

みんなのPython勉強会

2/7, 2018

いまNetwork embeddingが熱い！

辻 真吾 (@tsjshg)

shingo.tsuji@gmail.com

自己紹介

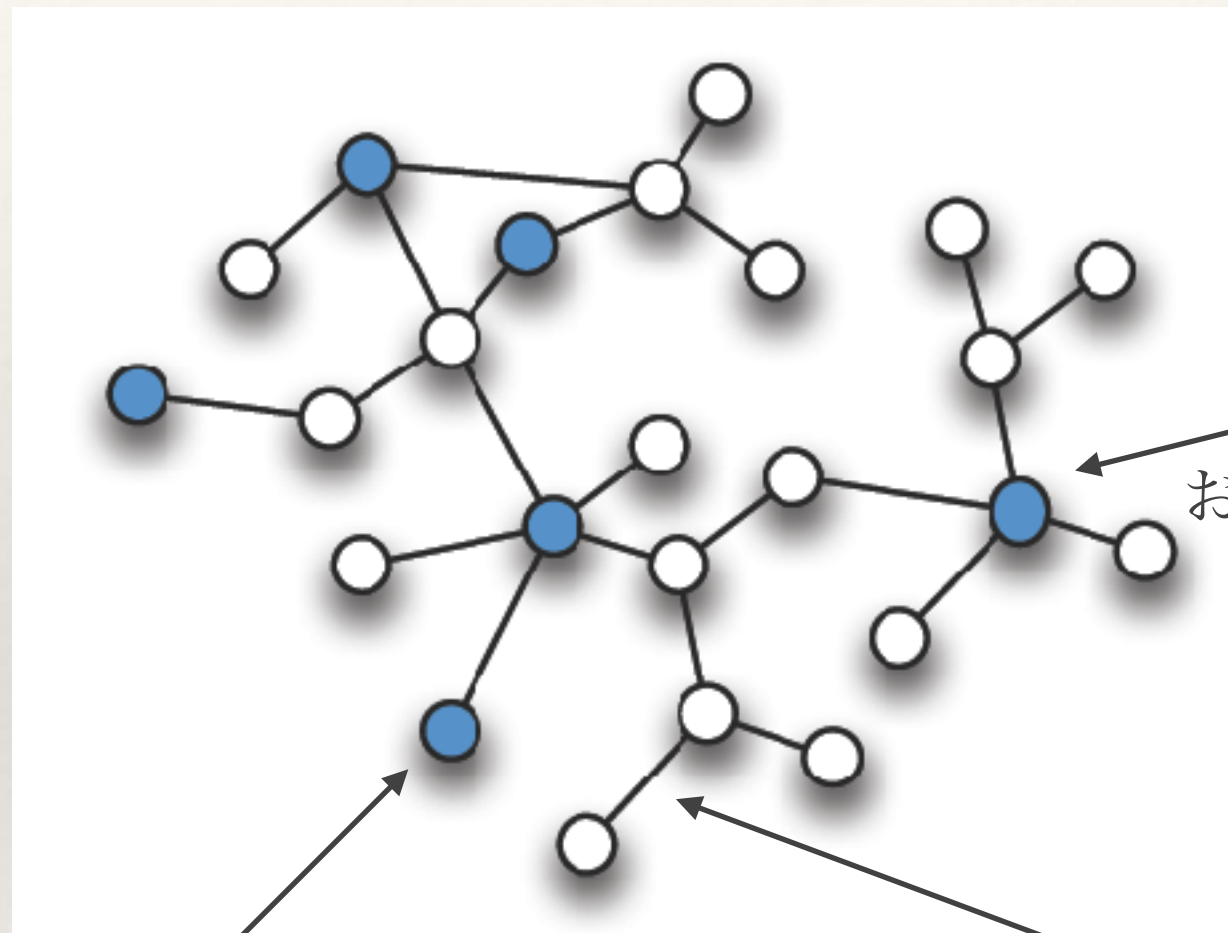
- ❖ 1975年生まれ 東京都足立区出身
- ❖ 工学部で、数理工学（応用数学）を学びました
- ❖ IT系ベンチャーでJavaを使ったWebアプリの開発を経て
- ❖ バイオインフォマティクス（生命科学と情報科学の融合分野）を学ぶため博士課程に入学
- ❖ 「Pythonスタートブック」の改訂版を準備中です

複雑ネットワーク科学

- ❖ 90年代おわりから2000年代初頭にブームとなった
- ❖ いろいろなものを、ノード（頂点）とエッジ（辺）でモデル化
 - ❖ WWW (world wide web)
 - ❖ 友達関係
 - ❖ タンパク質の生体内の相互作用
- ❖ 英語ですが、最高の教科書
 - ❖ barabasi.com/networksciencebook



ネットワークまたはグラフ



次数 (degree)

お隣さんの数。この場合は4

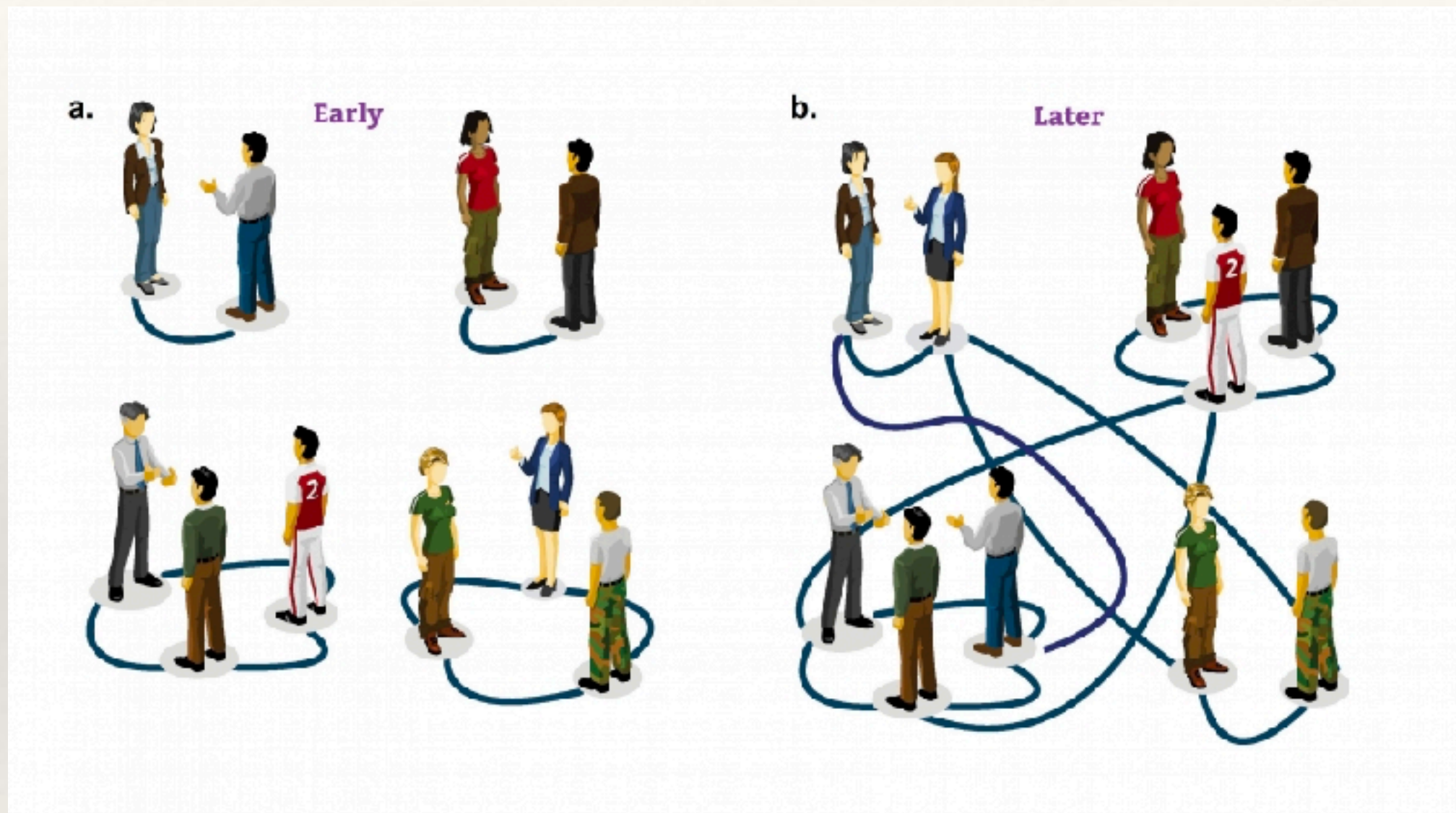
ノード (node)

