

R を利用した社会ネットワーク分析

分析例

1 はじめに

隣接行列から描画する方法を学んだり，あるいはその構造指標の理論を学んでも，それがいったい何の役に立つのか，どのように活かすのかはなかなか分からないものです。これは社会ネットワーク分析（SNA）のようなデータ解析に限らず，あらゆる学問に共通して言えることなのですが，実際にデータに触れて，自ら分析するという経験がなければ実感として身にはつかないものだと思います。

そこで，ここでは SNA の知識を利用して，実際のデータで分析する方法について説明します。なお，本章では以下の R のパッケージを使用しますので，最初にインストールしておいて下さい。

- sna
- latentnet

インストール終了後，以下のコマンドを実行します。

```
library(sna)
library(latentnet)
```

これで本章で説明する分析のための準備が整いました。

2 修道僧の友好関係の分析

2.1 使用するデータ

ここでは Sampson の修道院のデータ (Sampson, 1968) を利用して，SNA を利用した分析方法の解説を行います。Sampson (Sampson, S.F) が知覚の実験で，ある修道院に滞在していたとき，修道僧間で政治的な衝突が生じ，数人が修道院から追放・自主退院になるという事件が生じたそうです。

その事件以前に Sampson は，修道僧同士の好意関係を測定していたので，離反事件の背後にある好意関係に基づくネットワークの考察ができました。この事件における隣接行列は `library(latentnet)` 内の `sampson` というオブジェクトに対して

```
data(sampson)
dat <- as.matrix.network(samplike)
dat
```

とすることで得られます。出力結果を整理すると以下のような隣接行列になります。

```
Romul    0 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0
Bonaven  0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 0 0 0
Ambrose  0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0
Berth    0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0
Peter    1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0
Louis    0 1 0 0 1 0 1 0 0 0 1 0 0 1 0 0 0 0
Victor   1 0 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0
Winf     0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0
John     0 1 0 0 0 0 1 1 0 1 1 0 0 0 0 1 0 0
Greg     0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0 0 0
Hugh     0 0 0 0 0 1 0 1 1 1 0 1 0 0 0 0 0 0
Boni     0 1 0 0 0 0 0 1 1 1 1 0 1 0 0 0 0 0
Mark     0 0 0 0 0 0 1 1 1 1 0 0 0 1 0 0 0 0
Albert   0 0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 0 0
Amand    0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1
Basil    0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 1 1
Elias    0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1
Simp     0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 1 1 0
```

この修道院からまず最初に追放された修道僧は、Greg、Basil、Elias、Simpの4名でした。また彼らの追放を受けて、すぐにJohn、Mark、Hugh、Boni、Albertの5名が自主退院したのだそうです。結局最後までとどまったのはBonaven、Berth、Ambrose、Louisの4名だったそうです。

2.2 データの読み込み

上記で示した修道僧の友好関係データを『sampson.txt』のように外部ファイルとして準備されているとします。このファイルを『C:\work』に保存します。これを行列としてRに読み込みましょう。コマンドは以下のようになります。

```
dat <- read.table("C:\\work\\sampson.txt", header=T, row.names=1)
dat
```

2.3 ネットワークの描画

次にこの隣接行列のオブジェクトをラベル付きの有向グラフで描画してみましょう。コマンドは以下になります。

```
# 力学的アルゴリズムによる配置
gplot(dat, mode="fruchtermanreingold", displaylabels=T,
      vertex.cex=5, vertex.col="white", label.pos=5, label.cex=0.7)
# 円環状に配置
gplot(dat, mode="circle", displaylabels=T,
      vertex.cex=5, vertex.col="white", label.pos=5, label.cex=0.7)
```

