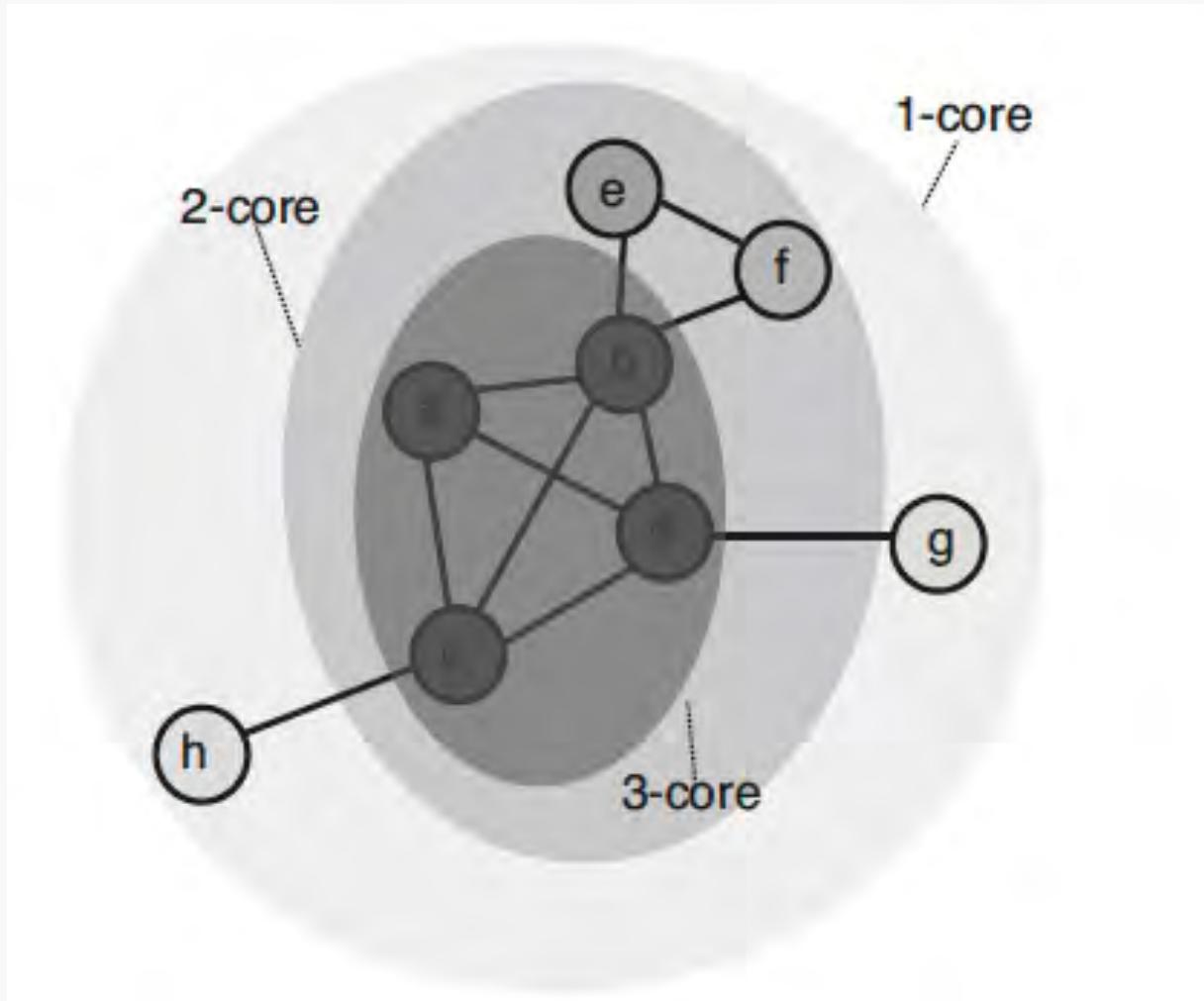


# PageRank

- 固有ベクトル中心性と同様の概念に基づく
- 連結でないグラフや有向グラフにも利用可能

$$PR(v) = \frac{1 - \alpha}{N} + \alpha \sum_{u \in \Gamma(v)} \frac{PR(u)}{k_u^{\text{out}}}$$

## コアネス (k-コア指標)



## 分析例：ハイテク企業の管理職のネットワーク

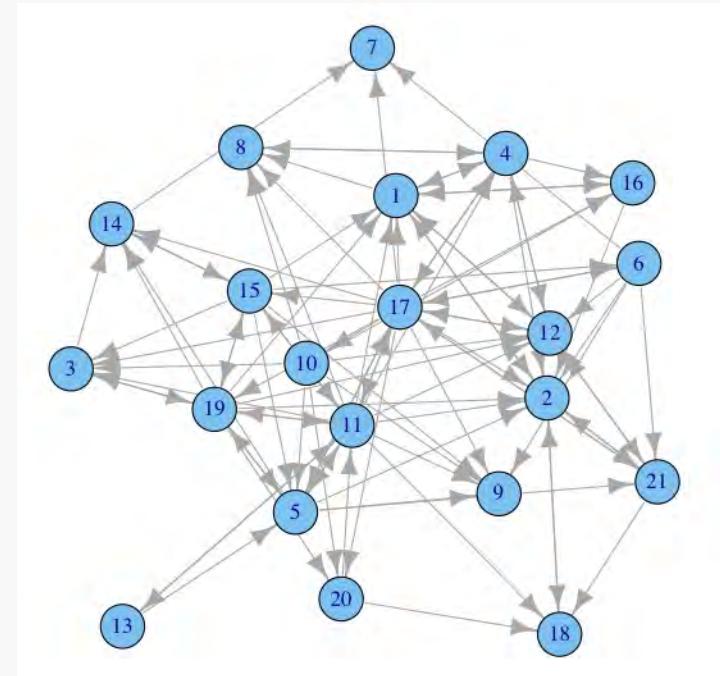
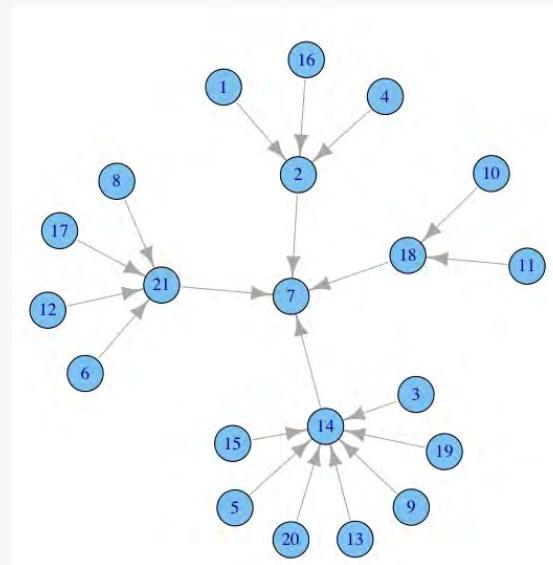
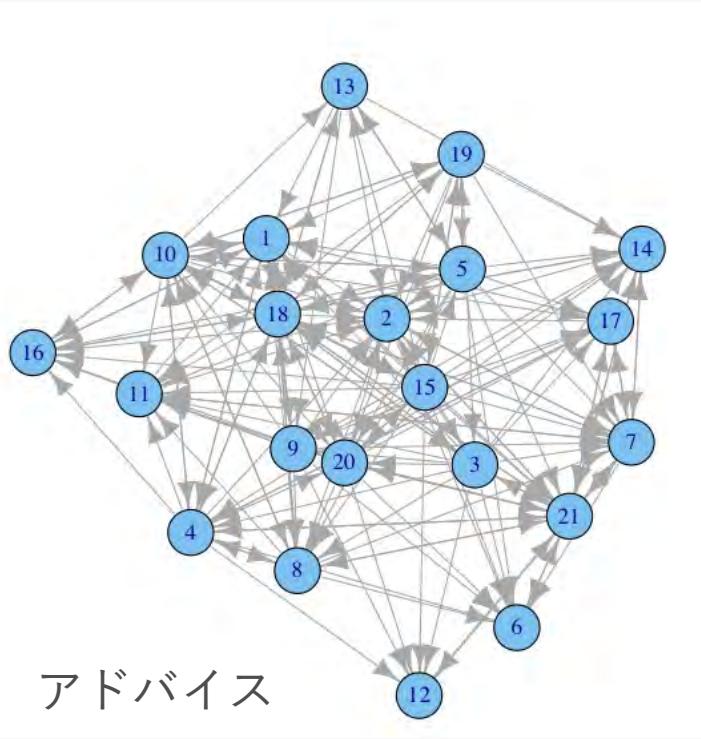
- ハイテク企業の管理職 21 人のネットワーク

- 「仕事上のアドバイスを求める」
- 「友人である」
- 「報告をする」

という 3 種類のリンクが存在

- 出展： Stanley Wasserman and Katherine Faust. Social Network Analysis: Methods and Applications. Cambridge University Press, 1994

# ネットワークの可視化



報告

## 各ネットワークの中心人物の特定

- \$ python3 calc\_centrality.py high\_tech/advice.txt
- \$ python3 calc\_centrality.py high\_tech/friend.txt
- \$ python3 calc\_centrality.py high\_tech/report.txt

## 入次数に基づく順位

入次数に基づく順位	アドバイス	友人	報告
1	2	2	14
2	18	1	7
3	21	12	21
4	1	5	2
5	7	9	18

## 研究者ネットワークの解析

- ある研究分野の権威を中心性を使って抽出する
- 抽出した権威がどの程度確からしいか評価したい
- 研究者の論文の被引用数と中心性の関係を調査する

# データセットと分析の再現方法

- 応用物理分野の共著ネットワーク
- 各研究者の citation 数
- 約24万ノード程度

チュートリアル資料中にある論文の共著関係ネットワークの中心性と論文の被引用数の関係を求めるには、以下のようにプログラムを実行します。

```
python3 calc_cent_citation.py aps/apscoauthor.csv aps/citation_count.txt >result.txt
```

ペアプロットは、以下のようにプログラムを実行すると、pairplot.epsという名前でファイルが出力されます。かなり点数が多いので、時間がかかるのに注意。

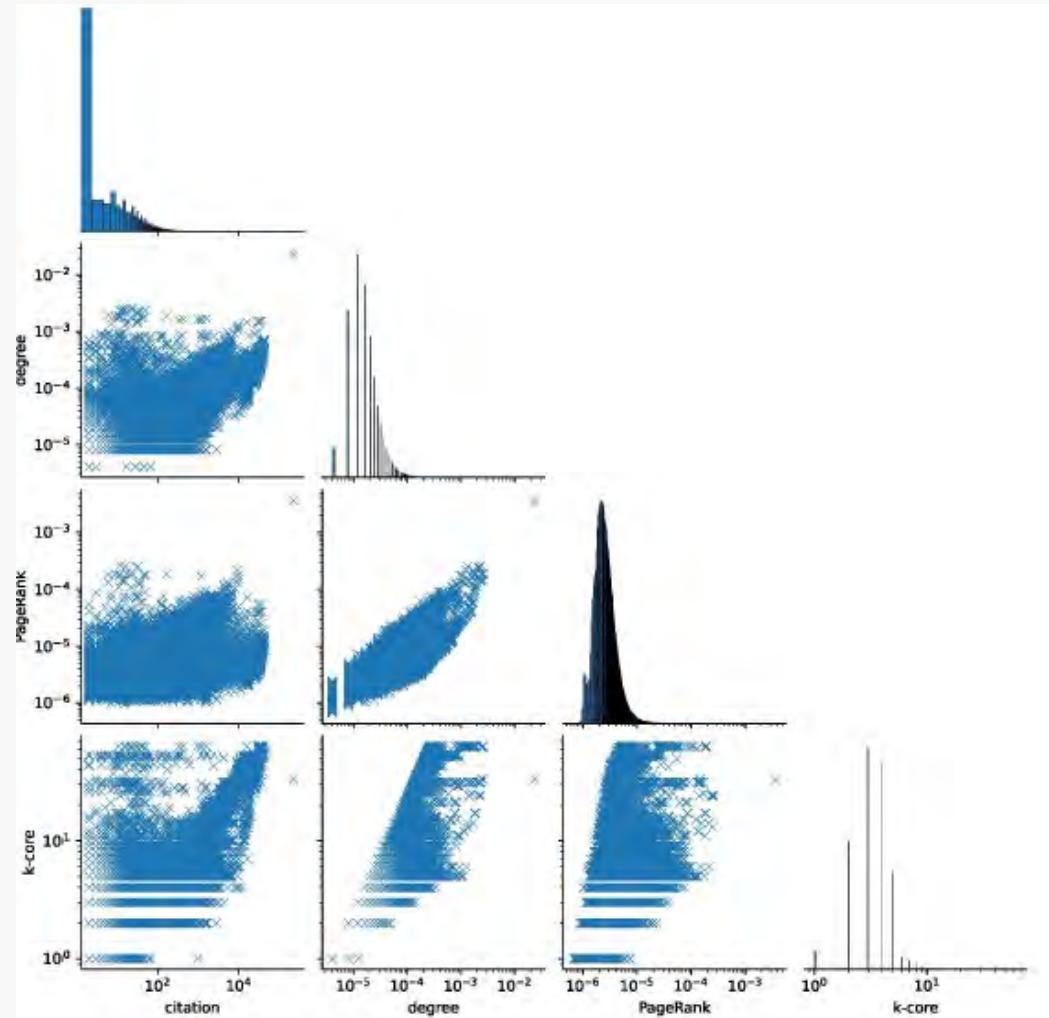
```
python3 pair_plot.py result.txt
```

# 分析結果：被引用数と中心性

$r=0.45$

$r=0.32$

$r=0.69$



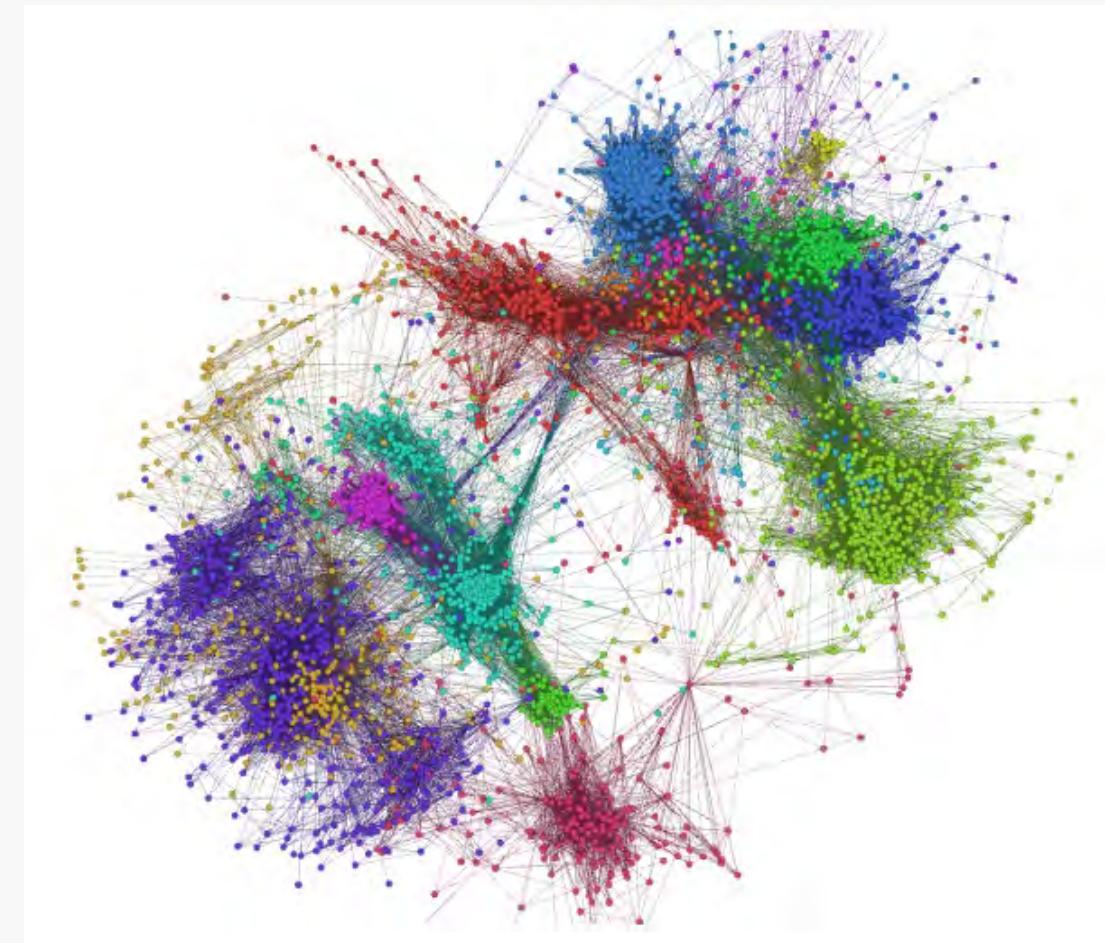
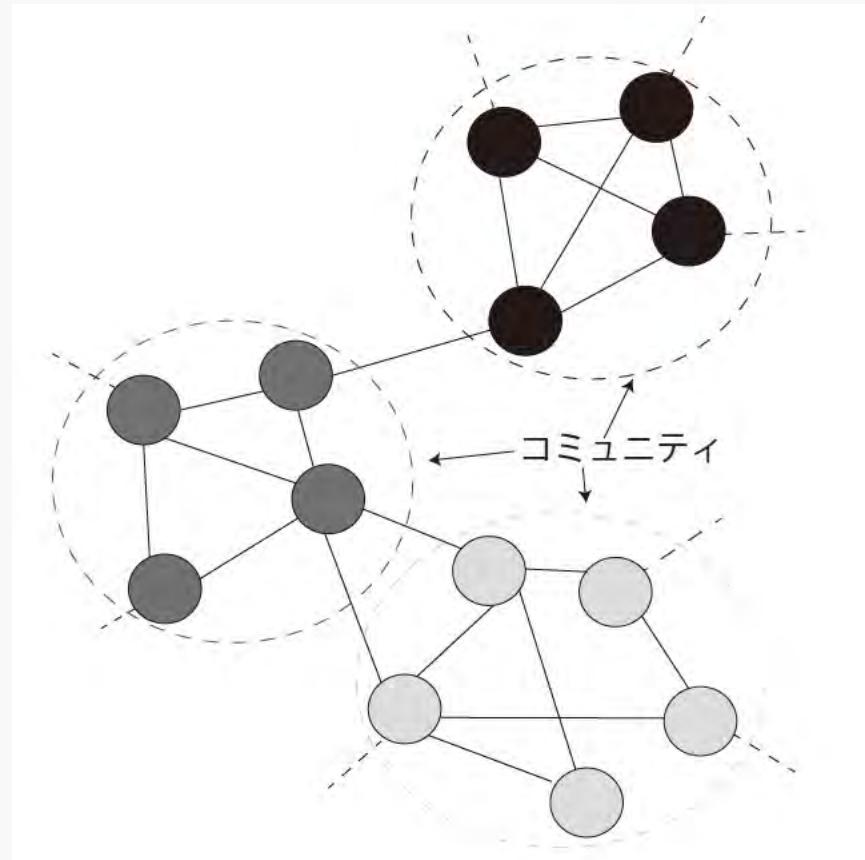
## まとめ