エッジ (edge)

WWWならWebサイト、交友関係なら1人の人など

サイト間のリンクや友達関係など

6次の隔たり



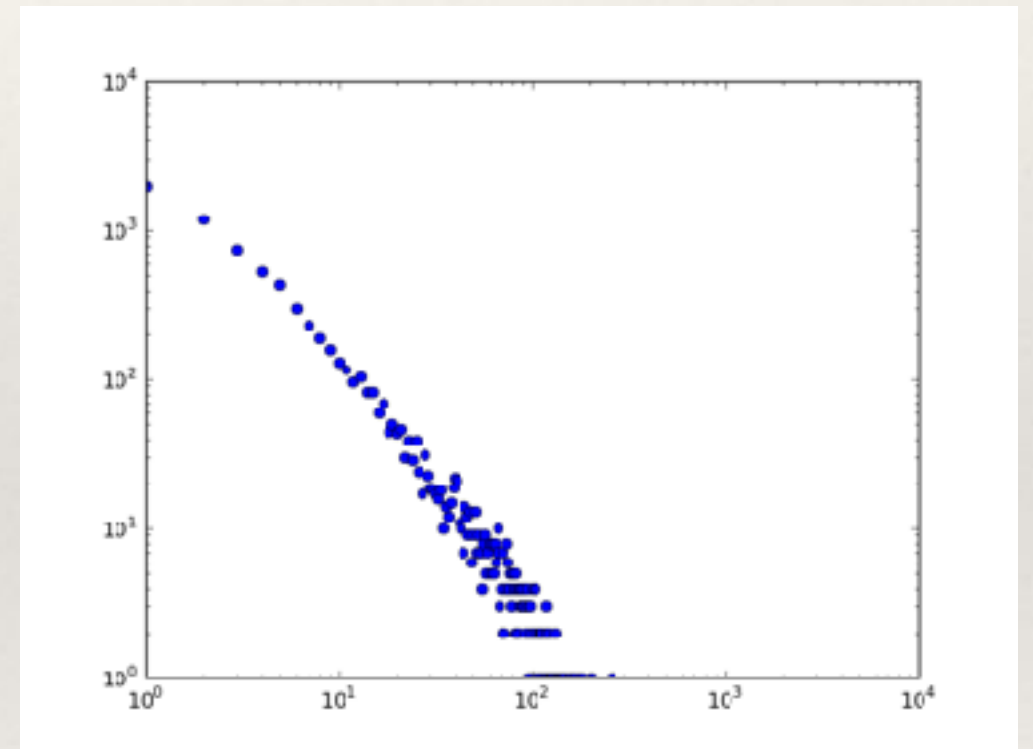
$$\text{2ノード間の平均的な距離} = \frac{\ln N}{\ln \langle k \rangle} = \frac{\ln 7 \times 10^9}{\ln 200} = 4.28 \dots$$

スケールフリーネットワーク

- ❖ 実際のネットワークは、ランダムネットワークではモデル化できない
- ❖ 発見はWeb
- ❖ 次数の分布がなめらかに変化
- ❖ 次数が極端に大きいノードは、ハブ (hub) と呼ばれる

実際のタンパク質相互作用ネットワーク

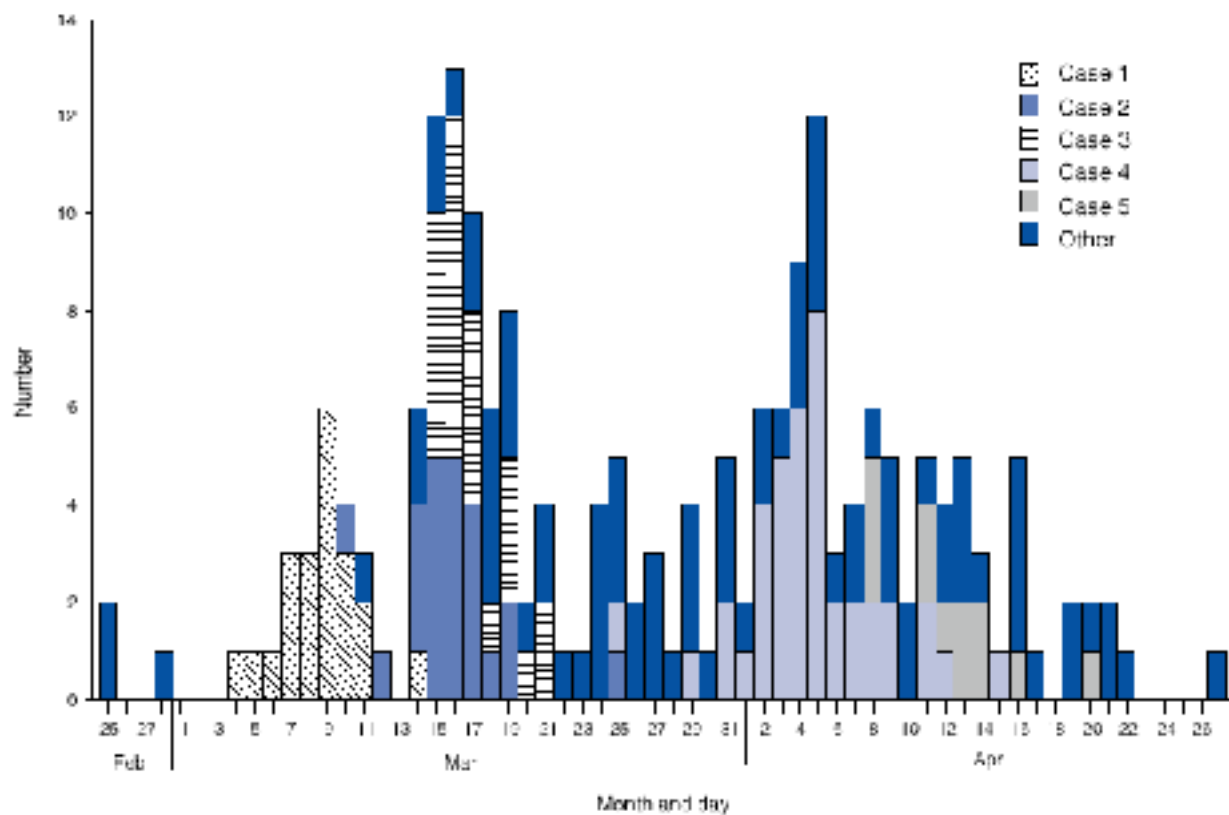
頻度



ノードの次数

SARSのSuper Spreader

FIGURE 1. Number of probable cases of severe acute respiratory syndrome,* by date of fever onset and reported source of infection — Singapore, February 25–April 30, 2003