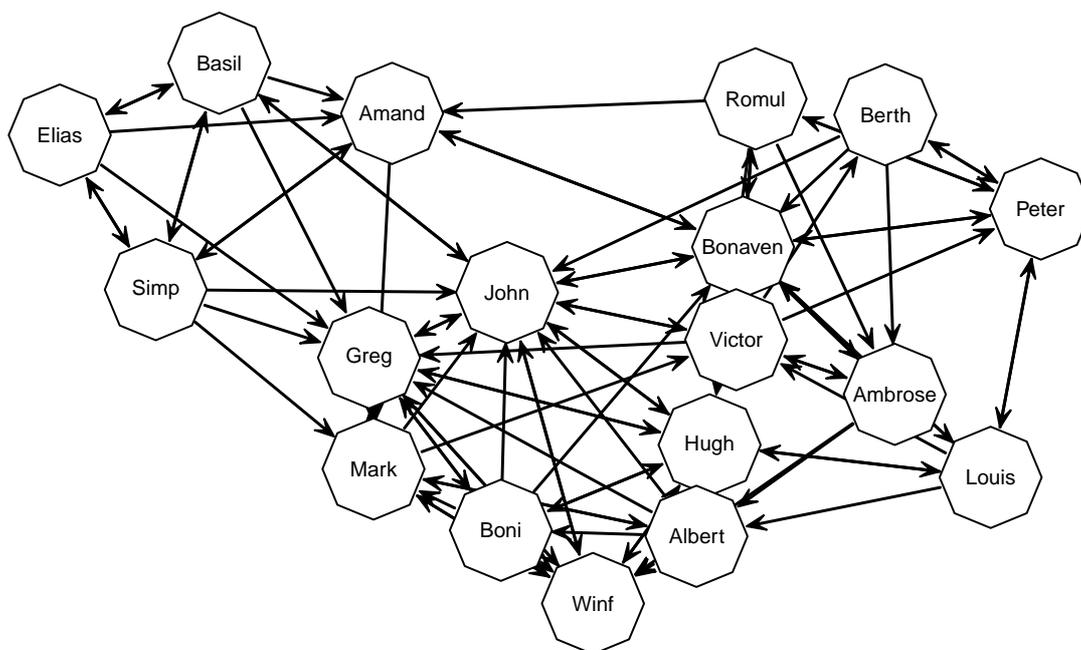


図1 修道僧の友好関係ネットワーク1 (力学的アルゴリズムによる配置)

図1は力学的アルゴリズムによる配置(引数 `mode="fruchtermanreingold"` を使用), 図2は円環状に配置するアルゴリズム(引数 `mode="circle"`)を使用して描画しました。どちらのグラフが見やすいかは一概には言えませんが, 今回の例で言えば図1の方が比較の見やすいと思います。なお, ここで言う『見やすい』とは, ネットワークの構造, すなわち関係性を把握しやすいという意味です。

目視でも, 『Bonaven』『John』『Greg』あたりに多くの有向辺が刺さっていることがわかります。おそらくこの3人が修道院内で多くの権力を持っており, 政治的闘争が起こったのでしょう。この3人のうちBonavenのみが残ったところを鑑みるに, 最終的な権力