- 中心性解析は古くから研究されている重要なタスク
- 応用範囲は幅広い
- 本チュートリアルで紹介した中心性は今でも研究に用いられる
- 実践的な中心性の計算例を紹介
  - 企業のコミュニケーションパターンの解析
  - 研究者ネットワークにおける権威の特定

# 第5章：コミュニティ抽出

---

# ネットワークのコミュニティ構造



## コミュニティの定義

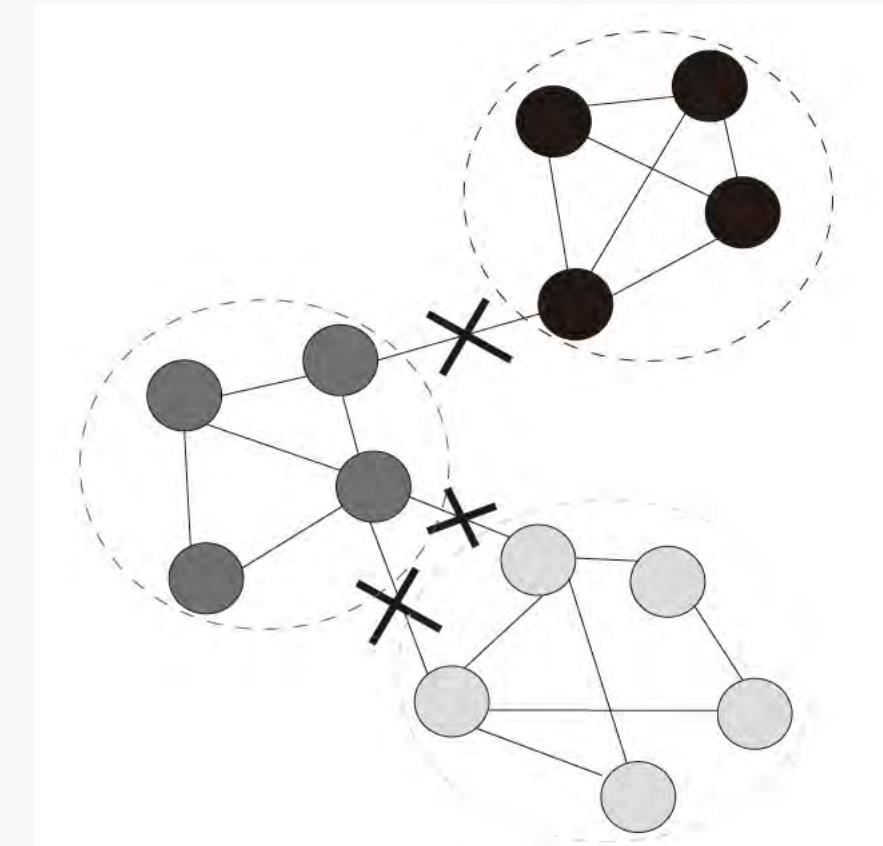
- 厳密な合意の取れた定義はない
- 以下の 2 つの性質は広く合意されている
  - 「コミュニティ内のリンクが密」
  - 「コミュニティ間のリンクは疎」

## コミュニティの抽出

- 中心性解析と並んでネットワーク分析の中心的タスク
  - ネットワークからコミュニティを抽出できれば…
    - 情報推薦
    - 情報拡散の予測
    - 脳機能の推定
- などに有用

# Girvan-Newman 法

- コミュニティ抽出問題発展のきっかけ
- 「コミュニティの境界らしい」リンクを切斷する



# リンクの境界らしさの定量化とGN法のアルゴリズム

## ● リンクの媒介中心性：リンクを通過する最短経路の数

$$C_{\text{bet}}(e) = \frac{\sum_{s,t \in V, s \neq t \neq i} g_{st}(e)}{\sum_{s,t \in V, s \neq t} g_{st}}$$

Girvan Newman 法

1. 各リンクの媒介中心性を計算する。
2. ネットワークから媒介中心性が最大のリンクを削除する。
3. リンクを削除したネットワークで媒介中心性を再計算し、媒介中心性が最大のリンクを削除する。
4. ステップ 3 を所望のコミュニティ数（連結成分の数）に分かれるまで繰り替えす。

# Girvan Newman 法の性質

- ノード数の3乗のオーダーの計算量
  - 大規模ネットワークには使えない
- コミュニティの数を事前に決めておかないといけない  
→ これに対する解決法を Girvan と Newman が提案

# コミュニティらしさを評価する指標 モジュラリティ

- モジュラリティの定義：

コミュニティ $i$ のノード同士をつなぐリンクの数

$$Q(C) = \sum_{c_i \in C} \left( \frac{e_{c_i}}{M} - \left( \frac{a_{c_i}}{2M} \right)^2 \right)$$

コミュニティ $i$ に属するノードの次数の和

- モジュラリティが大きいコミュニティ分割が良い分割と考える

## モジュラリティ最大化によるコミュニティ抽出

- モジュラリティがコミュニティの良さを表すのであれば、直接的にモジュラリティを最大化すればよい
- ただし、厳密な最大化は NP 困難
- 様々なヒューリスティックが提案される
- 代表的な Newman 法と Louvain 法を紹介

# Newman 法

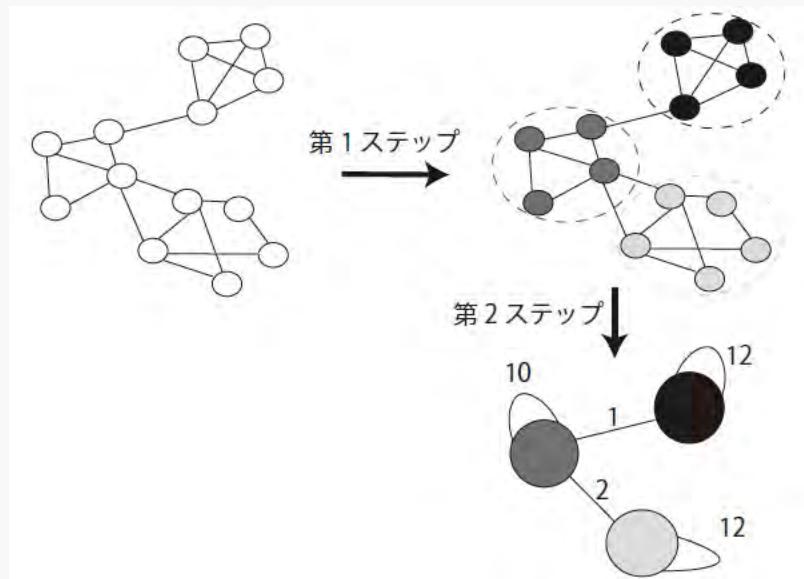
- 全ノードが1つのコミュニティの状態からスタート
- モジュラリティの増加が最大となるように2つのコミュニティを貪欲的に併合

— Newman 法 —

1. 各ノードそれぞれを一つのコミュニティとする (ノード  $V = \{v_1, v_2, \dots, v_N\}$  をそれぞれ  $C_1 = \{c_1, c_2, \dots, c_N\}$  に割り当てる)。
2. 現在のコミュニティ分割  $C_t$  に属する全コミュニティの組合せについて、コミュニティを併合した時のモジュラリティの増加分  $\Delta Q(i, j)$  を求める。
3.  $\Delta Q(i, j)$  が最大となるコミュニティ  $c_i$  と  $c_j$  を併合することでコミュニティ分割  $C_{t+1}$  を得る。
4. ステップ 2、3 を全コミュニティが併合され1つのコミュニティとなるまで繰り返す。
5. モジュラリティ  $Q(C_t)$  が最大となる  $C_t$  を出力する。

# Louvain法

- 第1ステップ：各ノードをモジュラリティの増加が最大となる隣接ノードのコミュニティに併合
- 第2ステップ：コミュニティを1つのノードとして集約



# Louvain 法のアルゴリズム

## — Louvain 法 —

全ノードがそれぞれ異なるコミュニティに所属する状態からスタートし、モジュラリティが上昇しなくなるまで以下を繰り替えす。

### 1. (局所的なモジュラリティの最適化)

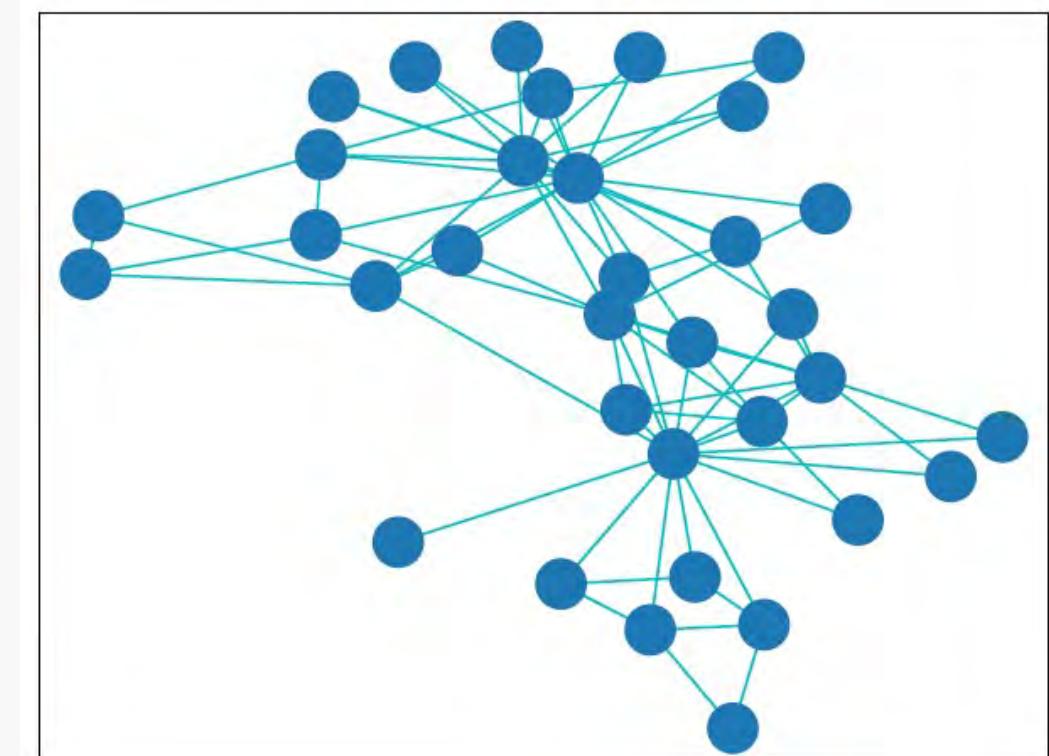
- 各ノード  $i$  を隣接するノードと同じコミュニティに併合した際のモジュラリティの増分  $\Delta Q$  を求め、 $\Delta Q > 0$  が最大となるコミュニティに、ノード  $i$  を併合する。
- 上記の手順を全ノードについて、コミュニティの変化が発生しなくなるまで繰り返す

### 2. (コミュニティの集約) 1つのコミュニティを構成する全てのノードを1つのノードに集約したネットワークを構築する。集約前のコミュニティ内リンク数の2倍の重みを持つ自己ループを集約後のネットワークの各ノードに生成する。また、集約後のノード $i$ 、 $j$ 間のリンクの重みは、集約前のコミュニティ $i$ 、 $j$ 間のリンク数に等しい値とする。集約したネットワークに対して、ステップ1(局所的なモジュラリティの最適化)を実施する。

## コミュニティ抽出をめぐる議論 (p.58)

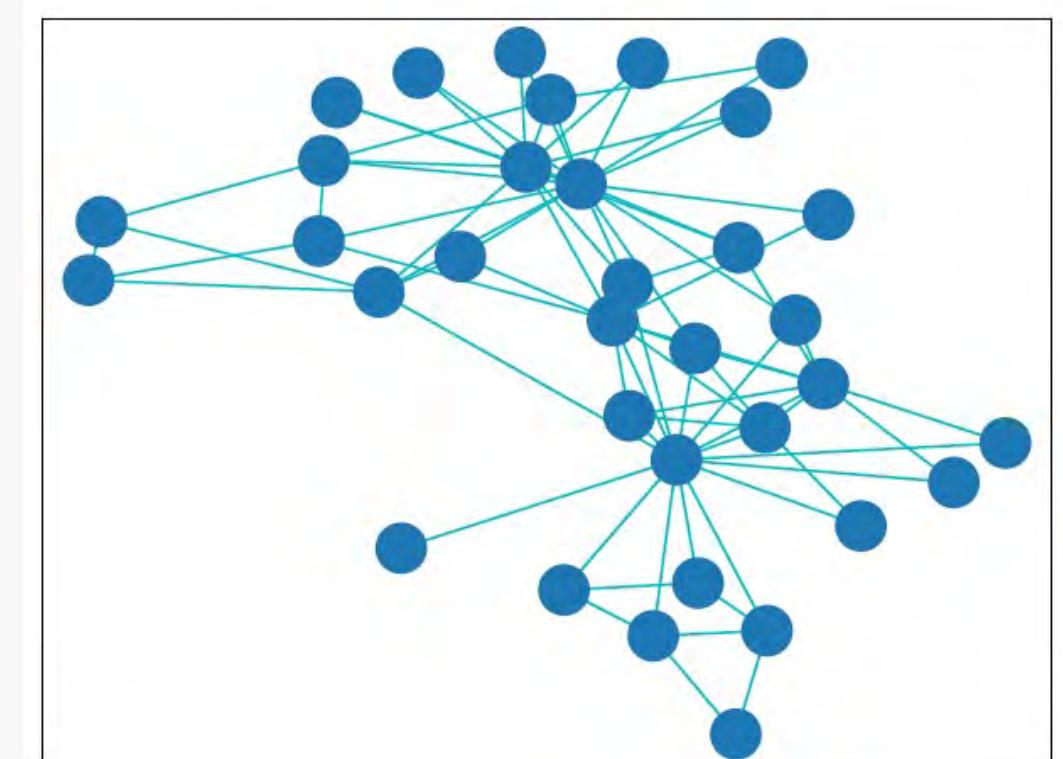
- モジュラリティ最大化はそもそも良くないという主張も存在
- 経験上、明確なコミュニティ構造があれば、モジュラリティ最大化は悪くない選択肢