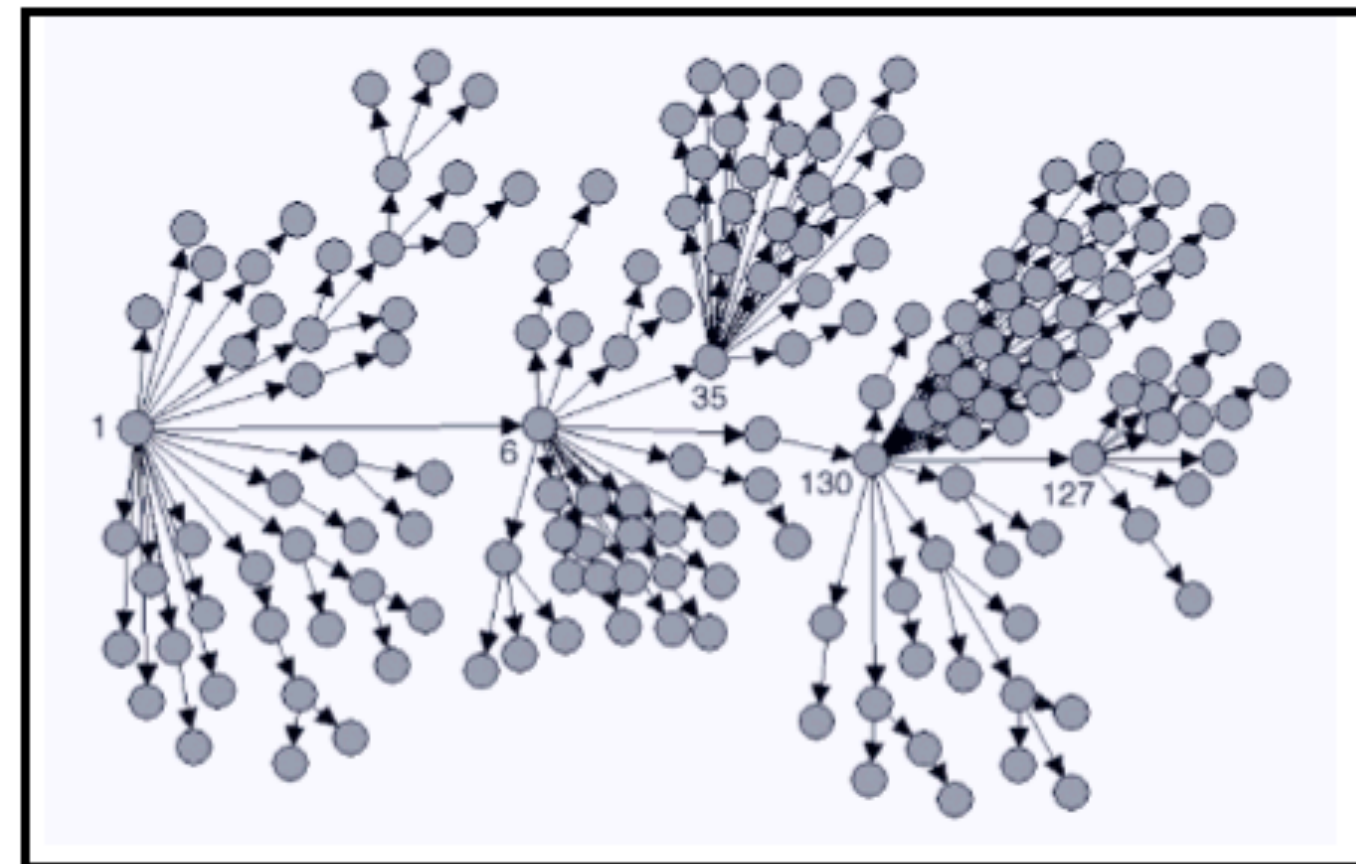


* n = 201.



Centers for Disease Control and Prevention
CDC 24/7: Saving Lives. Protecting People. Saving Money through Prevention.™

FIGURE 2. Probable cases of severe acute respiratory syndrome, by reported source of infection* — Singapore, February 25–April 30, 2003



* Patient 1 represents Case 1; Patient 6, Case 2; Patient 35, Case 3; Patient 130, Case 4; and Patient 127, Case 5. Excludes 22 cases with either no or poorly defined direct contacts or who were cases translocated to Singapore and the seven contacts of one of these cases.

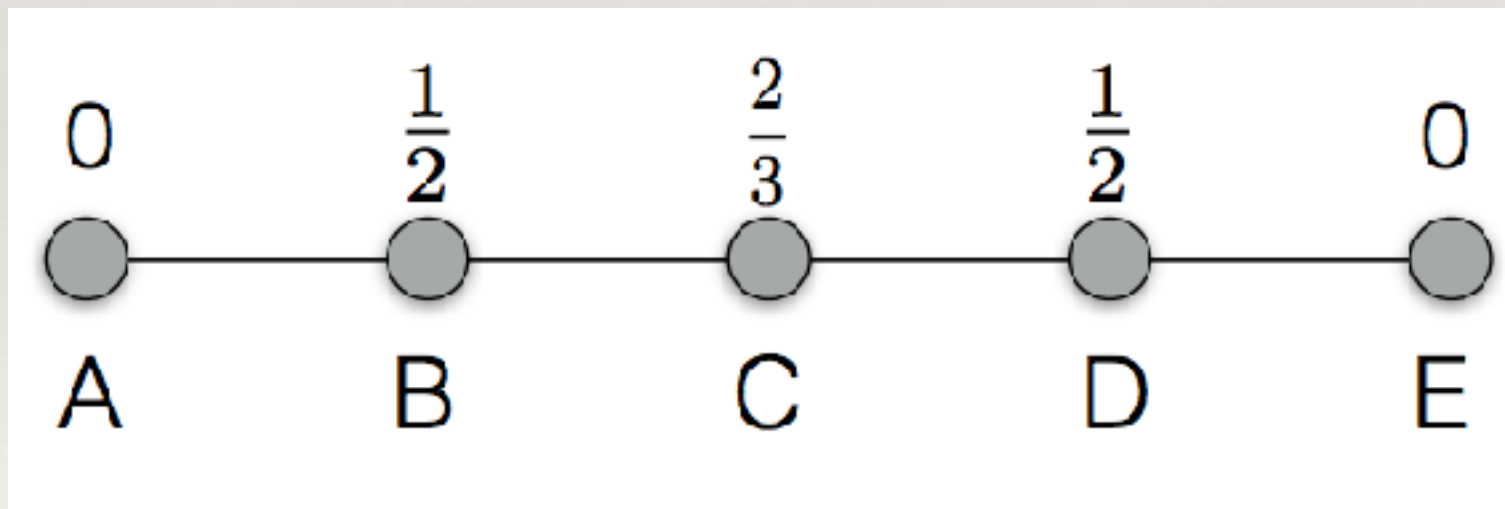
Reference: Bogatti SP. Netdraw 1.0 Network Visualization Software. Harvard, Massachusetts: Analytic Technologies, 2002.