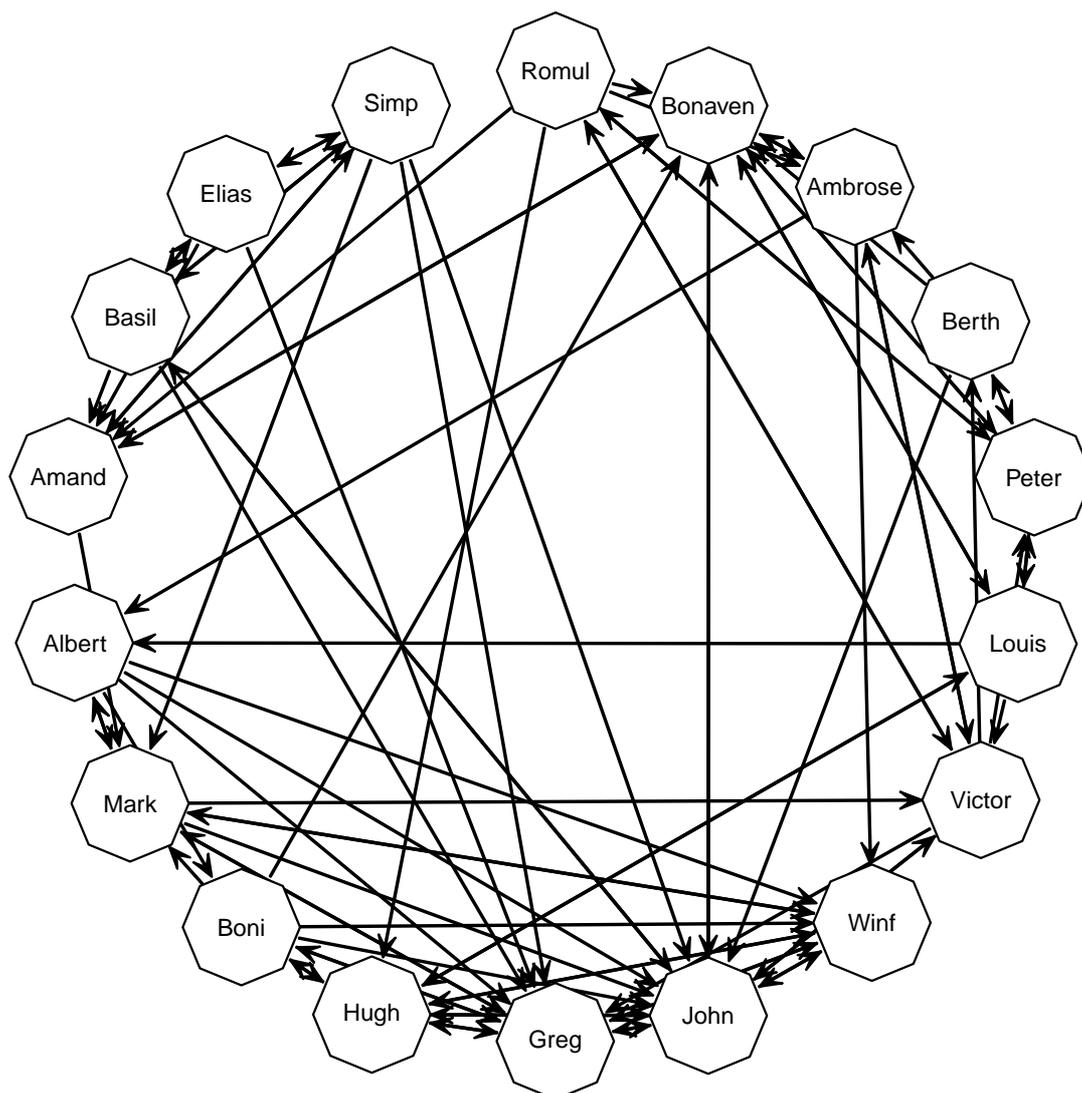


図 2 修道僧の友好関係ネットワーク 2 (円環状に配置)

闘争に勝ち残ったのは Bonaven 一派だったということです。

これでもかなりの情報が得られたわけですが、ネットワークの描画だけだとこれ以上の考察はできません。そこで次に、このネットワークの構造を特徴付ける様々な指標を見ていきましょう。