# コミュニティ抽出の実践



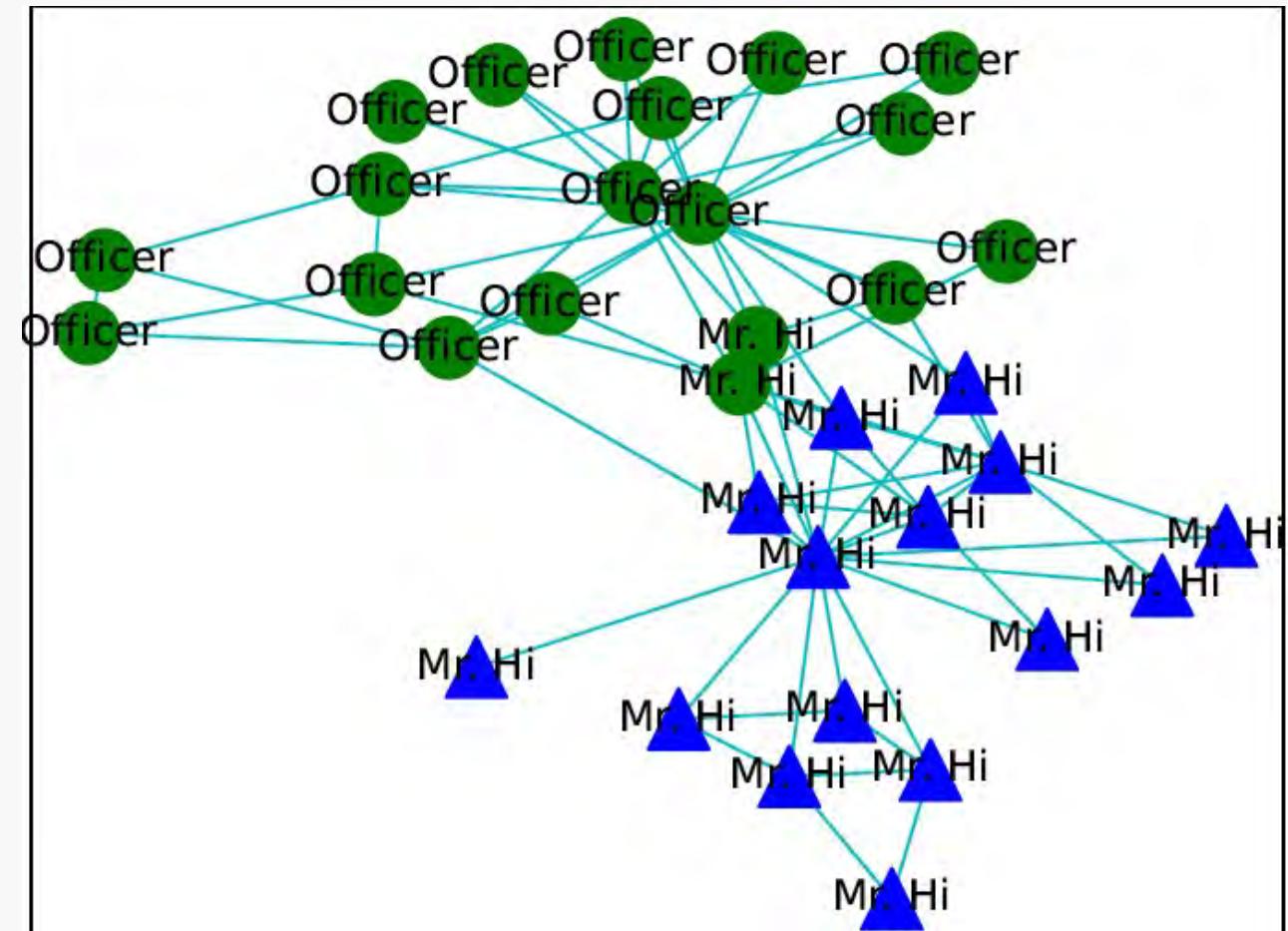
# コミュニティ抽出の実践：Zachary's Karate Club

- ある大学の空手クラブのネットワーク
- 2人の中心人物の対立によって  
2つの派閥が存在
- コミュニティ抽出のベンチマーク  
に広く用いられる



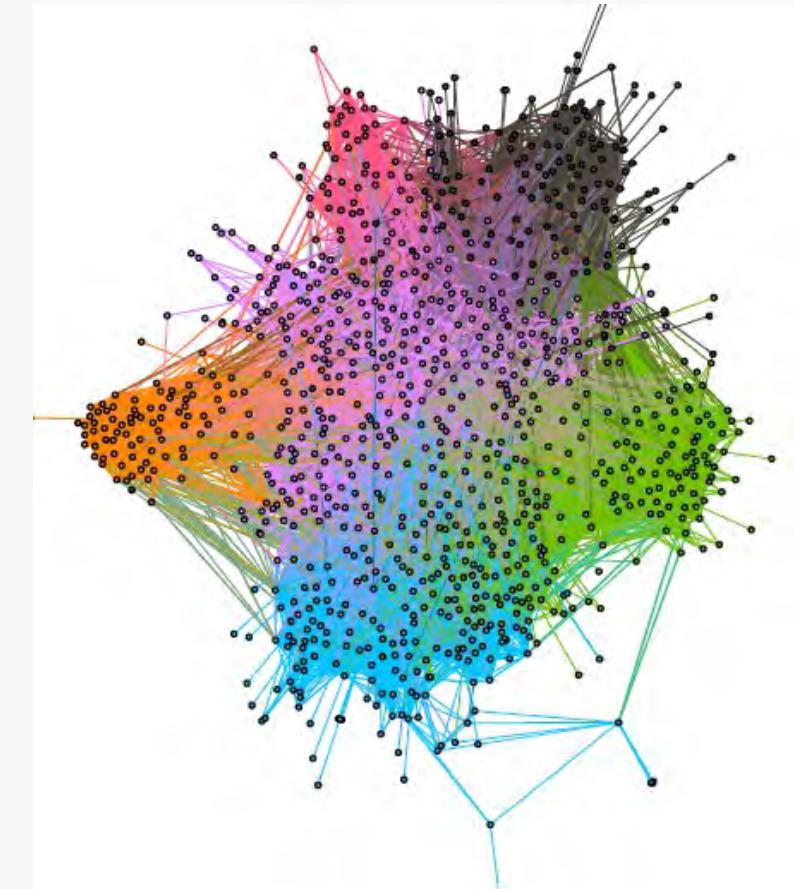
# Karate Club ネットワークのコミュニティ抽出結果

- GN法を適用
- ネットワーク構造から派閥を特定可能



# ネットワーク分析の実践: Email ネットワーク

- Newman 法の拡張 (CNM 法) で  
コミュニティを抽出
- これの良さをどのように評価するか?



# 真のコミュニティが分かっている時の コミュニティ抽出結果の評価

- Rand Index

$$RI = \frac{TP + TN}{TP + FP + TN + FN}$$

	同じコミュニティ	異なるコミュニティ
同じラベル	TP	FP
異なるラベル	FN	TN

- さきほどの電子メールの例では 0.77
- 実際には、チャンスレベルのコミュニティ分割の RI で補正した Adjusted Rand Index がよく用いられる

# コミュニティ抽出のプログラム

チュートリアル資料中にある論文の共著関係ネットワークの中心性と論文の被引用数の関係を求めるには、以下のようにプログラムを実行します。

```
python3 calc_cent_citation.py aps/apscoauthor.csv aps/citation_count.txt >result.txt
```

ペアプロットは、以下のようにプログラムを実行すると、pairplot.epsという名前でファイルが出力されます。かなり点数が多いので、時間がかかるのに注意。

```
python3 pair_plot.py result.txt
```

## まとめ

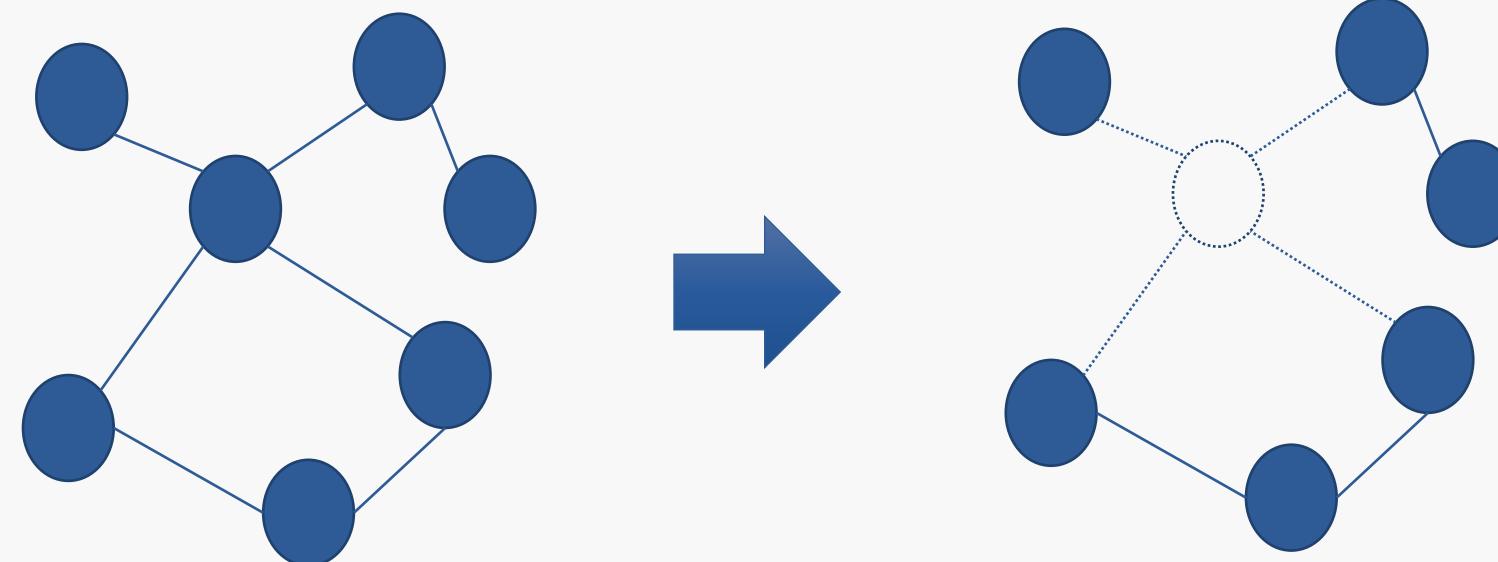
- コミュニティ抽出は中心性と並んで重要な分析手法
- 現在進行形で研究が進んでいる
- 経験的にはモジュラリティ最大化の手法は計算量的にも得られるコミュニティの質的にも有用
- 2つの実ネットワークでコミュニティを抽出した例を紹介

# 第6章： ネットワークのロバスト性

---

## ネットワーク内のノードの欠落と ネットワークの連結性

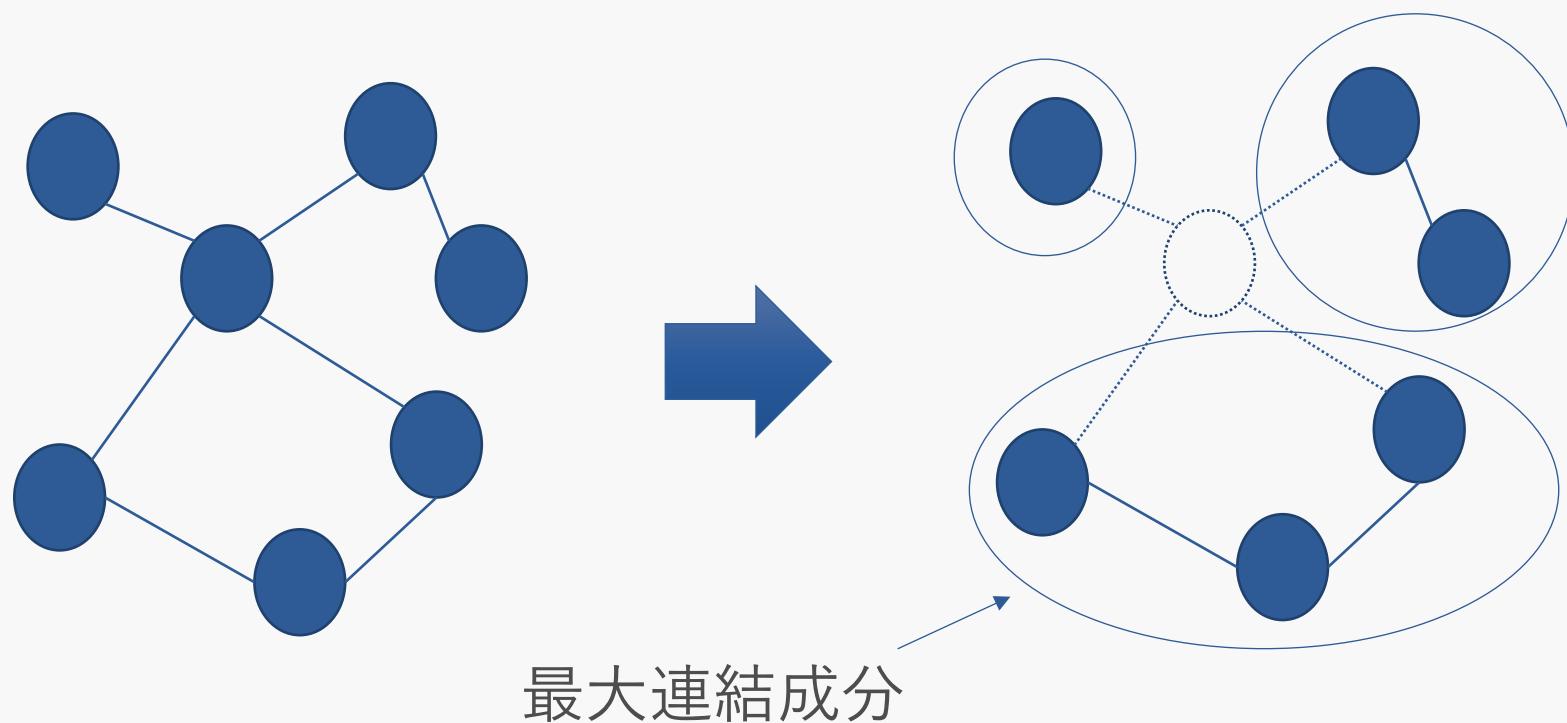
- ネットワークからノードが欠落すると  
連結なネットワークが非連結になる場合がある
- 通信網、交通網、電力網などではノードが欠落 (e.g., 故障、攻撃  
を受ける)しても他のノード間の連結性は維持されてほしい



# インターネットのアキレス腱 [Albert, Jeong, Barabasi, Nature 2000]

- ランダムなノード除去に頑健
- ハブへの攻撃に脆弱

# ロバスト性の評価指標：最大連結成分の大きさ



# バラバシらの研究をなぞってみよう

## Usage

次数順、ランダム、HDA (high degree adaptive) に従ってノードを削除した時の最大連結成分のサイズを求めるには、以下のようにプログラムを実行します。