極端に多くの人と接触がある人物を介して、原因ウィルスが急速に広まる様子をモデル化

Betweenness Centrality

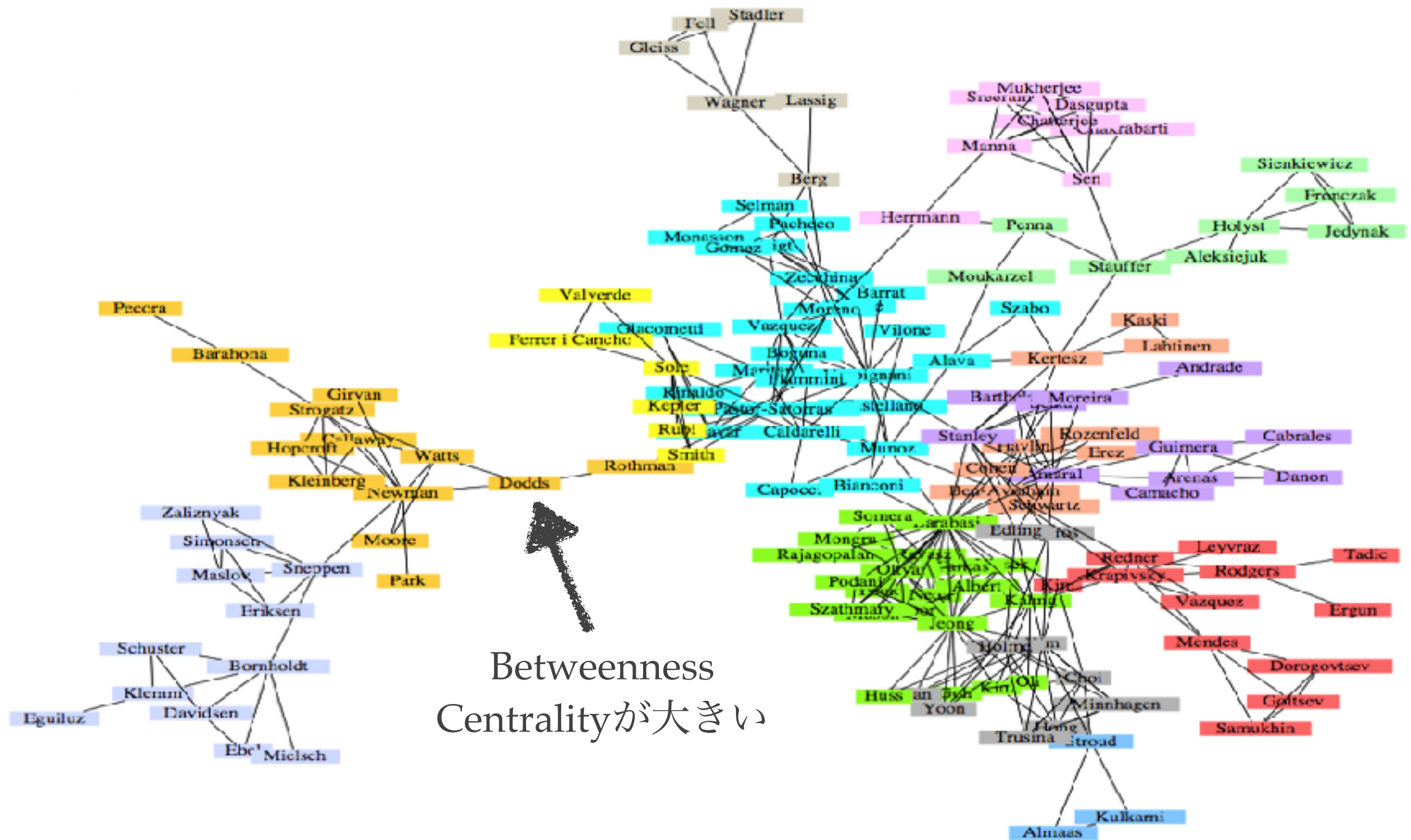
$$C_B(v_i) = \frac{\sum_{j < k} \frac{g_{jk}(v_i)}{g_{jk}}}{(N-1)(N-2)/2}$$