2.4 ネットワーク構造の諸指標

2.4.1 距離

とりあえず最初に、ノード間の距離を求めてみましょう。使用する関数は `geodist` です。

```
geodist(dat)
```

上記のコマンドを実行すると以下のような出力が得られます。(ただし、資料に入りきら

なかったため若干編集してあります。)

```

$counts
 [1,] [2,] [3,] [4,] [5,] [6,] [7,] [8,] [9,] [10,] [11,] [12,] [13,] [14,] [15,] [16,] [17,] [18,]
[1,] 1 1 1 2 1 3 1 2 3 2 1 1 1 1 1 4 1 1
[2,] 1 1 1 1 1 1 3 2 1 1 2 5 1 2 1 1 2 1
[3,] 1 1 1 1 2 1 1 1 4 3 1 1 2 1 1 4 5 1
[4,] 1 1 1 1 1 2 2 2 1 1 1 3 5 1 1 1 1 2
[5,] 1 1 3 1 1 1 2 8 2 7 2 3 3 1 2 2 4 2
[6,] 2 1 2 2 1 1 1 2 4 3 1 2 1 1 1 4 5 1
[7,] 1 5 1 1 1 1 1 3 1 1 3 1 1 1 1 1 1 2
[8,] 2 1 3 2 4 1 2 1 1 1 1 2 1 1 2 1 1 1
[9,] 1 1 2 1 2 2 1 1 1 1 1 2 2 6 2 1 1 1
[10,] 2 2 4 2 5 1 2 1 1 1 1 1 1 1 3 1 1 1
[11,] 3 3 5 3 1 1 2 1 1 1 1 1 3 1 4 1 1 1
[12,] 3 1 1 3 1 2 2 1 1 1 1 1 1 1 1 1 1 2
[13,] 1 1 1 1 1 5 1 1 1 1 3 2 1 1 3 1 1 1
[14,] 2 2 4 2 4 6 2 1 1 1 4 1 1 1 3 1 1 1
[15,] 2 1 1 2 1 1 1 1 3 2 7 3 1 1 1 1 1 1
[16,] 1 2 3 1 3 4 1 2 1 2 1 3 3 1 1 1 1 1
[17,] 7 1 1 7 1 2 6 1 3 1 1 1 3 3 1 1 1 1
[18,] 2 2 4 2 4 4 2 3 1 1 2 1 1 1 1 1 1 1

$gdist
 [1,] [2,] [3,] [4,] [5,] [6,] [7,] [8,] [9,] [10,] [11,] [12,] [13,] [14,] [15,] [16,] [17,] [18,]
[1,] 0 1 1 2 1 2 1 2 2 2 1 2 2 2 1 3 3 2
[2,] 2 0 1 2 1 1 2 2 1 2 2 3 2 2 1 2 3 2
[3,] 2 1 0 2 2 2 1 1 2 2 2 2 2 1 2 3 4 3
[4,] 2 1 1 0 1 2 2 2 1 2 2 3 3 2 2 2 3 3
[5,] 1 1 2 1 0 1 2 3 2 3 2 3 3 2 2 3 4 3
[6,] 2 1 2 2 1 0 1 2 2 2 1 2 2 1 2 3 4 3
[7,] 1 2 1 1 1 2 0 2 1 1 2 2 2 2 2 2 3 3
[8,] 3 2 3 3 3 2 2 0 1 1 1 2 1 2 3 2 3 3
[9,] 2 1 2 2 2 2 1 1 0 1 1 2 2 3 2 1 2 2
[10,] 3 2 3 3 3 2 2 1 1 0 1 1 1 2 3 2 3 3
[11,] 3 2 3 3 2 1 2 1 1 1 0 1 2 2 3 2 3 3
[12,] 3 1 2 3 2 2 2 1 1 1 1 0 1 2 2 2 3 3
[13,] 2 2 2 2 2 3 1 1 1 1 2 2 0 1 3 2 3 3
[14,] 3 2 3 3 3 3 2 1 1 1 2 1 1 0 3 2 3 3
[15,] 3 1 2 3 2 2 2 2 2 2 3 3 1 2 0 2 2 1
[16,] 3 2 3 3 3 3 2 2 1 1 2 2 2 3 1 0 1 1
[17,] 4 2 3 4 3 3 3 2 2 1 2 2 2 3 1 1 0 1
[18,] 3 2 3 3 3 3 2 2 1 1 2 2 1 2 1 1 1 0

```

ここで [1] ~ [18] はそれぞれ 『1.Romul』 『2.Bonaven』 『3.Ambrose』 『4.Berth』 『5.Peter』 『6.Louis』 『7.Victor』 『8.Winf』 『9.John』 『10.Greg』 『11.Hugh』 『12.Boni』 『13.Mark』 『14.Albert』 『15.Amand』 『16.Basil』 『17.Elias』 『18.Simp』 を表しています。

ノード間の最短距離を知るには \$gdist を見ます。最後まで残った権力者 『2.Bonaven』 の行を見ると最も遠いノードは値が 3 の 『12.Boni』 と 『17.Elias』 です。『17.Elias』 は権力闘争の最初の頃に追放された人物でした。また 『12.Boni』 はすぐに自主退院した修道僧です。組織で生き残るためには、権力者から遠すぎてはダメということでしょうか？