ランダム

```
python3 random_remove.py network/ba.txt
```

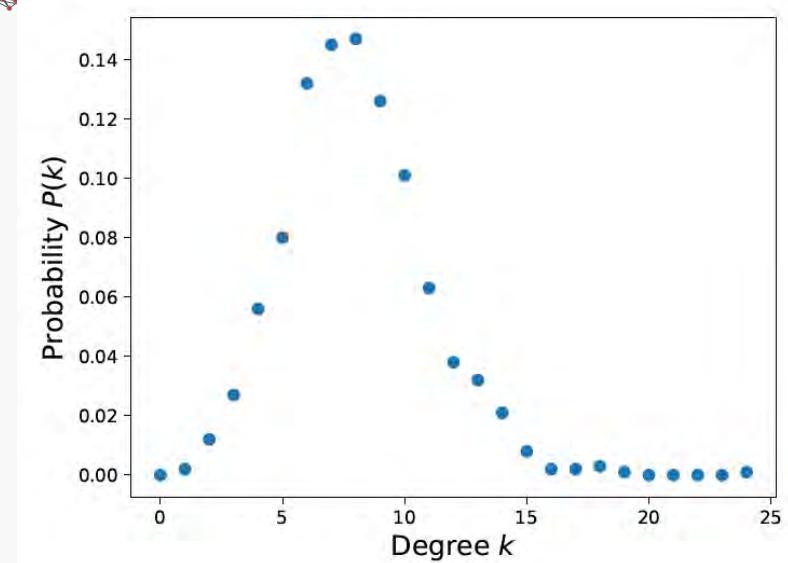
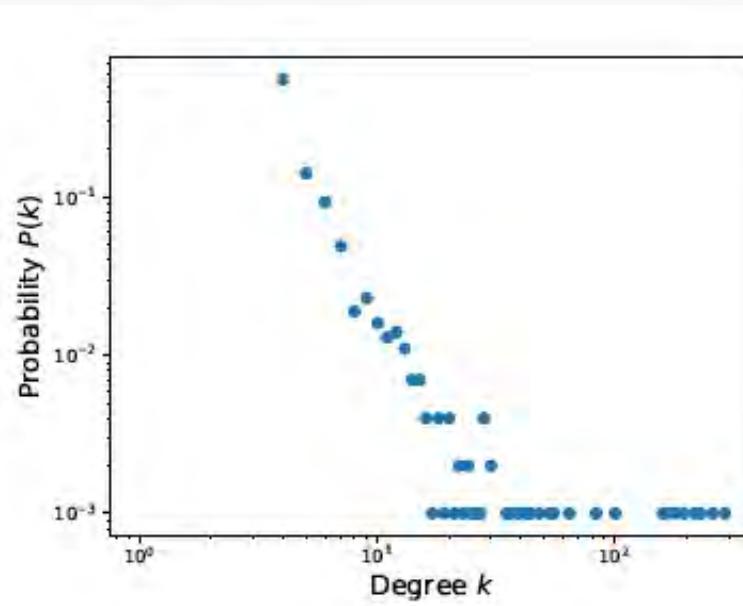
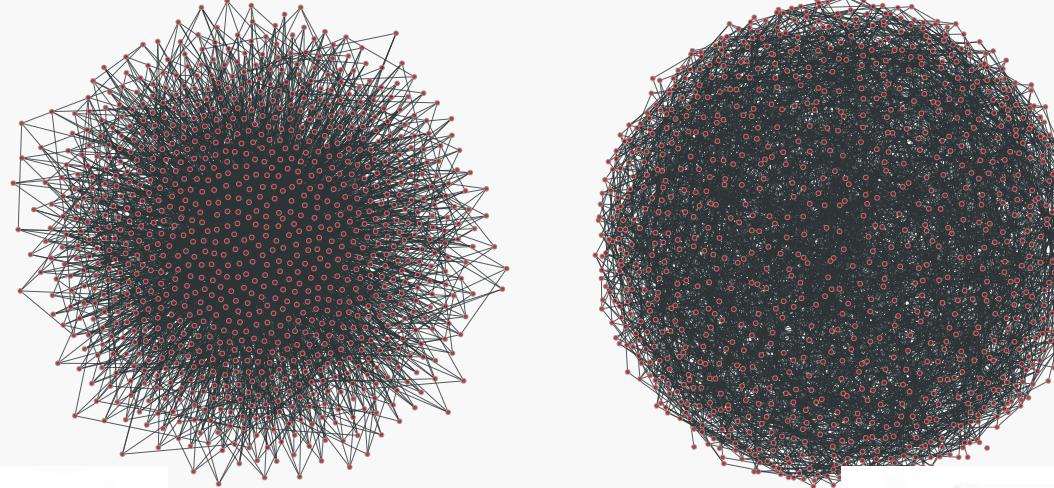
次数順

```
python3 degree_remove.py network/ba.txt
```

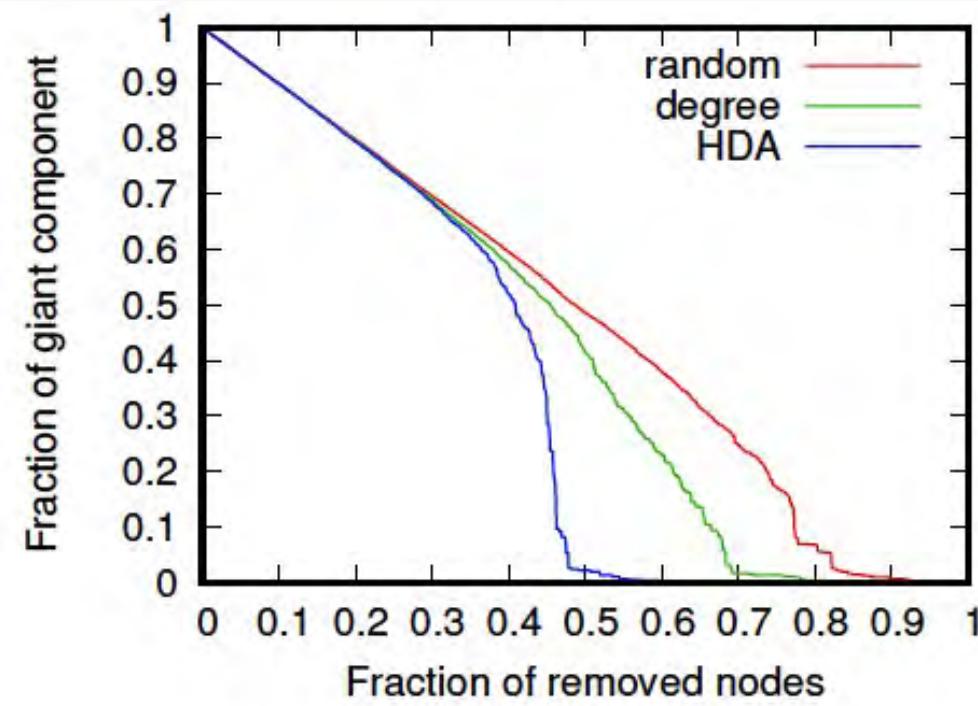
HDA

```
python3 hda_remove.py network/ba.txt
```

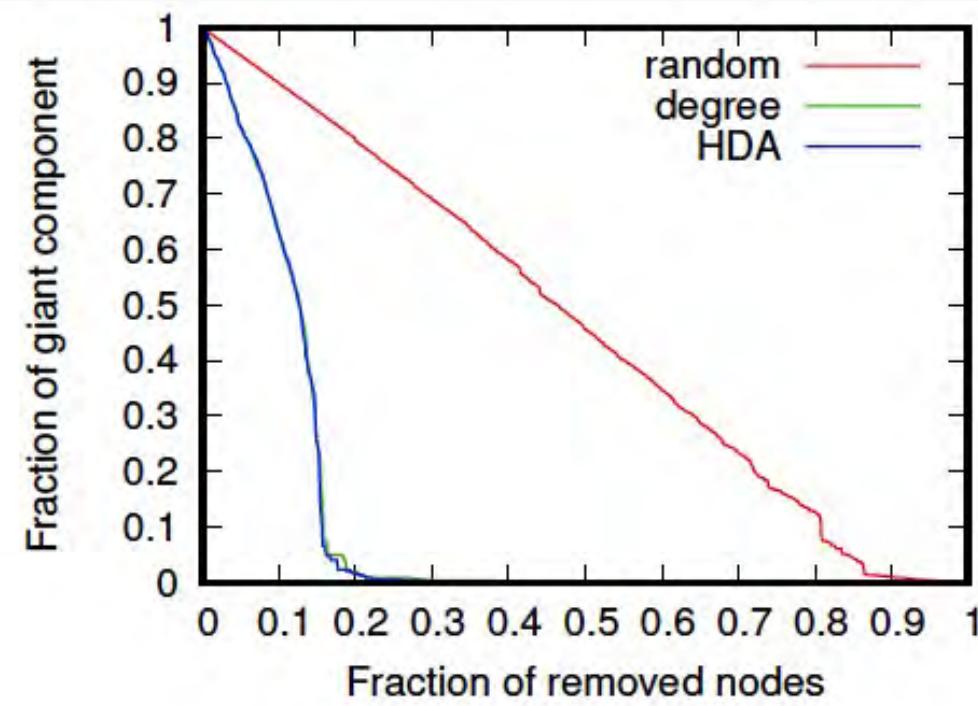
# BA グラフと ER グラフ



# 削除するノードの割合と最大連結成分の割合の関係

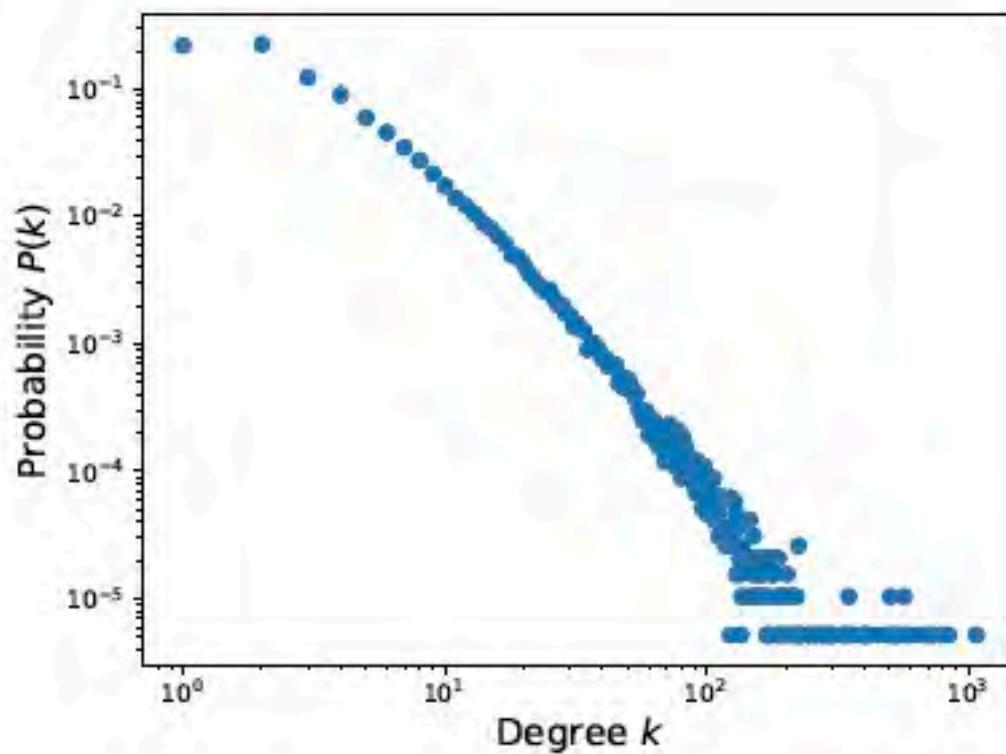


(a) ER グラフ

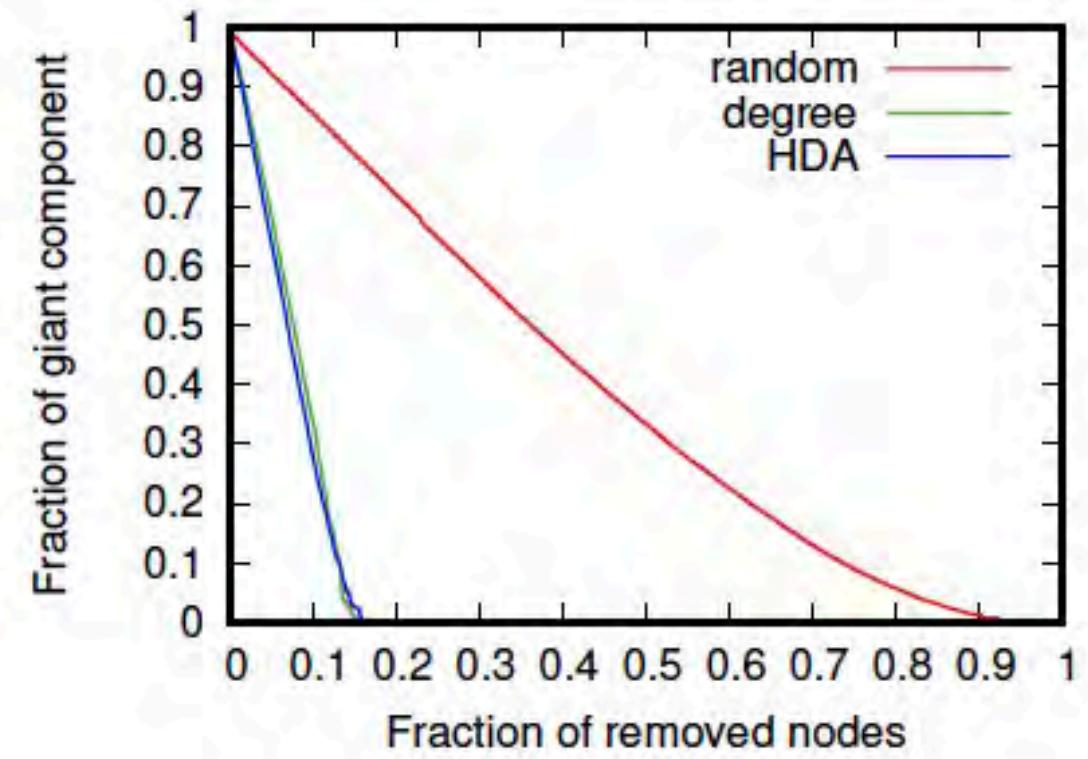


(b) BA グラフ

## ルータレベルトポロジの場合



(a) 次数分布



(b) 最大連結成分

## まとめ

- ネットワークのロバスト性：ネットワークのノードの欠落に対して連結性がどの程度維持されるか
- ノードを削除した時の最大連結成分の大きさを評価
- 次数分布がべき分布のネットワークはハブを狙った攻撃に脆弱
- より発展的なトピック
  - どのようにしてロバストなトポロジを作るか？
  - どのようにして効率的にネットワークを分断するか？