$g_{jk}(v_i)$ のなかで*i*を通る経路の数
 g_{jk} j と*k*を結ぶ最短経路の数

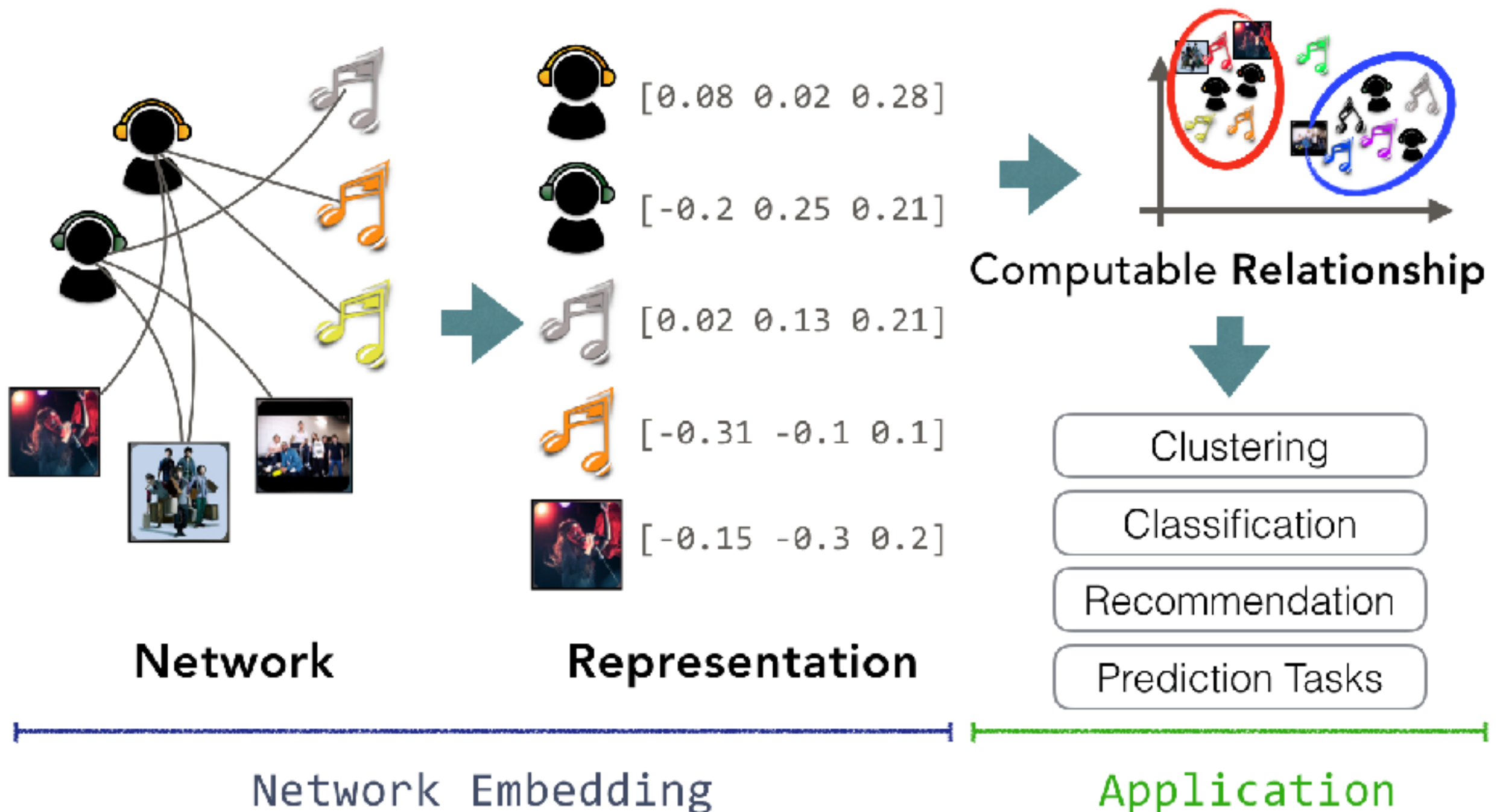


自分以外のノード間の接続に、自分がどれくらいの影響度合いで関わっているかを表現

コミュニティ検出への応用



Network embedding

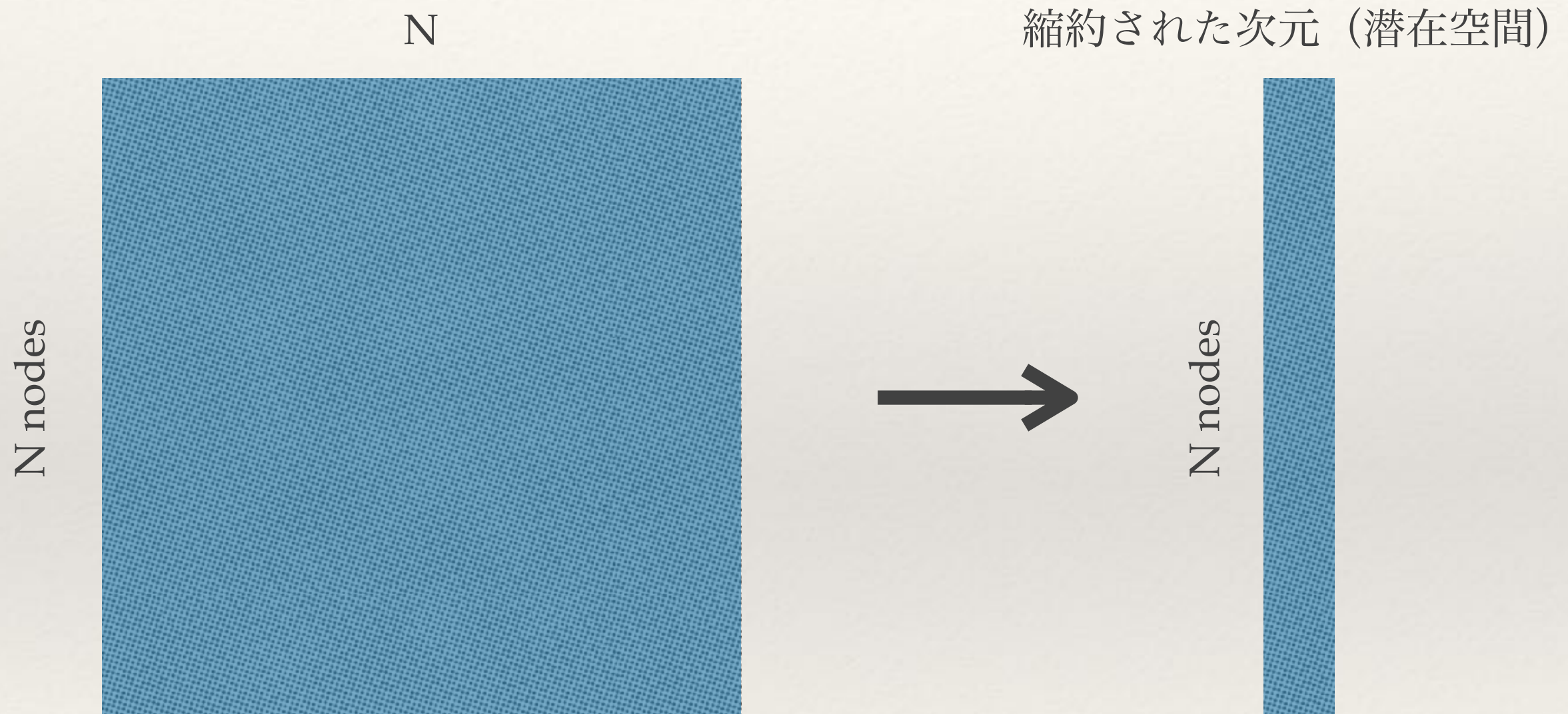


隣接行列

	node1	node2	node3	node4	node5	node6	node7	node8	node9	node10	node11	node12
node1				1					1			
node2			1					1				1
node3		1		1								
node4	1		1		1	1	1			1		
node5				1								
node6				1							1	
node7				1							1	
node8		1										
node9	1									1		1
node10				1					1			
node11						1	1					
node12		1							1			