また、最も近いノードは値が 1 の 『3.Ambrose』 『5.Peter』 『6.Louis』 『9.John』 『15.Amand』 の 5 名です。このうち 『5.Peter』 『9.John』 『15.Amand』 の 3 名は追放・自主退院に追い込まれています。最終的に残った 『3.Ambrose』 と 『6.Louis』 は 『2.Bonaven』 の側近ないしは取り巻きだった可能性が見て取れます。

一方、『2.Bonaven』 と 『3.Ambrose』 『6.Louis』 以外に最後まで修道院に残ったのは 『4.Berth』 です。この人物と 『2.Bonaven』 との距離は 2 です。権力者と近すぎず、遠すぎずの位置を確保し、上手く立ち回ったためにこの組織で生き残った、処世の上手い人物

だったのでしょう。

2.4.2 密度

次に密度を求めてみます。密度は本来、2つ以上のネットワークを比較するのに用いられるもので、今回のようにたった1つのネットワークの密度を求めても仕方がないのですが、参考までに計算してみましょう。コマンドは以下になります。

```
gden(dat, mode="digraph")
```

この結果、『0.2875817』となりました。密度 d はグラフに含まれる頂点数に関係なく $0 \leq d \leq 1$ の範囲で動きますので、今回の結果はかなり密度は低かったと言えるでしょう。

これは逆に言えば、有向辺を引く余地があった、すなわちもっと友好関係を築ける可能性もあったということです。そうしていれば、修道院の内部事情はより違った結果になったかもしれません。

2.4.3 相互性

さらに相互性を調べてみます。が、その前に、まずは有向辺の種類と数を見てみましょう。

```
> dyad.census(dat)
      Mut Asym Null
[1,]  28   32   93
```

互いに友好関係にあるのは28本、一方通行の関係は32本、互いに関係なしなのは93本という結果でした。修道院の大きさがどれほどなのか分かりませんが、修道僧18名しかないような組織でこのような結果はかなり歪(いびつ)な構造であるような気がします。この修道院では日本的な『和』の状態ではなく、支配・被支配あるいは依存する・されるような状態にあったということではないかと推察されます。

相互性スコアを求めると以下のようにになりました。

```
> grecip(dat, measure="dyadic.nonnull")
      Mut
0.4666667
```

この結果も、一部の権力者が相互に結びつき、その周りに追従者がいるという構造を反映しているのではないのでしょうか。

2.5 ブロックモデリング

追放された者, 自主退院した者, 院にとどまった者は, それぞれ同じような役割を演じるノードであると推測できますが, この仮説を検証する為にブロックモデリングを実行してみましよう。

```
# クラスタ分析 (ユークリッド距離, ウォード法を使用) のオブジェクトを作成
clust <- equiv.clust(dat, method="euclidean", mode="digraph",
                    cluster.method="ward")

# ブロック数を 4 にして実行
blockmodel(dat, ec=clust, k=4, mode="digraph")
```

結果は以下のようになりました。(資料に入りきらないため多少編集してあります。)

```
Network Blockmodel:
Block membership:
Romul   Bonaven  Ambrose  Berth   Peter   Louis   Victor
  1         2         1         1         1         1         2
Winf    John     Greg     Hugh    Boni    Mark    Albert
  3         3         3         3         3         3         3
Amand   Basil   Elias    Simp
  4         4         4         4

Reduced form blockmodel:
          Block 1    Block 2    Block 3    Block 4
Block 1  0.40000000  0.8000000  0.1714286  0.05000000
Block 2  0.70000000  0.0000000  0.2142857  0.12500000
Block 3  0.02857143  0.2857143  0.7142857  0.03571429
Block 4  0.00000000  0.1250000  0.2500000  0.83333333
```

まず Block4 に分類された人たちは『Amand』を除き, すべて最初に追放された修道僧です。また, 続いて自主退院した人たち (『John』『Mark』『Hugh』『Boni』『Albert』) がすべて Block3 に分類されていることが分かります。当初の仮説では『Bonaven』が権力者の位置づけとしていましたが, それは Block2 で表されています。『Victor』もここにカテゴライズされていますが, 上に立つという役割は 2 人も必要なかったため, 権力闘争の途中で追い出されたのでしょう。そして Block1 が『Bonaven』の追随者の位置づけのブロックになると考えられます。その証拠に最後まで生き残った『Berth』『Ambrose』『Louis』がこのブロックにカテゴライズされています。