# 第7章： ネットワーク上のウィルス拡散

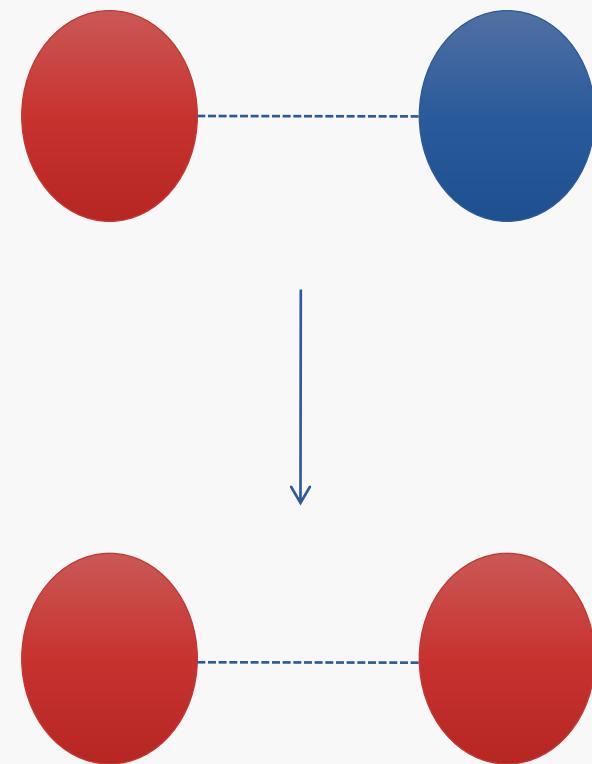
---

---

# 新型コロナウィルス COVID-19

## ウィルス拡散とネットワーク

- 人と人との接触でウィルスが拡散
- 人と人の接触パターン (=ネットワーク) が拡散のパターンに大きく影響



# 現代社会のネットワーク構造とウィルス拡散

- 高い人口密度



- 容易な長距離移動

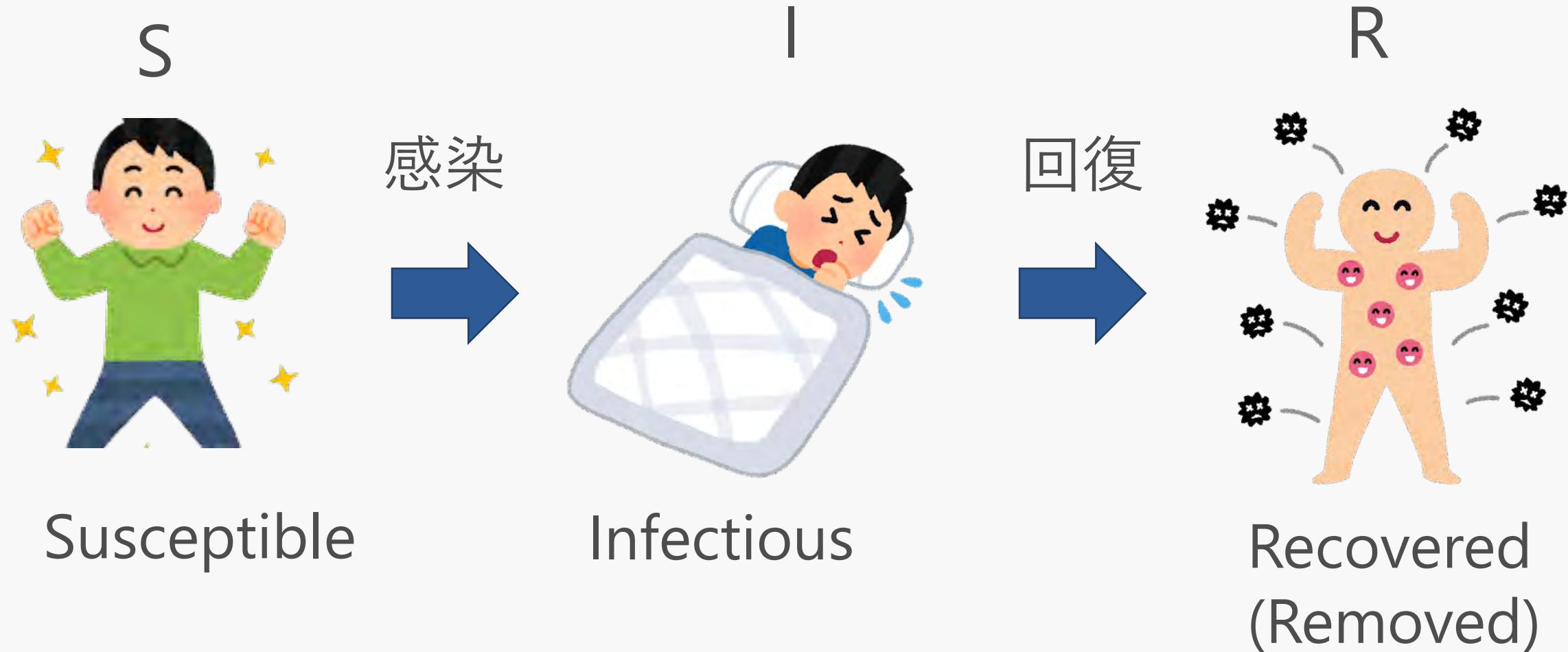


ウィルスにとって都合のいい状況

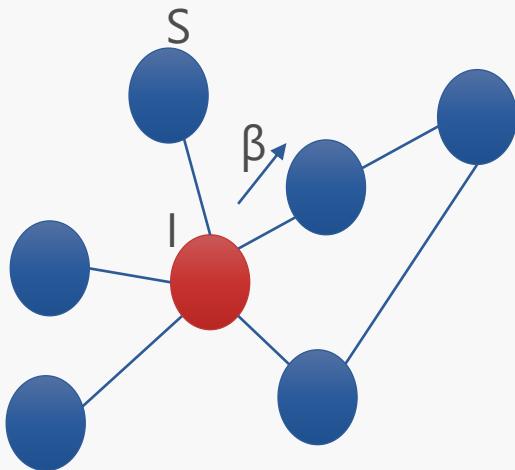
## ウィルス拡散に関する疑問

- 人のネットワークのウィルスの広がりやすさは?
- 接触頻度を削減することの効果は?
- 長距離の移動を制限することの効果は?

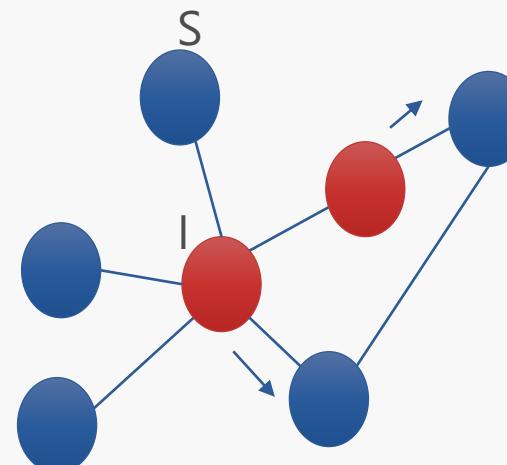
## ウィルス拡散のモデル：SIR モデル



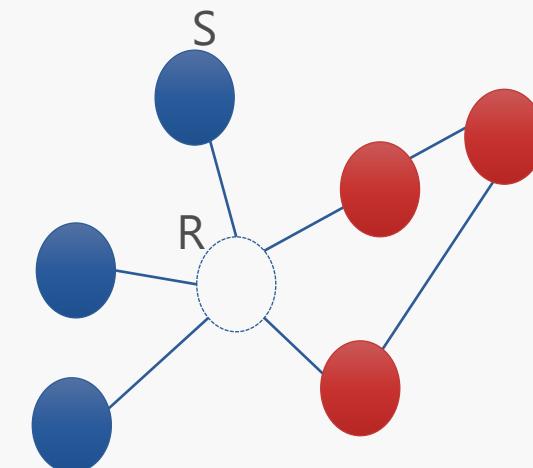
# ネットワーク上のウィルス拡散



状態 I のノードは  
状態 S の隣接ノー  
ドを確率  $\beta$  で感  
染させる



状態 I のノードは隣接  
ノードの状態とは無関  
係に確率  $\gamma$  で状態 R に  
遷移する



# SIRモデルのシミュレーション：概要

- 1000人の社会を考える
- 時刻 0 に1000人の中からランダムに選んだ1人だけを病気に感染させる
- 病気に感染した人は単位時間ごとにその人の友人それぞれに対して確率 4% で病気を感染させる
  - 25 回に1回感染させるくらいの確率
- 病気に感染した人は単位時間ごとに確率20%で病気が治る
  - 平均 5 ステップで治るくらいの確率
- それぞれの人は平均 8 人の友人を持つ
- 各時刻ごとに感染した人の数を求める

# シミュレーションプログラム (sir.py)

- 1回の試行

\$ python3 sir.py network/ba.txt

- 100回試行して平均を求める

\$ sh run-sir.sh network/ba.txt

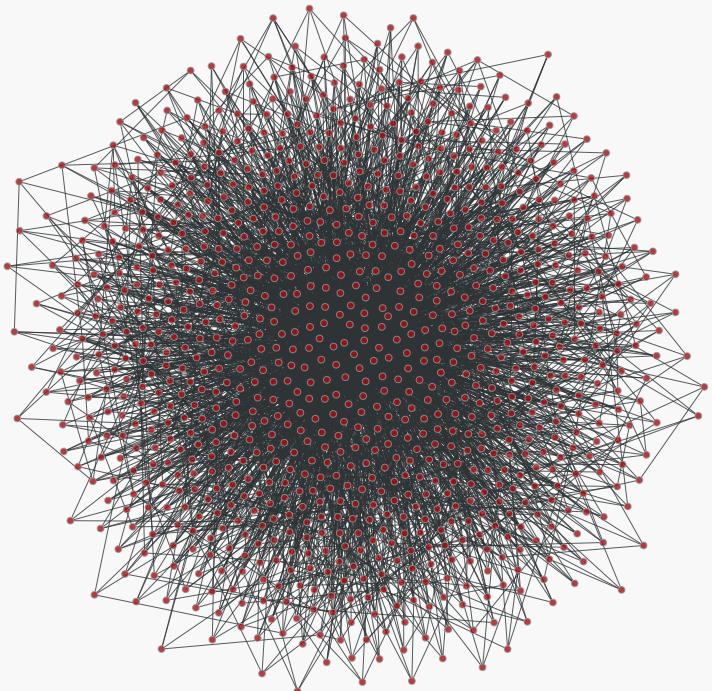
## SIR モデルの状態遷移

```

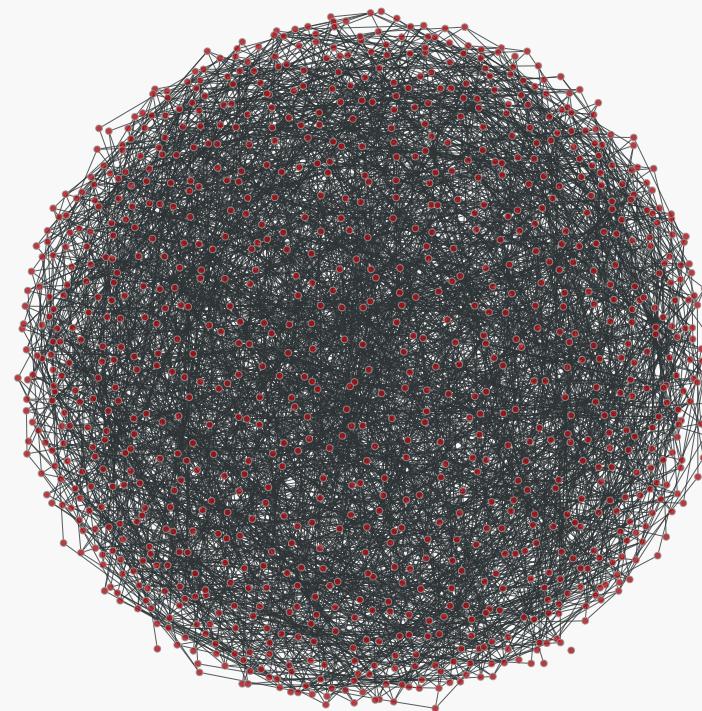
18 max_count = 100
19 num_r=0
20 for time in range(1,max_count):
21     inflist=list(inf.keys())
22     shuffle(inflist)
23     for v in inflist:
24         for nbr in g[v]:
25             if nbr not in inf:
26                 if (random()<beta):
27                     inf[nbr]=1
28                 if(random()<r):
29                     del inf[v]
30                     g.remove_node(v)
31                     num_r+=1
32             num_inf=num_r+len(inf)
33             # 時刻、状態 S の数、I の数、R の数、R+I の数
34             print(time,len(g.nodes()),len(inf),num_r,num_inf)

```

# 次数分布の異なる2つのネットワーク

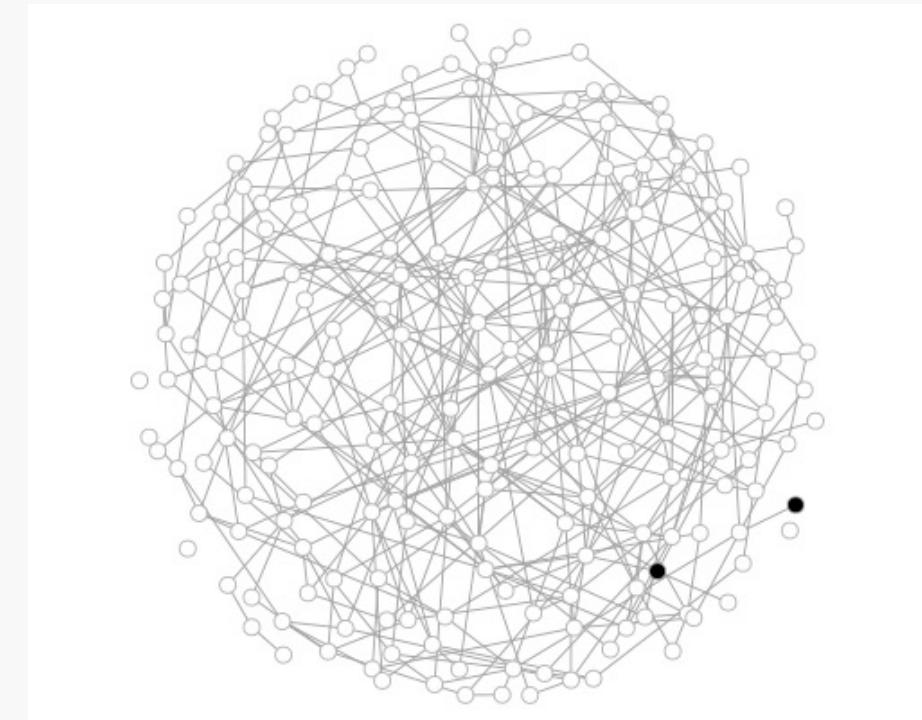
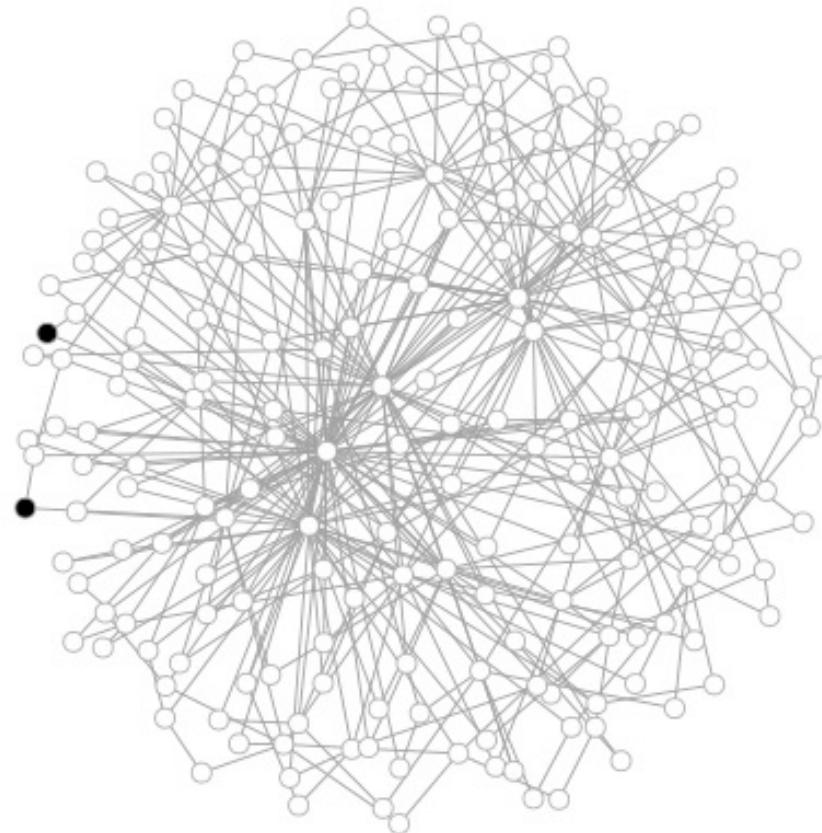


