

第 5 章: 発見 (5.2. ネットワークデータ)

今井耕介 著

『社会科学のためのデータ分析入門 (QSS)』

2026-03-09

5.2 ネットワークデータ

社会ネットワーク分析

- ▶ **ノード (Nodes/Vertices):** 個人や組織などの実体。
- ▶ **エッジ (Edges/Links):** ノード間の関係 (友人、取引、引用など)。
- ▶ **目標:** ネットワークの構造を理解し、中心的な役割を果たすノードを特定する。

5.2.1 ルネサンス期フィレンツェの婚姻ネットワーク

florentine データの読み込み (1)

▶ フィレンツェの家族間の関係データを読み込みます。

1. ローカルに保存したデータの読み込み (推奨)

```
florence <- read.csv("florentine.csv", row.names = "FAMILY")
```

(参考) URL から直接読み込むことも可能

```
# florence <- read.csv("https://ayumu-tanaka.github.io/QSS/QSS_Data/florentine.c
```

隣接行列 (Adjacency Matrix) (2)

- ▶ ネットワークを行列形式で表現します。
- ▶ セルの値が 1 なら関係があり、0 ならありません。

```
# 最初の 5 家族分の隣接行列を表示
```

```
as.matrix(florence)[1:5, 1:5]
```

##	ACCIAIUOL	ALBIZZI	BARBADORI	BISCHERI	CASTELLAN
## ACCIAIUOL	0	0	0	0	0
## ALBIZZI	0	0	0	0	0
## BARBADORI	0	0	0	0	1
## BISCHERI	0	0	0	0	0
## CASTELLAN	0	0	1	0	0

5.2.2 グラフと中心性指標

igraph によるネットワークの作成

- ▶ **無向グラフ**: 関係に方向がない場合 (婚姻など)。

```
library(igraph)
# 隣接行列からグラフを作成
g <- graph_from_adjacency_matrix(as.matrix(florence), mode = "undirected")
# プロット
plot(g, main = "Marriage Network in Florence")
```

Marriage Network in Florence



中心性指標 (1): 次数中心性

▶ **次数中心性 (Degree):** 直接つながっているエッジの数。

つながりが多い家族トップ 3

```
sort(degree(g), decreasing = TRUE)[1:3]
```

```
##      MEDICI GUADAGNI STROZZI  
##           6         4         4
```

中心性指標 (2): 近接中心性

▶ **近接中心性 (Closeness):** 他の全てのノードへの距離の短さ。

他の家族へのアクセスが容易な家族

```
sort(closeness(g), decreasing = TRUE)[1:3]
```

```
##      MEDICI      RIDOLFI      ALBIZZI
```

```
## 0.04000000 0.03571429 0.03448276
```

中心性指標 (3): 間中心性

- ▶ **間中心性 (Betweenness)**: ノード間の最短経路上の通過回数。
- ▶ 「仲介者」としての重要度を測る。

```
# ネットワークのハブとなる家族  
sort(betweenness(g), decreasing = TRUE)[1:3]
```

```
## MEDICI GUADAGNI ALBIZZI  
## 47.50000 23.16667 19.33333
```

5.2.3 Twitter のフォローネットワーク

Twitter データの読み込み (1)

▶ 上院議員のフォロー関係データと属性データを読み込みます。

1. ローカルに保存したデータの読み込み (推奨)

```
twitter <- read.csv("twitter-following.csv", stringsAsFactors = FALSE)
```

```
senator <- read.csv("twitter-senator.csv", stringsAsFactors = FALSE)
```

(参考) URL から直接読み込むことも可能

```
# twitter <- read.csv("https://ayumu-tanaka.github.io/QSS/QSS_Data/twitter-follo
```

```
# senator <- read.csv("https://ayumu-tanaka.github.io/QSS/QSS_Data/twitter-senat
```

有向グラフ (Directed Graph) (2)

```
# 隣接行列の初期化と作成
n <- nrow(senator); twitter.adj <- matrix(0, nrow = n, ncol = n)
colnames(twitter.adj) <- rownames(twitter.adj) <- senator$screen_name
for (i in 1:nrow(twitter)) {
  twitter.adj[twitter$following[i], twitter$followed[i]] <- 1
}
# グラフオブジェクトの作成
g.tw <- graph_from_adjacency_matrix(twitter.adj, mode = "directed")
```

5.2.4 PageRank アルゴリズム

PageRank とは

- ▶ 重要なノードからフォローされているノードをより高く評価する指標。
- ▶ Google の検索エンジンでも使われる基本的なアルゴリズム。

```
# PageRank の計算
```

```
senator$pagerank <- page_rank(g.tw)$vector
```

```
# PageRank が高い上院議員トップ 3
```

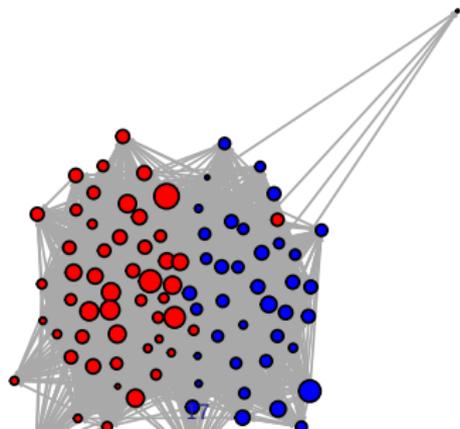
```
senator$name[order(senator$pagerank, decreasing = TRUE)][1:3]
```

```
## [1] "John McCain"      "John Cornyn"      "Martin Heinrich"
```

ネットワークの視覚化 (PageRank)

```
# 政党別の色分け (D=Blue, R=Red)
col <- ifelse(senator$party == "D", "blue", "red")
# PageRank に基づくサイズ調整
plot(g.tw, vertex.size = senator$pagerank * 500,
     vertex.color = col, vertex.label = NA,
     edge.arrow.size = 0.1, main = "US Senators Twitter Network")
```

US Senators Twitter Network



5.2.5 まとめ

まとめ

- ▶ **ネットワークデータ**: 要素間の「つながり」を隣接行列やグラフで表現する。
- ▶ **中心性指標**:
 - ▶ **次数**: つながりの多さ。
 - ▶ **近接**: 全体へのアクセスの良さ。
 - ▶ **間**: 仲介役としての重要性。
- ▶ **PageRank**: つながりの「質」を考慮した重要度。
- ▶ **視覚化**: グラフの描画によって直感的な理解を促す。