このことは縮約されたブロック行列からも解釈できます。【Reduced form block-model:】を見てみると権力者ブロックである Block2 と最も関係性が高いのは Block1 です（値は 0.7）。それに対して Block3 や Block4 はそれぞれ 0.214, 0.125 というように Block2 との関係性は低いです。Block1 にいることが、この組織（修道院）で生き残るための必要条件だったのでしょね。

3 フィレンツェ貴族間の婚姻関係の分析

3.1 使用するデータ

1300年から1500年にかけてのフィレンツェと言えば、経済的に大きく発展したことで有名です。この時期に経済発展した理由は、フィレンツェの長官により発行された債務や新税収システムなどのような、いくつかの国家的な財務政策が成功したからです。

また、文化的にも大きな躍進を見せました。いわゆるルネサンスです。建築、絵画、彫刻におけるルネサンス芸術は、15世紀をとおして大きく開花し、ボッティチェリ、レオナルド・ダ・ヴィンチ、ミケランジェロ、ラファエロなどの巨匠が活躍するルネサンス文化の中心地となって学問・芸術の大輪の花が開きました。

一方、政治体制も特徴的です。この頃のヨーロッパと言えば、王家を中心とする絶対主義が政治の中心でした。そんな中フィレンツェは、現代から比べると未熟とはいえ、選挙による共和体制を採用していました。これは15世紀のヨーロッパの中で異色を放っていたと言えます。

さて、フィレンツェの共和制は有力貴族によるものでした。例えば、メディチ家などがかなり有名です。当然、政略結婚などが行われ、貴族間でさまざまな婚姻関係が結ばれました。ここでは1282年～1500年代におけるフィレンツェ貴族間の婚姻関係の隣接行列 (Padgett,1994) を取り上げ、SNA を使って分析してみましょう。まず始めにデータをお示しします。

Acciaiuoli	0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
Albizzi	0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0
Barbadori	0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0
Bischeri	0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0
Castellani	0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0
Ginori	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Guadagni	0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1
Lamberteschi	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
Medici	1 1 1 0 0 0 0 0 0 0 0 0 1 1 0 1
Pazzi	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
Peruzzi	0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0
Pucci	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Ridolfi	0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1
Salviati	0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
Strozzi	0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0
Tornabuoni	0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0

全部で 16 家あります。ルネサンス時代に慣れ親しんだ人であればこのままでも良いのですが、そうでない人にはなかなか発音しにくい名前ですね。そこで、以下にカタカナ表記を書いておきます。

表 1 フィレンツェ貴族名のカタカナ表記

名前	読み方	名前	読み方
1 Acciaiuoli	アッチャイウオリ家	9 Medici	メディチ家
2 Albizzi	アルビッツィ家	10 Pazzi	パッツィ家
3 Barbadori	バルバドーリ家	11 Peruzzi	ペルッツィ家
4 Bischeri	ビスケリ家	12 Pucci	ブッチ家
5 Castellani	カステラーニ家	13 Ridolfi	リドルフィ家
6 Ginori	ジノリ家	14 Salviati	サルヴィアーティ家
7 Guadagni	グアダーニ家	15 Strozzi	ストロッツィ家
8 Lamberteschi	ランバーテスキ家	16 Tornabuoni	トルナブオーニ家

なお、この隣接行列は『両家の婚姻関係』を表していますので、対称行列になっています。したがって、SNA では無向グラフとして分析する必要があります。

3.2 データの読み込み

このデータも R (パッケージ `latentnet`) の中にありますので、今回は外部ファイルとはせず、直接読み込んでみましょう。コマンドは以下になります。

```
data(flo)
flo
```

上記のように `flo` というオブジェクトに格納されています。今後はこのオブジェクトを隣接行列として指定すれば良いでしょう。

3.3 ネットワークの描画

それでは早速ネットワークの描画から始めてみましょう。今回は無向グラフを描くので引数 `gmode` を `"graph"` にしてあります。

```
# 2次元描画
gplot(flo, gmode="graph", displaylabels=T)
```