node1の隣接ベクトルは、(0,0,0,1,0,0,0,0,1,0,0,0)の12次元ベクトル

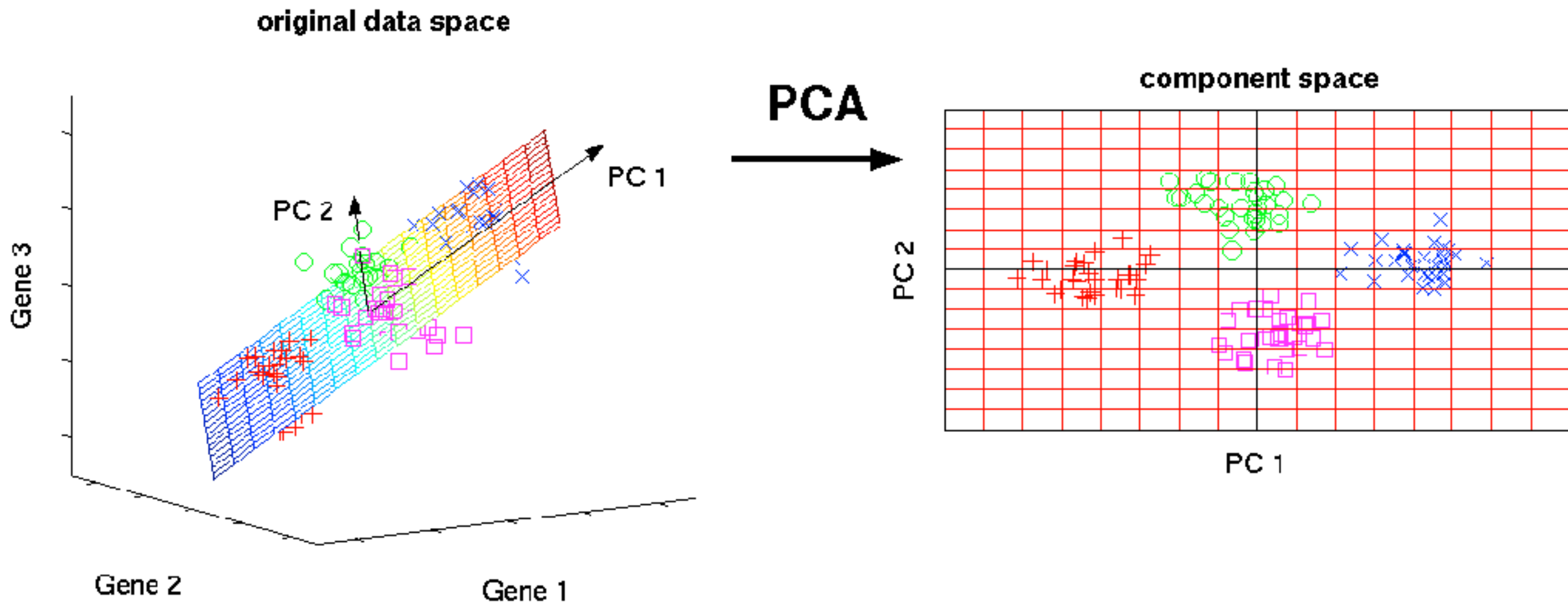
次元縮約



0が多くてスカスカ（スパース）な隣接ベクトルを、ぎゅっと濃縮

主成分分析 (PCA)