BAグラフ  
次数分布：べき分布



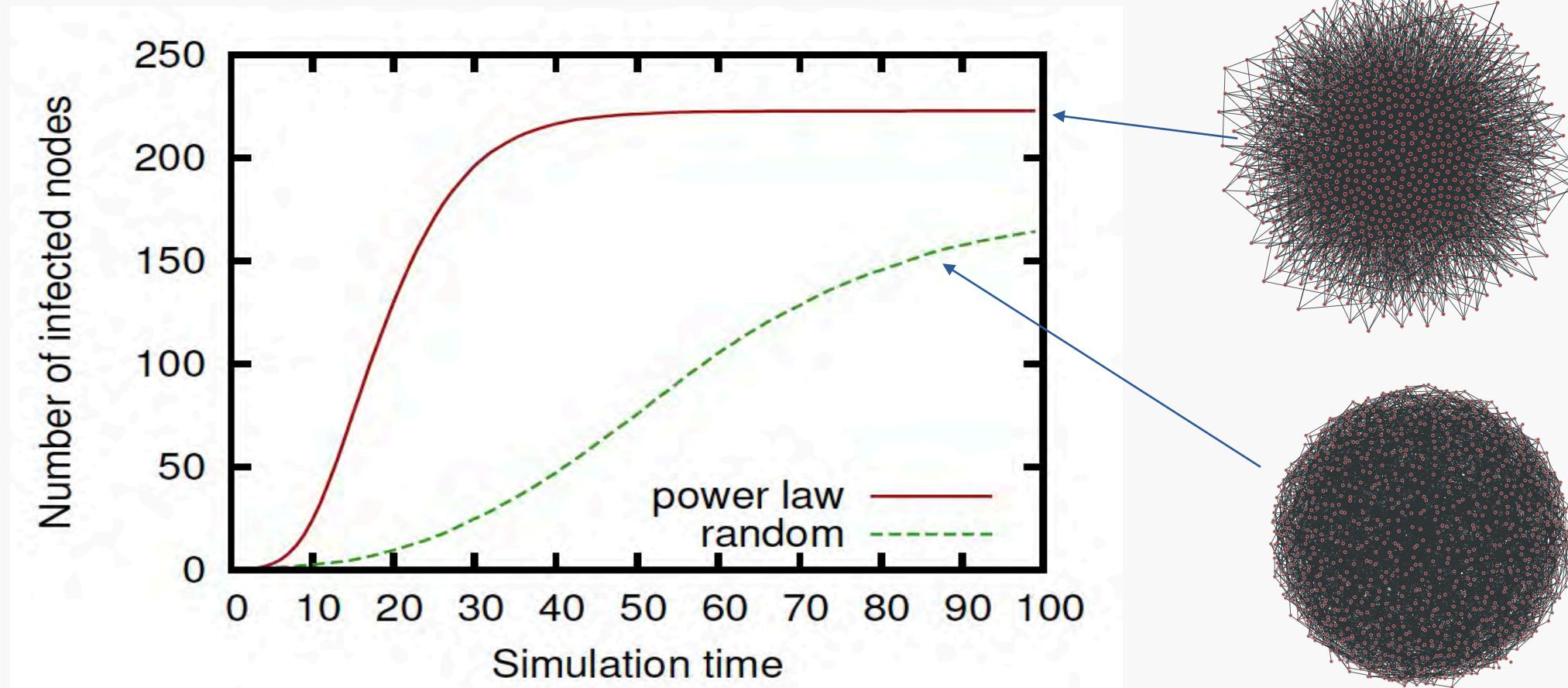
ERグラフ  
次数分布：二項分布

# SIRモデルのシミュレーション：デモ

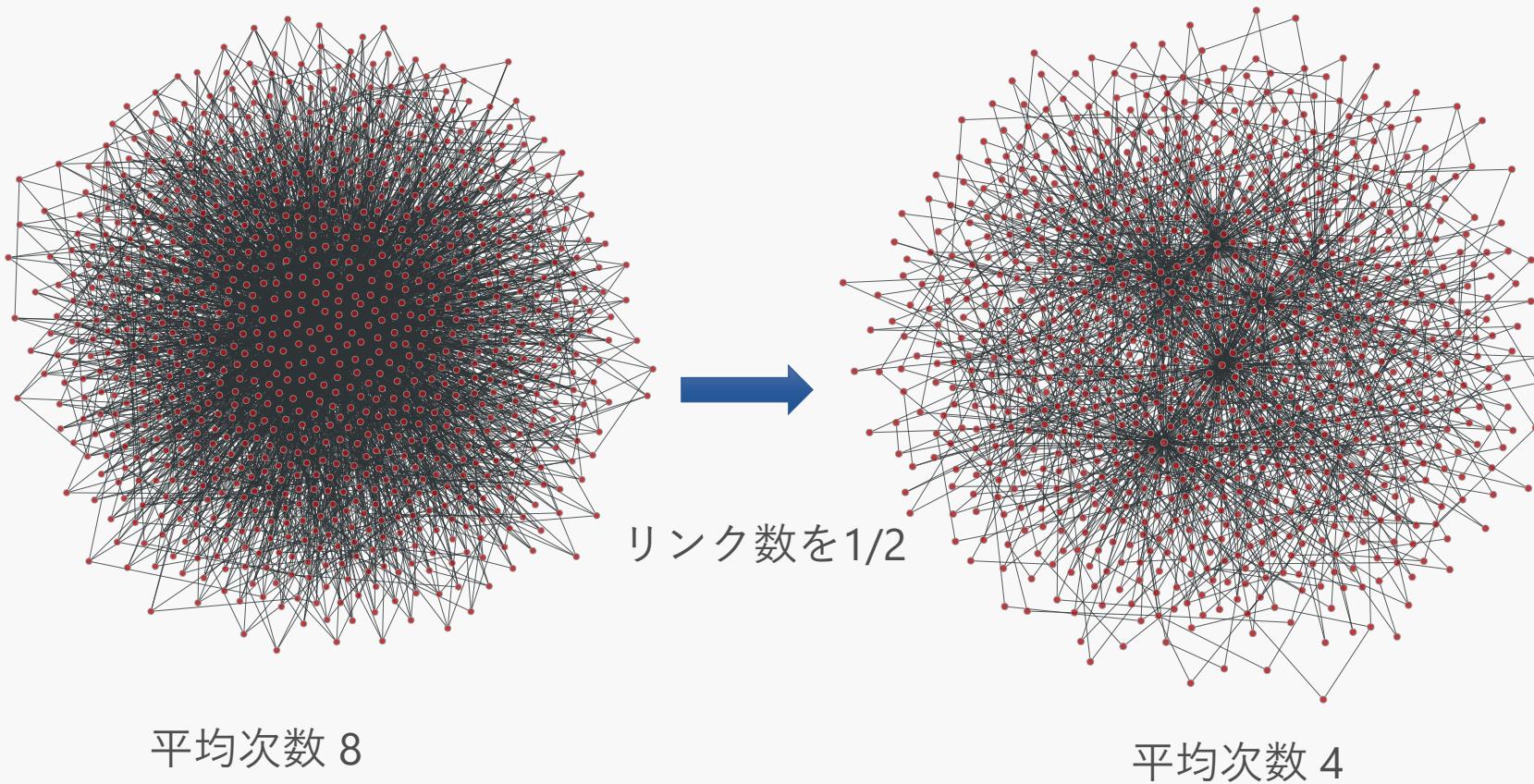


## 次数分布が感染の規模に与える影響

- 次数分布がべき分布のネットワークでウィルスが急速に拡散

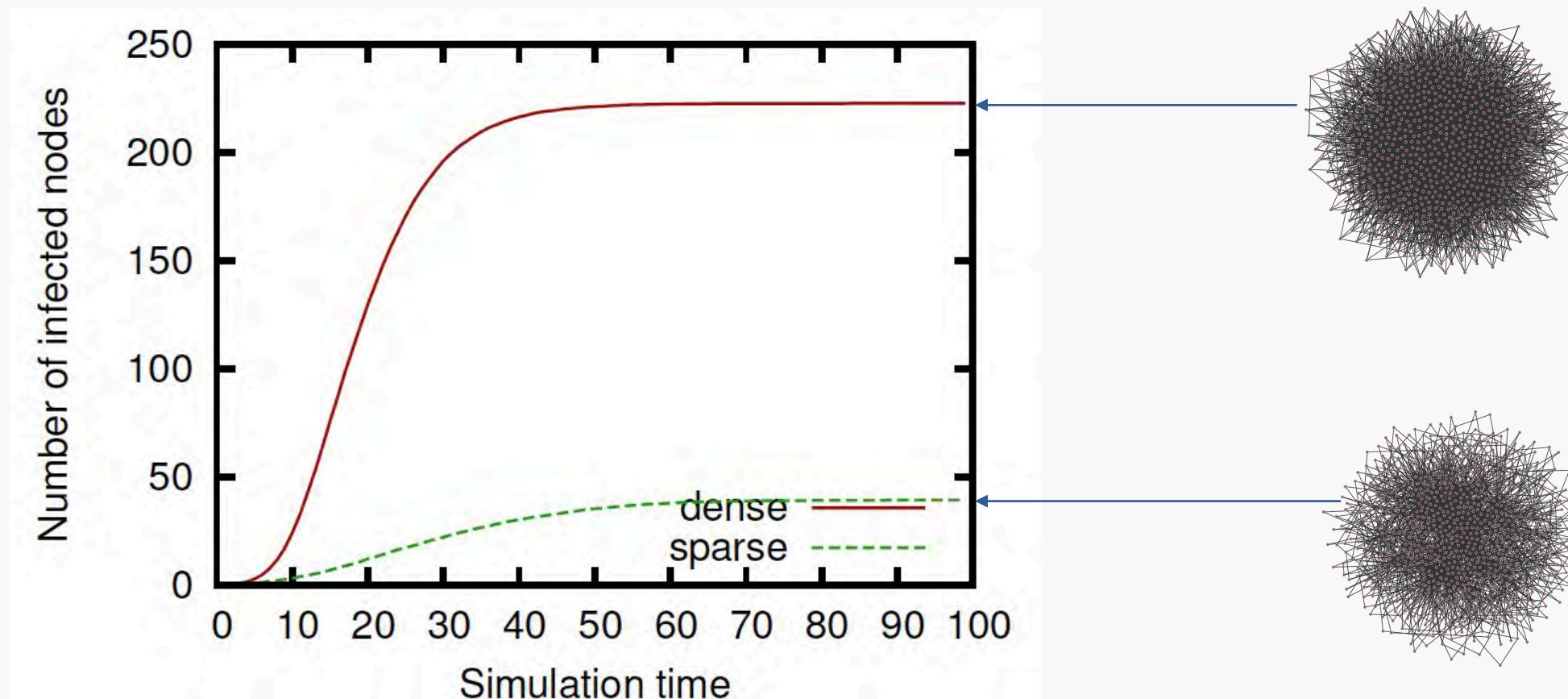


# 接触頻度の削減効果

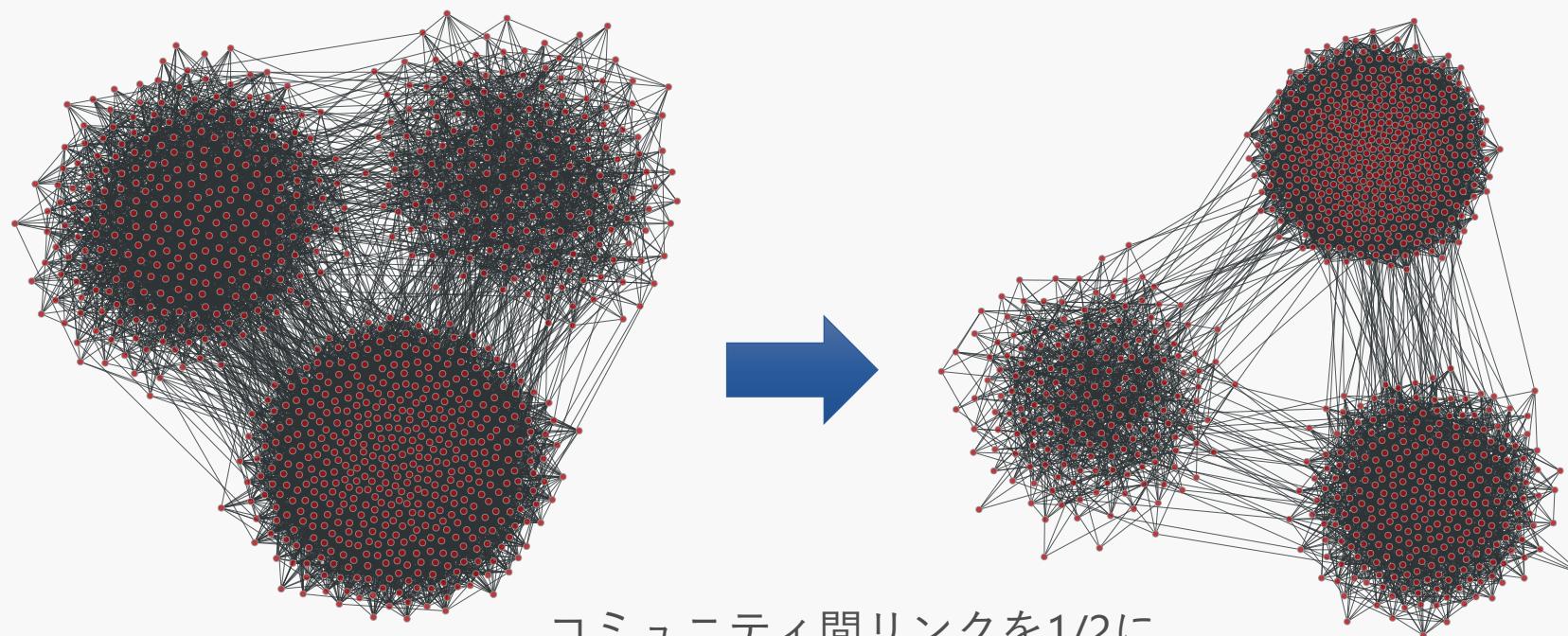


## 平均次数が感染の規模に与える影響

- 平均次数の削減が感染規模に大きく影響



# コミュニティ間接触の削減効果

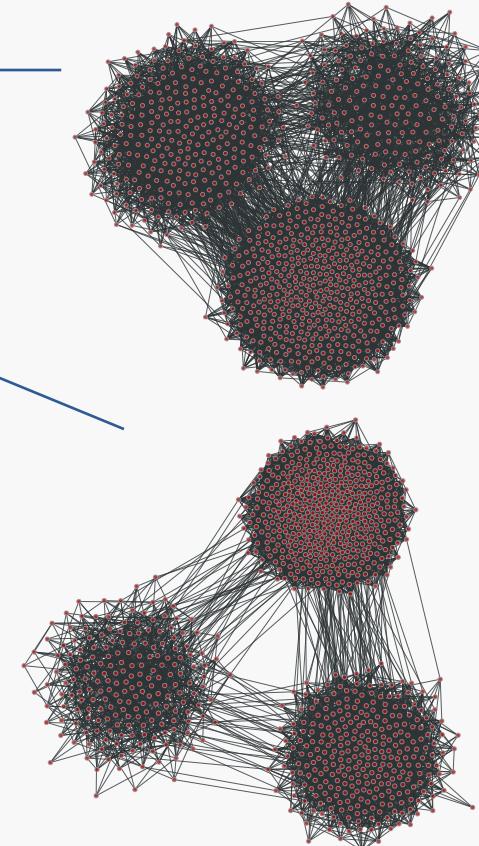
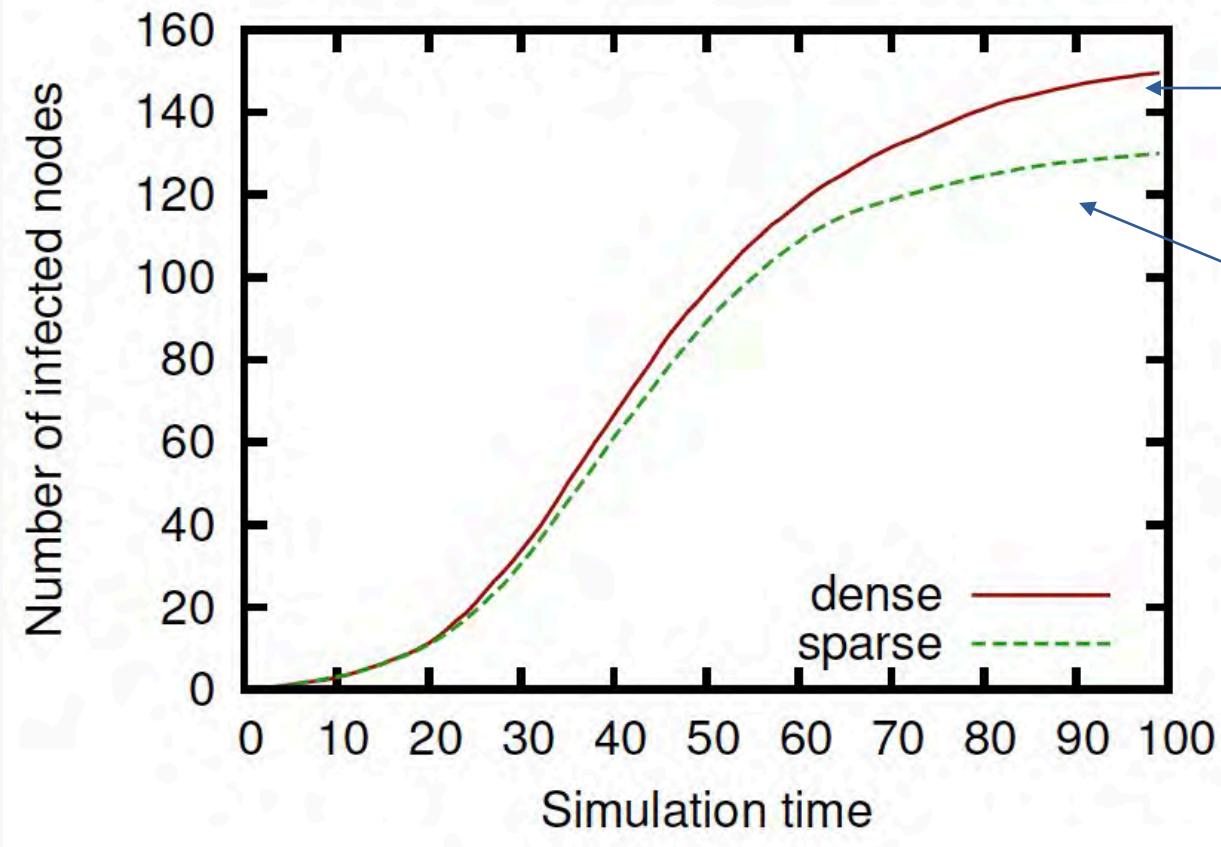


平均次数 20.28

平均次数 19.46

## コミュニティ間リンクの影響

- コミュニティ間のリンクを減らすだけでも感染規模は低下



## まとめ

- ネットワーク上の SIR モデルに基づく  
ウィルス拡散シミュレーションの方法を学習
- 感染症対策評価におけるネットワーク分析の有用性の一端を紹介
  - 人のネットワークはそもそもウィルスが広がりやすい構造である
  - 接触頻度の削減やコミュニティ間の接触の削減効果を定量的に確認
- より発展的なトピック
  - 誰にワクチンを打てば効果的に感染を抑えられるか?