描画されたネットワークを図 3 に示します。

この図を見て最初に気づくことは、ブッチ家だけがどことも婚姻関係を結ばず、孤立し

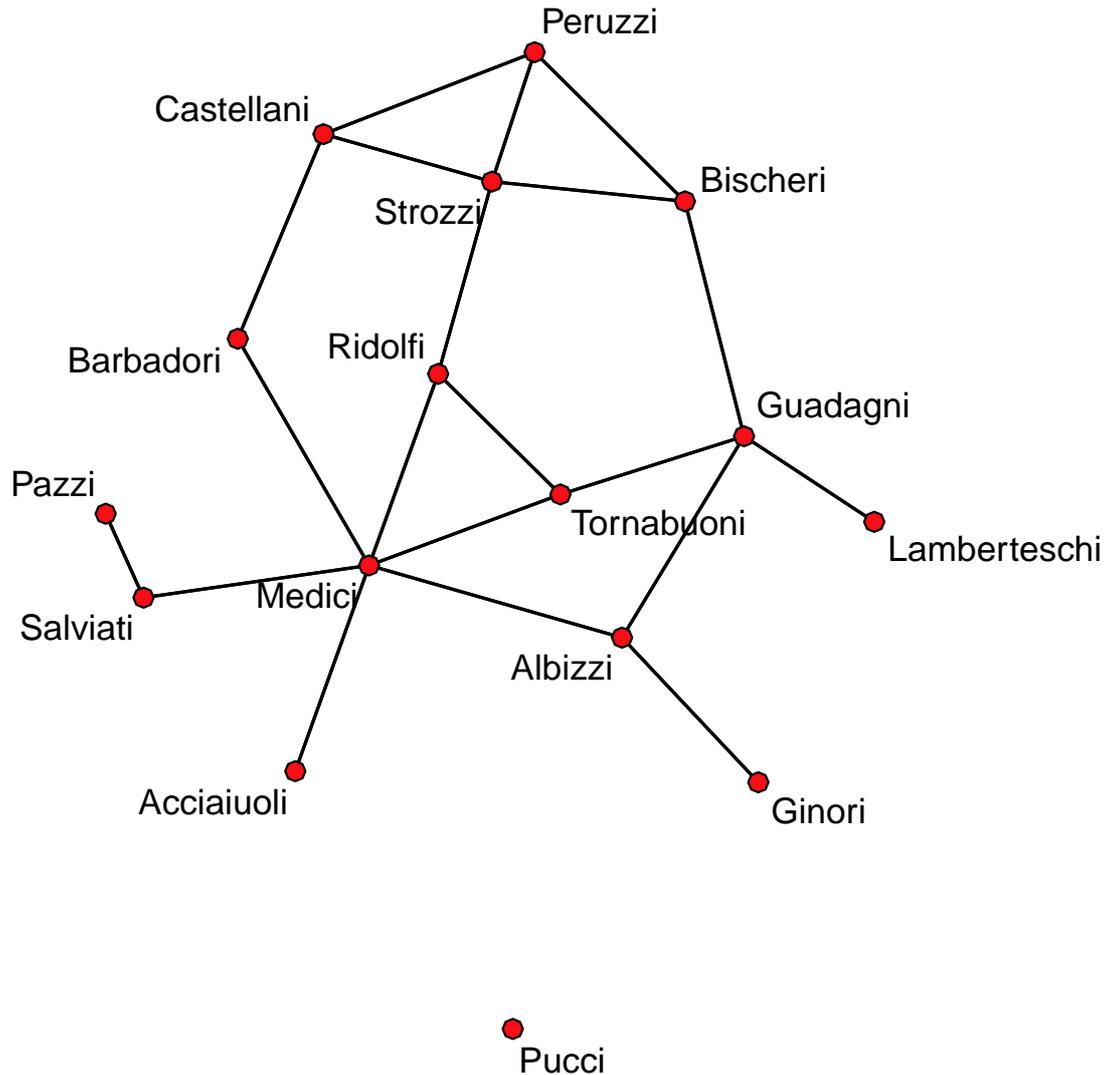


図3 フィレンツェ貴族間の婚姻関係ネットワーク

た状態にしていることが分かります。また、メディチ家は次数が最も多く、6家と婚姻関係を結んでいることも見て取れます。さすがフィレンツェ最大の貴族ですね。

参考までに、以下のコマンドで3次元ネットワークを描いてみましょう。

```
# 3次元描画
gplot3d(flo, gmode="graph", mode="kamadakawai", displaylabels=T,
        label.col="black", vertex.col="green", vertex.alpha=0.3,
        edge.col="gray", edge.alpha=0.3)
```

上記のコマンドを実行してみれば分かることなのですが、ぶっちゃけ2次元ネットワーク以上の情報は得られません。と、言いますか、むしろ2次元描画よりも見にくいです。しかし、マウスでぐりぐり図を動かすのは楽しいですし、何より見栄えがなかなか良いのでプレゼンテーションなんかには使えるかもしれません。

3.4 ネットワーク構造の諸指標

では次に、ネットワーク構造に関する諸指標を計算してみましょう。とりあえず『密度』と『距離』を求めてみます。コマンドは以下です。

```
# 密度（無向グラフなので mode を graph にする）
gden(flo, mode="graph")
# 距離
geodist(flo)
```

密度に関しては、値は 0.167 となりました。低いこと以外、特筆すべきことはないでしょう。

また、貴族間の距離に関しては以下のようにになりました。

```
$counts
  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16]
[1,]   1   1   1   3   1   1   2   2   1   1   2   0   1   1   1   1
[2,]   1   1   1   1   1   1   1   1   1   1   1   0   1   1   2   2
[3,]   1   1   1   2   1   1   2   2   1   1   1   0   1   1   1   1
[4,]   3   1   2   1   2   1   1   1   3   3   1   0   1   3   1   1
[5,]   1   1   1   2   1   1   2   2   1   1   1   0   1   1   1   2
[6,]   1   1   1   1   1   1   1   1   1   1   1   0   1   1   2   2
[7,]   2   1   2   1   2   1   1   1   2   2   1   0   1   2   1   1
[8,]   2   1   2   1   2   1   1   1   2   2   1   0   1   2   1   1