3次元が2次元へ縮約されている例



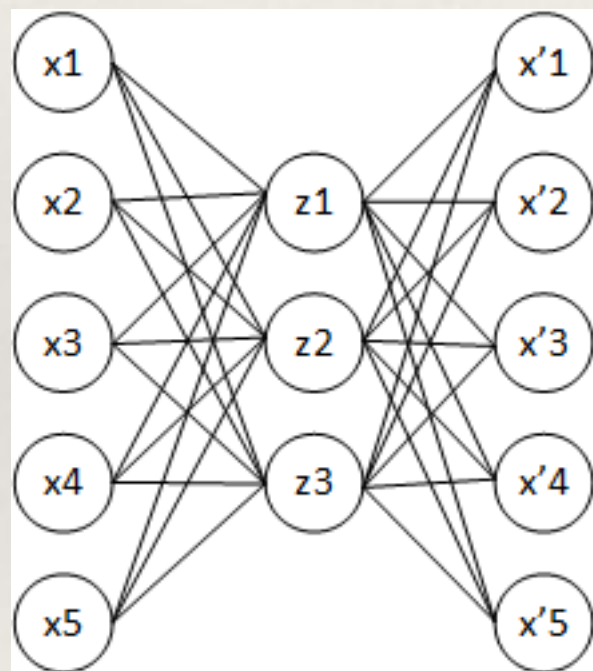
実際の計算は、古典的な行列の計算

Deep Learningを使う！

N次元を100次元にするDeep Autoencoder

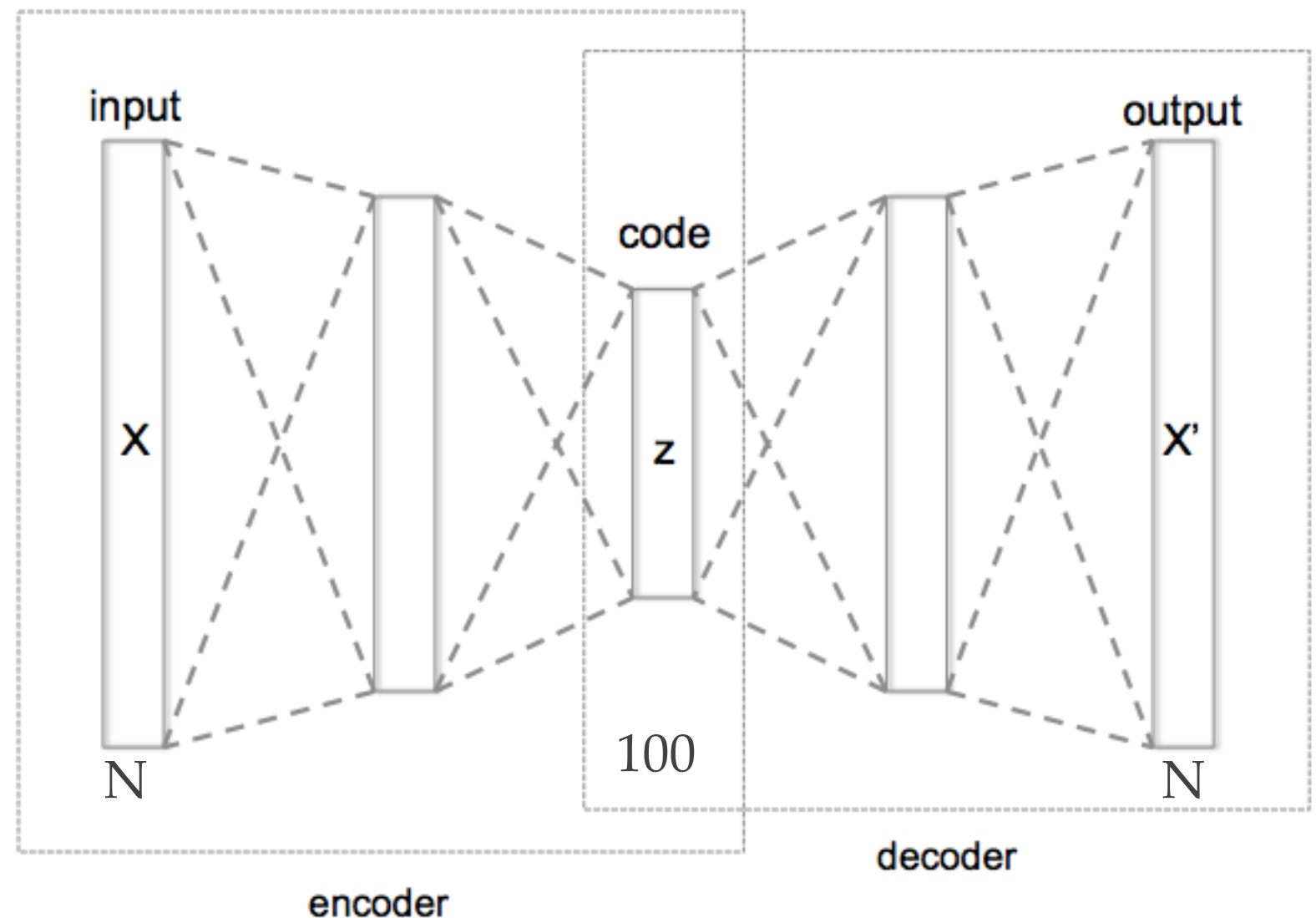
5次元を3次元にする

Autoencoder



入力

出力



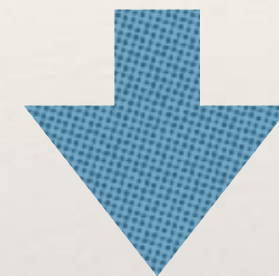
実際のデータ

N=8,502

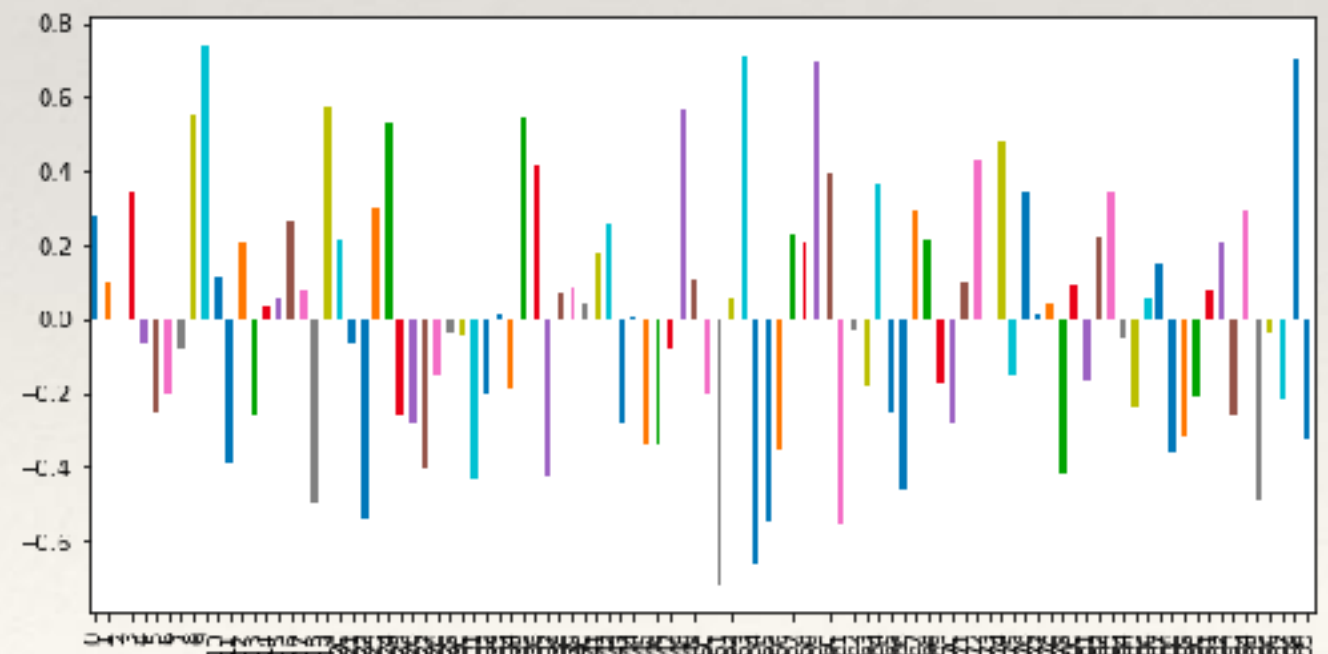


あるnodeの隣接ベクトル (8,502次元)