# 第8章： Twitterのエゴネットワーク解析

---

# Twitter

- 一般に広く用いられるソーシャルメディア
- 研究対象としてもよく用いられる
  - API が公開されている
  - 人の関係や行動を大規模に準リアルタイムにそこそこの粒度で観測可能

# 研究対象としてのTwitter

- 実社会を写す鏡としてのTwitter
  - インフルエンザの流行予測
  - 災害状況の可視化
- 観測可能な社会（の一部）としてのTwitter
  - 人のコミュニケーションパターンの大規模な解析
- 新たなメディアとしてのTwitter
  - フェイクニュースの拡散特性の分析
  - 口コミマーケティング手法の開発

# Twitter データの収集

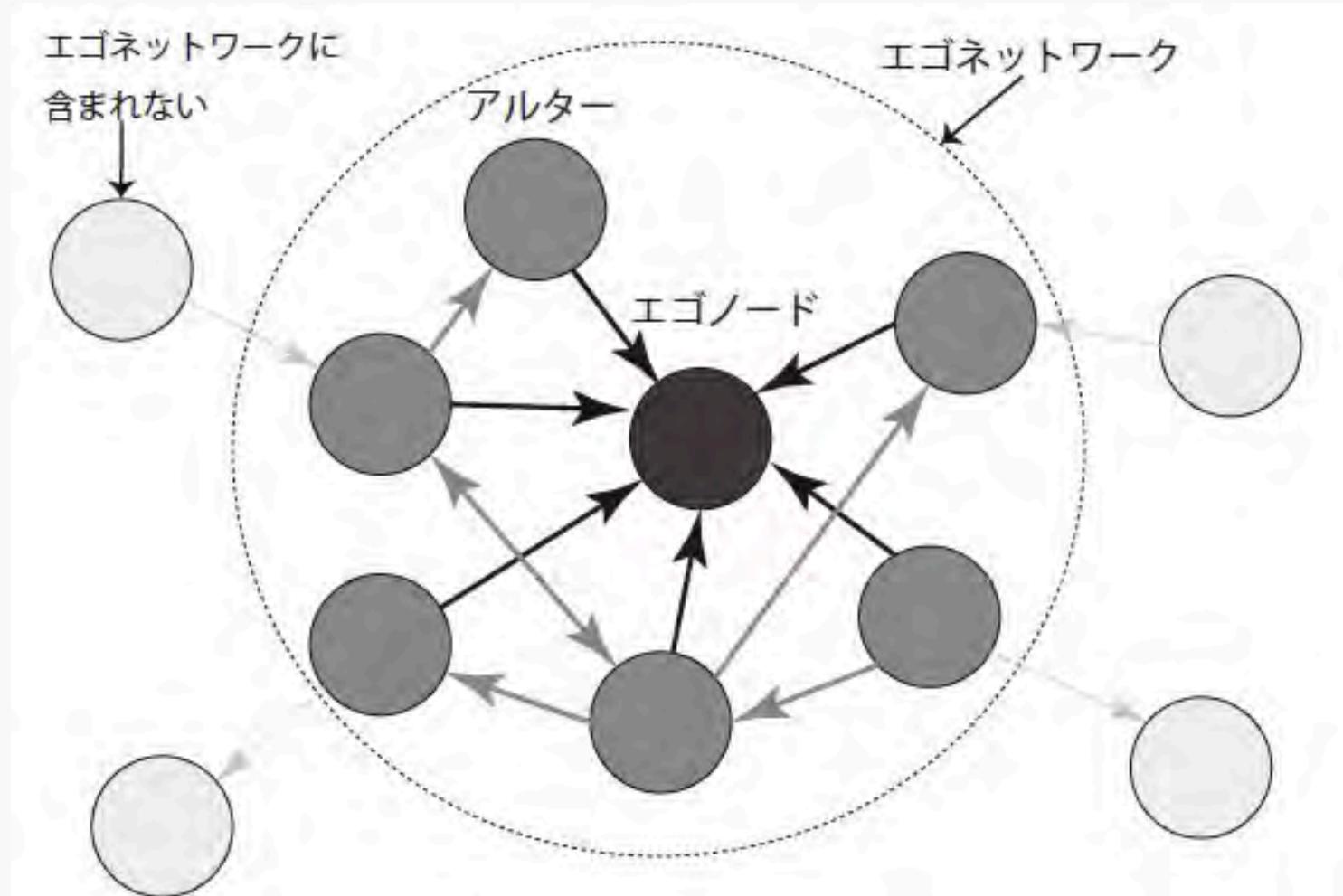
- Twitter API を利用
- ちょっと集めるだけなら思ったより簡単
- 大規模に集めるのは思った  
より難しい
  - その筋の人相談するといい

# Twitter API 利用までの流れ

- (Twitter 自体のアカウント作成)
- Developer 登録  
(<https://developer.twitter.com/en/portal/dashboard>)
- API キー&トークンの利用申請

このあたりのフローはしばしば変更になり、Web 上に情報もたくさんあるので詳細な説明は割愛

## 分析のお題：エゴネットワークの収集と分析



# 分析対象

- MIKA 主催/共催 学会のアカウント

- 電子情報通信学会 (ieice\_EIC)
- 電気学会 (IEEJ\_denki)
- 機械学会 (JSME\_Mech)
- 映像情報メディア学会 (ITE\_or\_jp)

# データの収集

まず、`get_followers.py` および `get_friends.py` の冒頭部分に API にアクセスするための key の情報を入力する。次に作業用ディレクトリに `followers` および `following` という名前のディレクトリを作成する。

```
mkdir followers  
mkdir following
```

`hoge` というユーザ名のフォロワーを取得する際には、以下のようにプログラムを実行する。

```
python3 get_followers.py hoge
```

これで、`followers/hoge.txt` に `hoge` のフォロワーリストが出力される。さらに `hoge` のフォロワーのフォロイーのリストを取得するには以下のようにする。

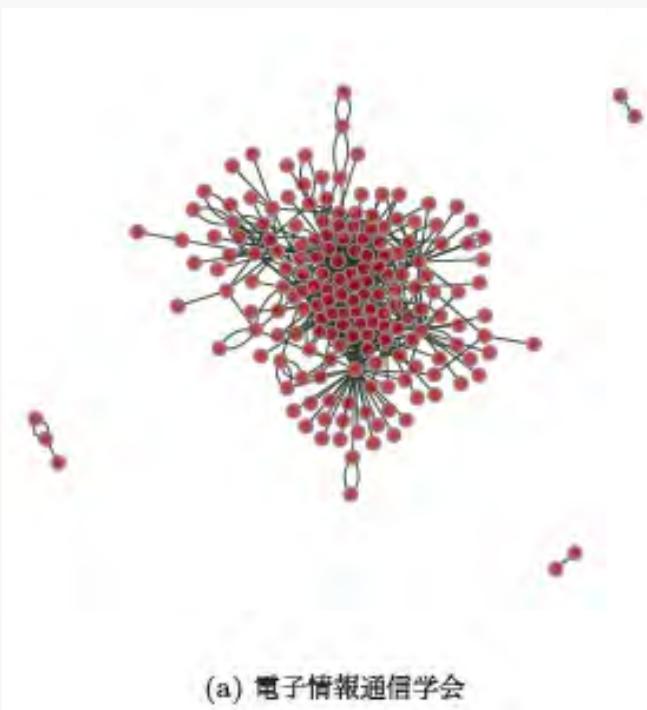
```
sh crawl.sh followers/hoge.txt
```

これで、各フォロワーのフォロイーのリストが `following/XX.txt` に保存される。`hoge` のエゴネットワークを構築するには、以下のようにプログラムを実行する。

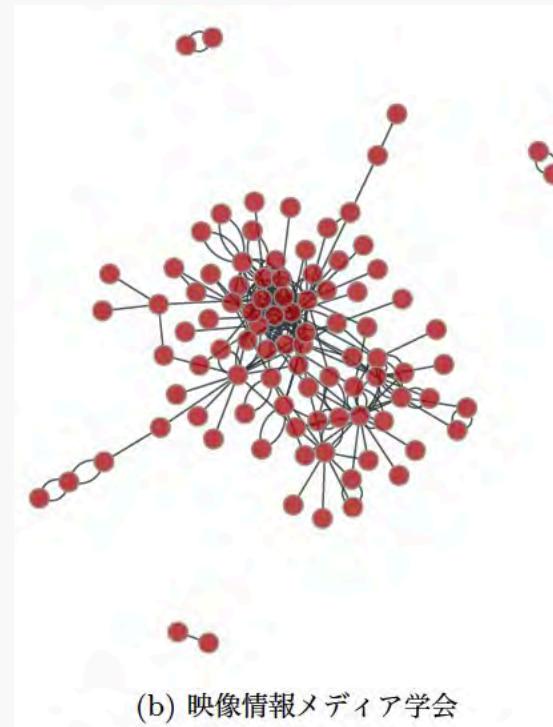
```
python3 make_egonet.py followers/hoge.txt
```

## 4学会のネットワーク

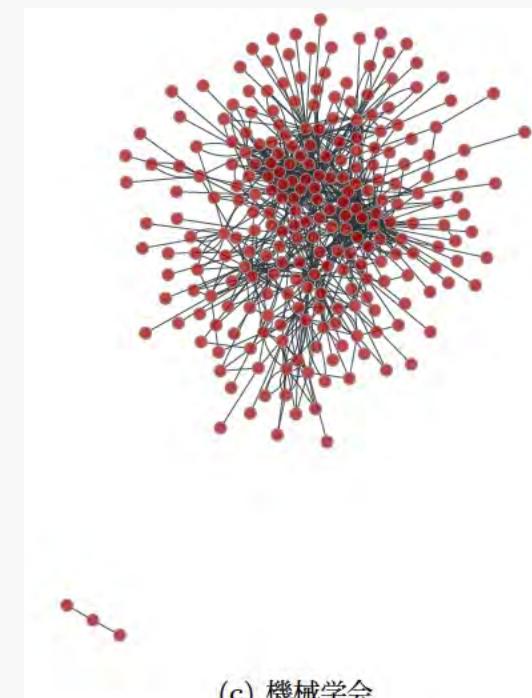
- コアと周辺
- 電気学会：密度高、機械学会：周辺多い？



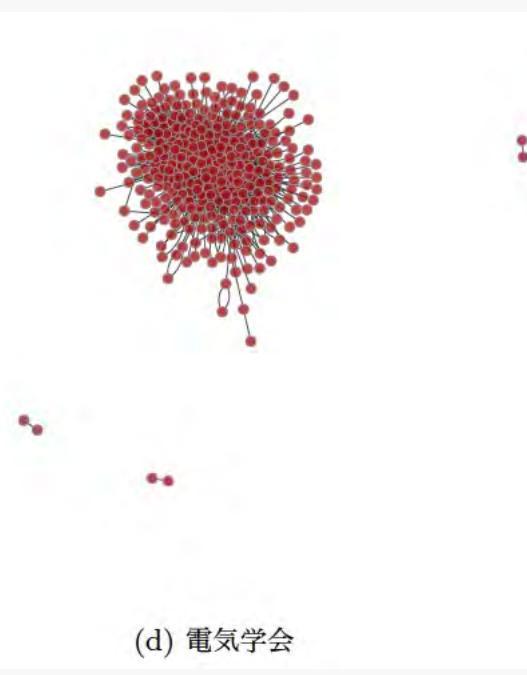
(a) 電子情報通信学会



(b) 映像情報メディア学会



(c) 機械学会



(d) 電気学会

## 4学会のフォロワーをまとめたネットワーク



## 4 学会のネットワークの特徴量

- そこそこ似ている
- 電気：結束が強い？
- 機械：色々な人がフォローしている？
- 4学会をまとめても意外とつながる

表 8.1: MIKA 関連アカウントのネットワークの特徴量

	信学会	映メ	機械学会	電気学会	4 学会マージ
ノード数	182	94	250	353	1069
リンク数	537	258	812	2168	5746
平均次数	4.3	4.2	5.36	9.02	8.45
クラスタリング係数	0.26	0.27	0.20	0.40	0.344
連結成分の数	4	4	2	5	4
最大連結成分の割合	0.96	0.94	0.99	0.97	0.99
最短経路長	3.12	3.26	3.32	2.8	2.8
直径	7	9	8	7	7
コミュニティ数	8	5	7	10	12
モジュラリティ	0.48	0.45	0.51	0.43	0.46

## まとめ

- Twitter は色々な研究に使える情報源
- データ収集にはコツが必要だが、ちょっとやってみるハードルはそんなに高くない
- MIKA 関連4学会のエゴネットワーク構造は全体的に似ている
- 4学会をフォローしている人同士は意外とつながっている

# 第9章： 研究者ネットワークの解析

---

# MIKA (Multiple Innovative Kenkyu-kai Association for Wireless Communication)

## 設立の背景

スマートフォン等の無線デバイスの高速化に加え、 IoT (Internet of Things) を代表とする新たな無線デバイスの普及が始まっている。さらに、多種多様なサービスやアプリケーションも誕生しており、このような多様化する無線トラヒックを安定的に収容するためには、無線アクセスからネットワークまでを総合的に捉えた無線システムの創出が必要とされる。

このような背景において、通信ソサイエティ(以降、通ソとする)における多分野の研究者間で、分野横断的な議論・連携を深めるために研究会共催やワークショップなどが精力的に開催されるようになってきた。

そこで、これらの活動を更に促進するために、2018年度には大規模開催の一例として、通ソに属する 10 研究会に加え、他ソサイエティの基礎・境界ソサイエティ(SITA)や NOLTAソサイエティ(CCS)と連携した「Multiple Innovative Kenkyu kai Association for wireless communications 2018 (MIKA2018)」が横断型研究会として初めて開催されるに至った。さらに 2019年度には、MIKA の継続開催として、通ソに属する 10 研究会と、基礎・境界ソサイエティ(SITA、 RCC、 WBS)、 NOLTA ソサイエティ(CCS)の計 14 の研究会協賛によるMIKA2019 が開催された、MIKA2018 および MIKA2019 は、共に 200 名以上の参加者があり、盛会を納めることができた。

これらのイベントの成功を受けて、分野横断型研究をさらに活性化させ、協賛するすべての研究会が平等な立場で運営する体制を構築するため、第三種研究専門委員会として本委員会を設立する。



## 信学会（通ソ）における分野横断の実際は？

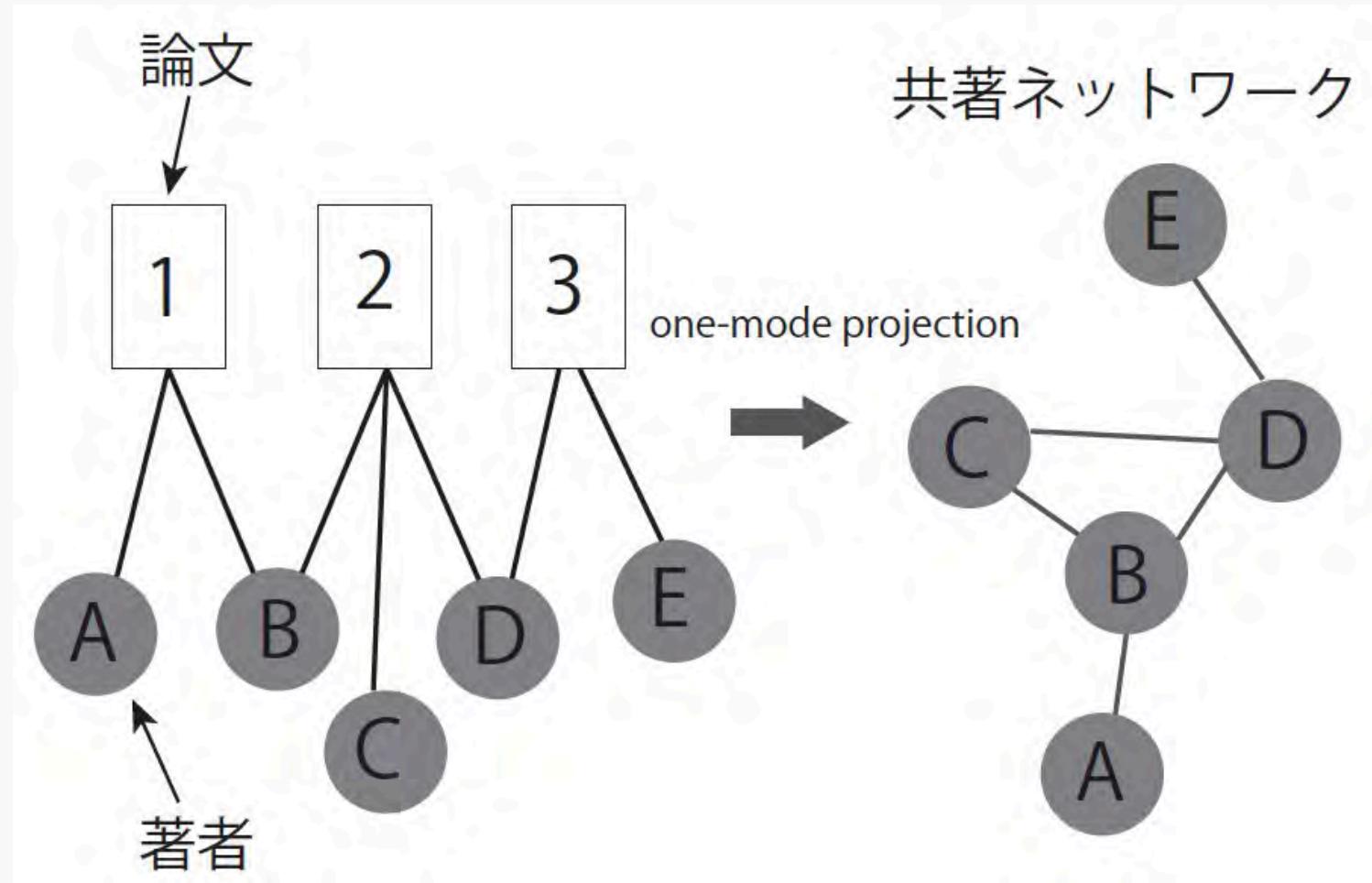
- これからは異分野融合！
  - 学際研究！
  - 横断型研究会！
- … と言ってはいますが、実際のところはどうなんでしょうか？

## MIKA関係研究者の共同研究ネットワークを解析

- 学会活動の中心=研究会
- MIKA 関連研究会における共著ネットワークを解析
- 実際分野横断はどのくらい進んでいるのか?
- 分野横断のキーパーソンは?

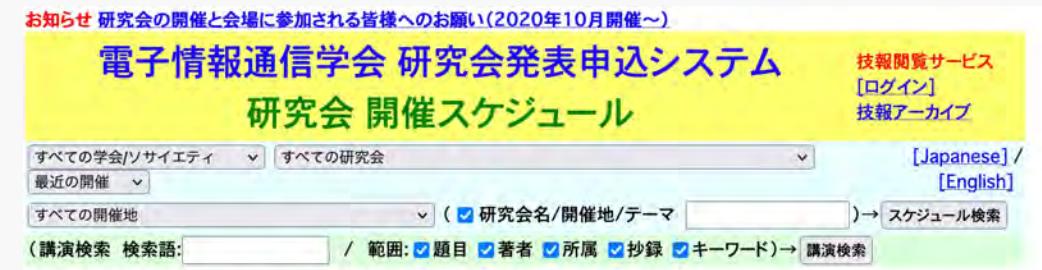
などの疑問に答える

## 共著ネットワーク



## データの収集

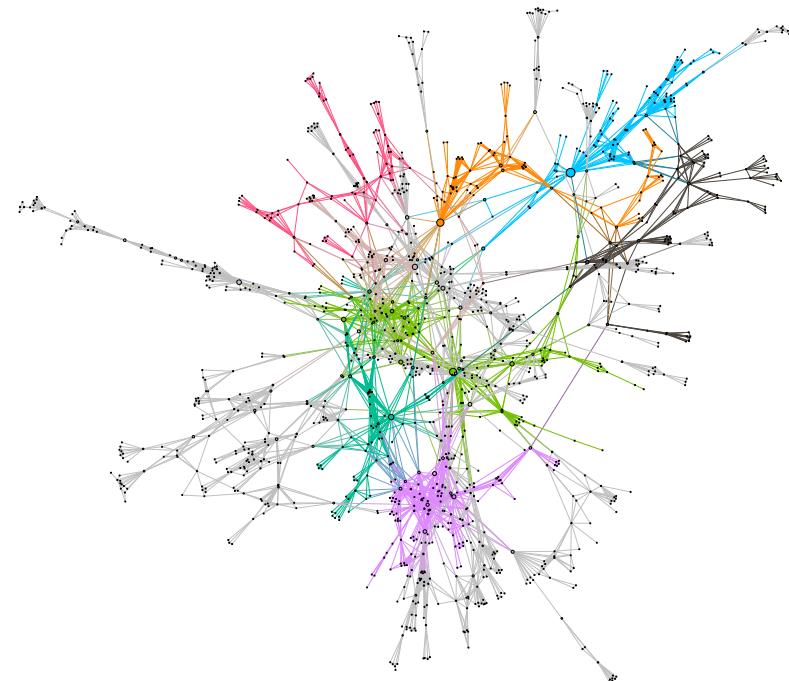
- 対象研究会 : A・P, SANE, CQ, SR, SRW, CS, NS, MICT, RCS, WPT, RCC, WBS, CCS
- 年度 : 2019-2020
- 研究会システムをスクレイピング  
(プログラムはGitに)
- 発表回数1回だけの著者は除く



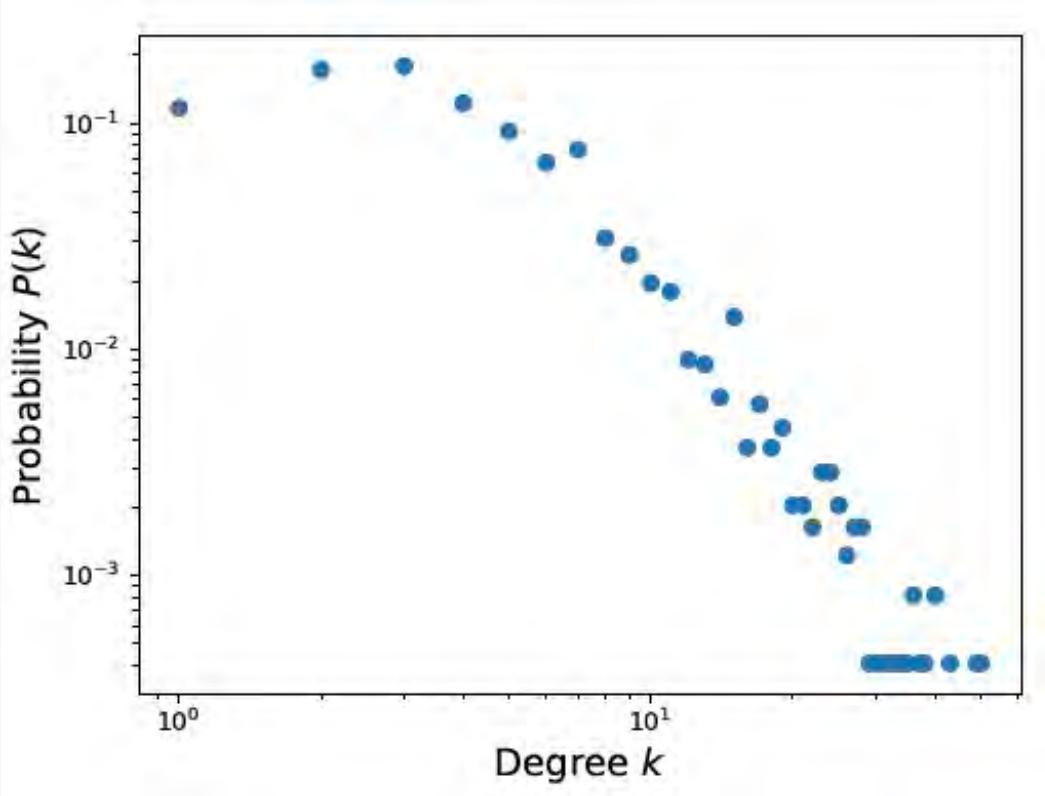
The screenshot shows the homepage of the 'Electronic Information Communication Society Conference Submission System'. It features a search bar for 'Conference Name/Location/Theme' and a search button. Below the search bar, there's a link to 'All Conference Schedule (From Today)'.

開催日	開催地	テーマ	研究会	発表申込締切	選択してください
2021年10月21日(木) - 10月22日(金) (予定)	オンライン開催	眼球運動(調節,瞳孔を含む), 空間知覚(奥行き知覚・運動知覚等), およびヒューマン情報処理一般 新型コロナウイルスの影響を考慮し,本研究会はオンラインで開催します.	HIP	[8月18日(水)]	<ul style="list-style-type: none"><li>詳細はこちら</li><li>締切済</li><li>開催プログラム</li><li>参加費について</li></ul> <small>技術完全電子化研究会</small>
2021年10月21日(木) - 10月22日(金)	オンライン開催	通信システム,一般,【IEEE通信ソサイエティ関西チャプター協賛】	ICTSSL, IEE-SMF, IN (連催)	[8月20日(金)]	<ul style="list-style-type: none"><li>締切済</li><li>開催プログラム</li><li>参加費について</li></ul> <small>技術完全電子化研究会 (ICTSSL/INのみ)</small>
2021年10月21日(木) - 10月22日(金) (予定)	オンライン開催	ロボティクスへの応用を見据えたネットワークソフトウェア技術,一般 新型コロナウイルス感染拡大防止のため,現地開催中止,オンライン開催のみに変更いたします.	NWS (第二種研究会)	[10月1日(金)]	<ul style="list-style-type: none"><li>詳細はこちら</li><li>申込受付中</li></ul>

# ネットワークの可視化

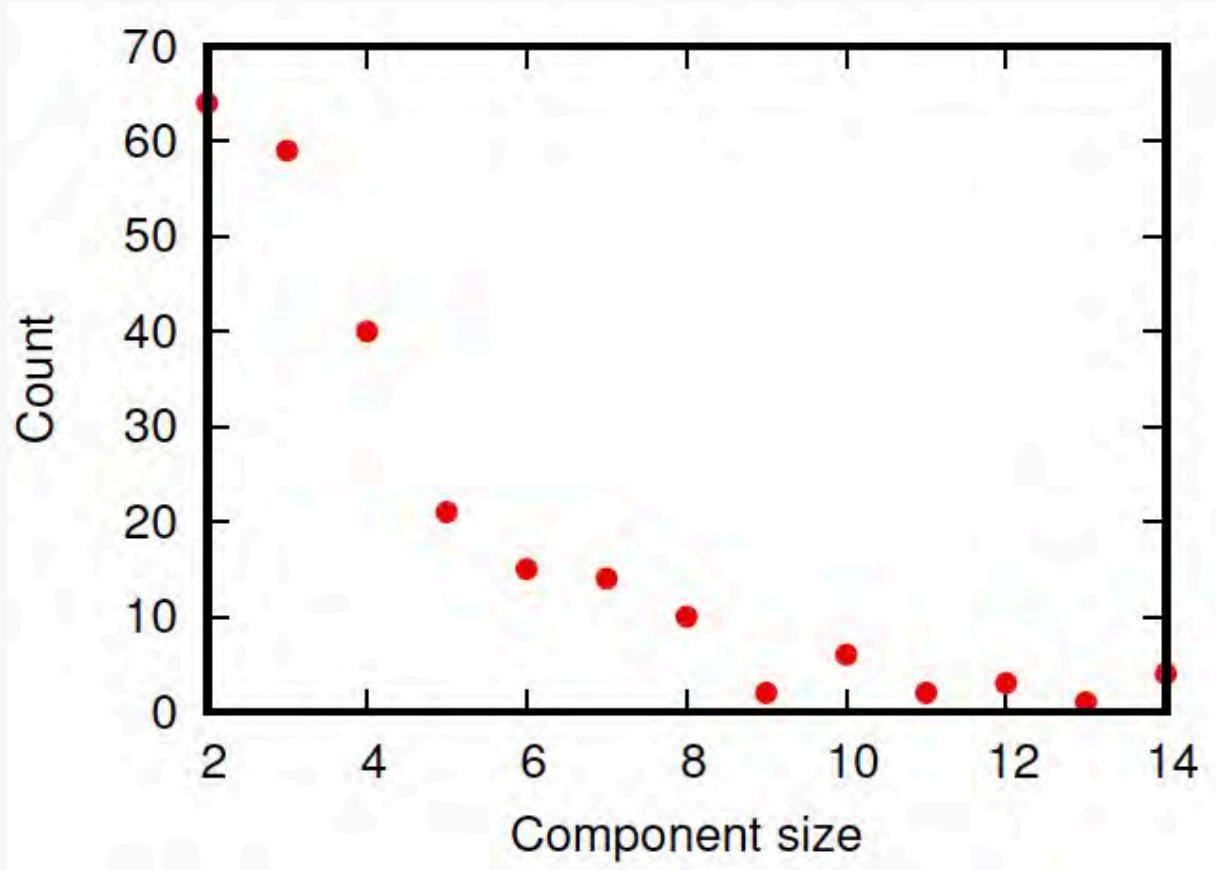


## ネットワークの特徴量



特徴量	値
ノード数	2446
リンク数	6426
平均次数	5.25
クラスタリング係数	0.73
連結成分の数	242
最大連結成分の割合	0.57
最短経路長	6.34
直径	17
コミュニティ数	31
モジュラリティ	0.82

## 連結成分のサイズの分布



## 考察

- 研専や研究分野を単位とすると思われる密なコミュニティの存在
  - 少数のハブやブリッジによって全体としてはゆるく接続
  - 直接関係ない分野同士も意外と近い
- 
- 研究会内での強い紐帯と分野横断の弱い紐帯を有する良い構造?  
c.f., 弱い紐帯の強さ: strength of weak ties

# 分野横断のキーパーソン：困ったらこの人に聞け！

表 9.2: MIKA 関連研専のネットワークにおける中心性の高い研究者

順位	次数中心性	媒介中心性
1	児島史秀 (NICT)	中尾彰宏 (東大)
2	藤井威生 (電通大)	岡本英二 (名工大)
3	田久 修 (信州大)	西森健太郎 (新潟大)
4	西森健太郎 (新潟大)	児島史秀 (NICT)
5	中尾彰宏 (東大)	岸山祥久 (NTT ドコモ)
6	須山 聰 (NTT ドコモ)	米田尚史 (三菱電機)
7	岡本英二 (名工大)	松村善洋 (JR 東海)
8	高田潤一 (東工大)	平栗健史 (日本工大)
9	岸 洋司 (KDDI 総合研究所)	今井哲朗 (長崎大)
10	米田尚史 (三菱電機)	藤井威生 (電通大)

## まとめ

- ネットワーク分析は研究分野の現状把握（予測?）にも有用
- MIKA 関連のネットワークはけっこう良い構造?
  - 各研専での強い紐帶の活性化
  - MIKA（横断型研究会）での弱い紐帶の維持  
でより良い構造に?

## 全体のまとめ

- ネットワーク分析：複雑な対象をネットワークで大胆に抽象化
- ネットワーク分析のコア技術：中心性、コミュニティ抽出
- ネットワーク分析の実践例：  
ネットワークのロバスト性解析  
ウィルスの拡散特性の分析  
Twitter ネットワークの解析  
研究者ネットワークの解析

## 全体のまとめ

- ネットワークとしての表現が適した対象は様々な場面に登場  
→ ネットワーク分析がみなさんの分野でも使える可能性?  
本資料や GitHub (<https://github.com/s-tugawa/MIKA21Tutorial>)  
を参考にぜひトライしていただきたい
- とはいえ、「明日から使えそう!」とはならないのが普通…  
→ 「明日から使えうこと」は既に誰かがやっているはず。  
「お隣りの分野の人はこういうのを面白がってるらしい」くらいのことを  
記憶に留めていただき、何かのきっかけで思い出していただければ