$(0,0,0,1,0,\dots,0,0,0,1)$



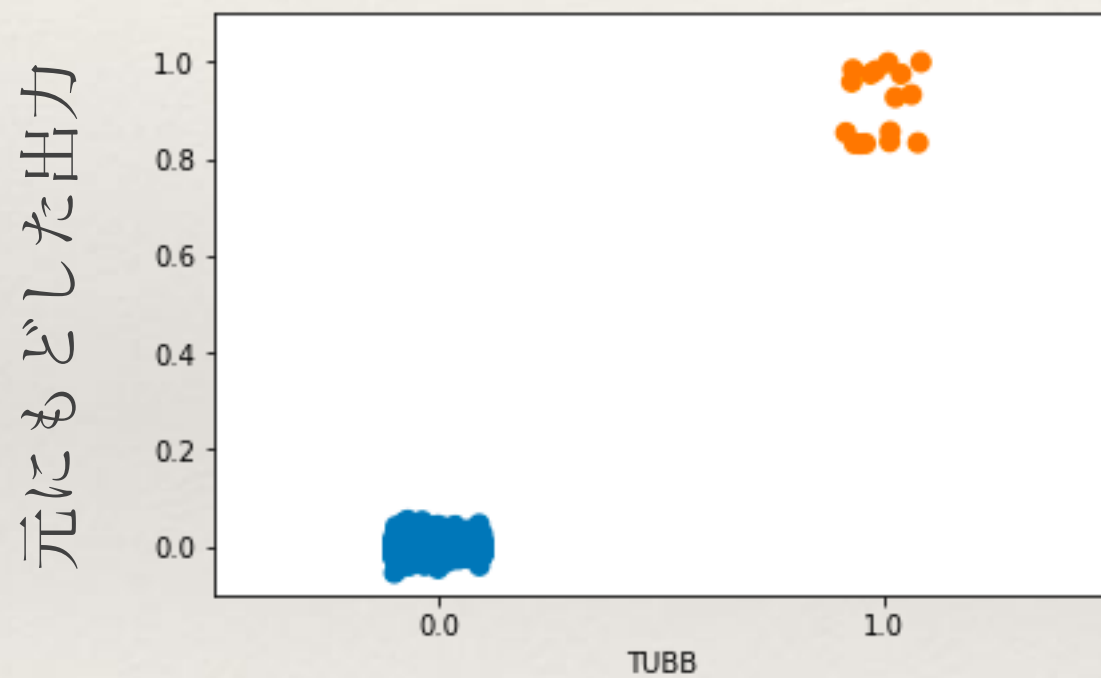
100次元ベクトルになる



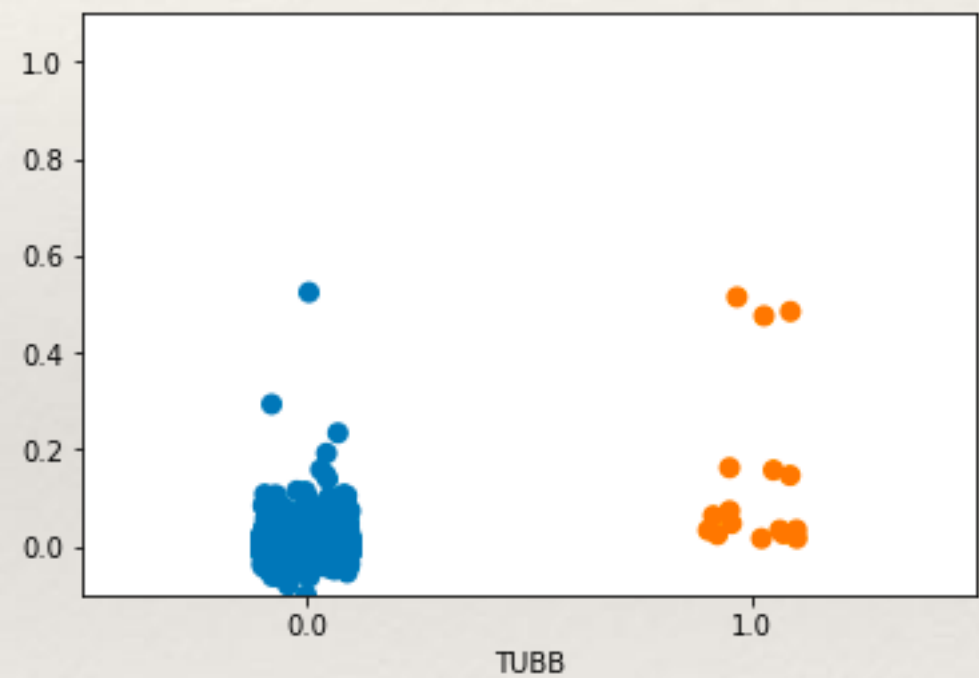
* ネットワークの図はイメージです。

どれくらい戻るのか

Deep Autoencoder

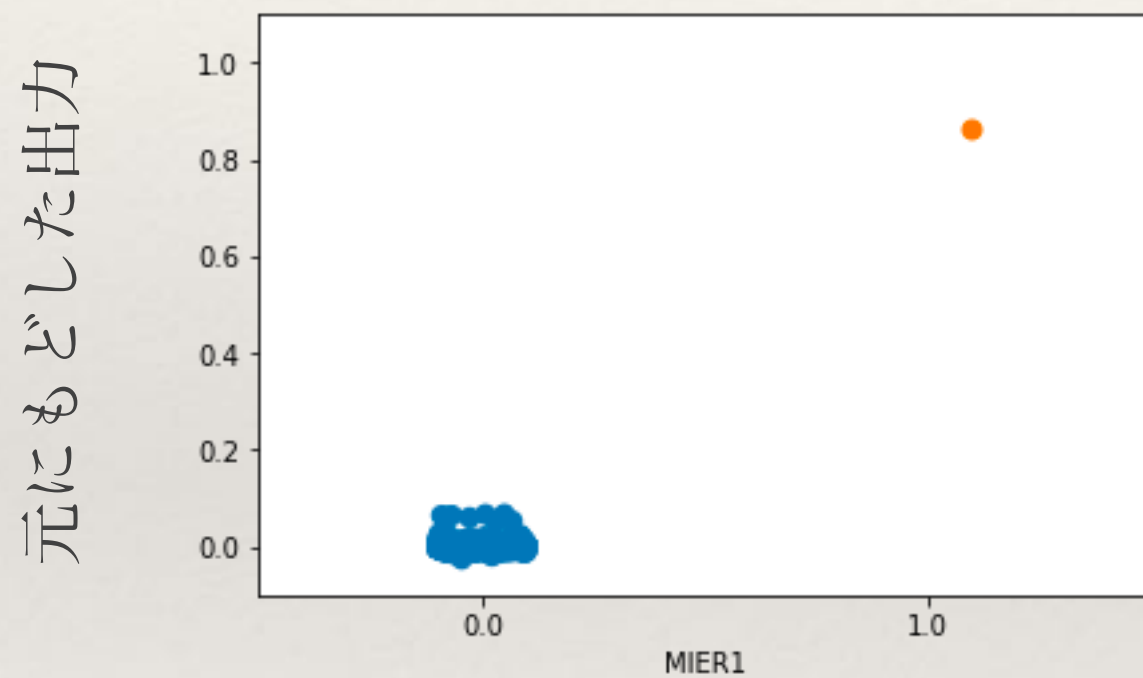


行列計算 (SVD)

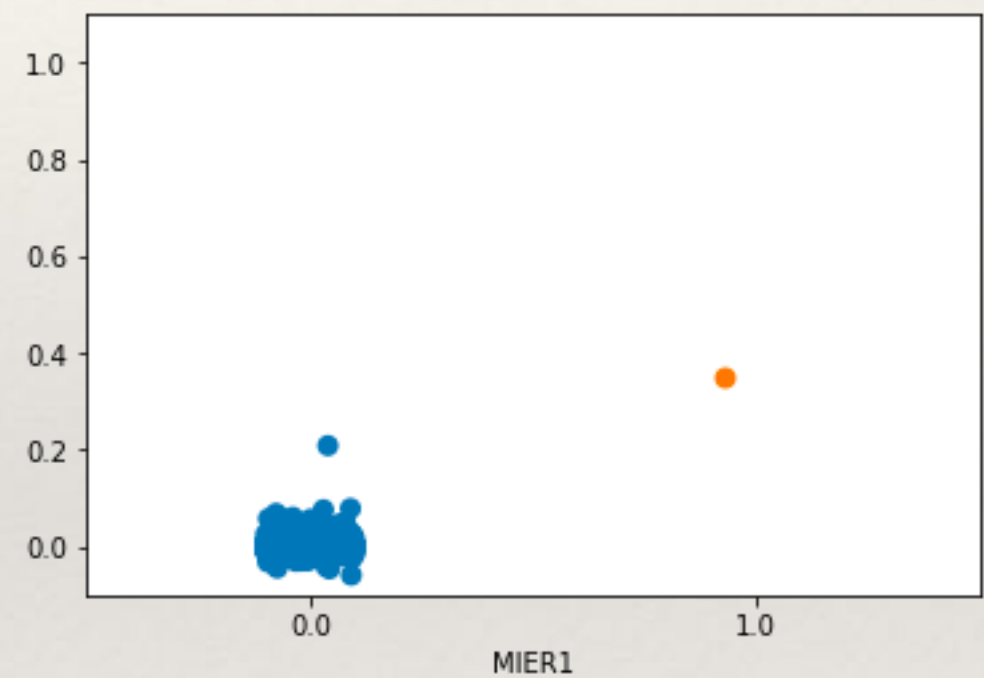


1/8502でも戻る

Deep Autoencoder



行列計算 (SVD)



Network embeddingの可能性



Network
Embedding

機械学習アルリズム
(各サンプルのベクトル
表現が必要)

これは使えそうだ！と思ったら、お気軽にお声かけ下さい。

Networkxが便利

NetworkX

Stable (notes)

2.1 — January 2018

[download](#) | [doc](#) | [pdf](#)

Latest (notes)

2.2 development

[github](#) | [doc](#) | [pdf](#)

Archive

Contact

[Mailing list](#)

[Issue tracker](#)



Software for complex networks

NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.



Features

- Data structures for graphs, digraphs, and multigraphs
- Many standard graph algorithms
- Network structure and analysis measures
- Generators for classic graphs, random graphs, and synthetic networks
- Nodes can be "anything" (e.g., text, images, XML records)
- Edges can hold arbitrary data (e.g., weights, time-series)
- Open source [3-clause BSD license](#)
- Well tested with over 90% code coverage
- Additional benefits from Python include fast prototyping, easy to teach, and multi-platform

まとめ

- ❖ 複雑ネットワーク科学という分野がある
- ❖ Network Embeddingが最近注目を集めている
- ❖ 次元縮約にDeep Autoencoderを使うといい
- ❖ 今後、いたるところで使われるようになる（気がする）