[9,]   1   1   1   3   1   1   2   2   1   1   2   0   1   1   1   1
[10,]  1   1   1   3   1   1   2   2   1   1   2   0   1   1   1   1
[11,]  2   1   1   1   1   1   1   1   2   2   1   0   1   2   1   2
[12,]  0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0
[13,]  1   1   1   1   1   1   1   1   1   1   1   0   1   1   1   1
[14,]  1   1   1   3   1   1   2   2   1   1   2   0   1   1   1   1
[15,]  1   2   1   1   1   2   1   1   1   1   1   0   1   1   1   1
[16,]  1   2   1   1   2   2   1   1   1   1   2   0   1   1   1   1

$gdist
  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16]
[1,]   0   2   2   4   3   3   3   4   1   3   4   Inf   2   2   3   2
[2,]   2   0   2   2   3   1   1   2   1   3   3   Inf   2   2   3   2
[3,]   2   2   0   3   1   3   3   4   1   3   2   Inf   2   2   2   2
[4,]   4   2   3   0   2   3   1   2   3   5   1   Inf   2   4   1   2
[5,]   3   3   1   2   0   4   3   4   2   4   1   Inf   2   3   1   3
[6,]   3   1   3   3   4   0   2   3   2   4   4   Inf   3   3   4   3
[7,]   3   1   3   1   3   2   0   1   2   4   2   Inf   2   3   2   1
[8,]   4   2   4   2   4   3   1   0   3   5   3   Inf   3   4   3   2
[9,]   1   1   1   3   2   2   2   3   0   2   3   Inf   1   1   2   1
[10,]  3   3   3   5   4   4   4   5   2   0   5   Inf   3   1   4   3
[11,]  4   3   2   1   1   4   2   3   3   5   0   Inf   2   4   1   3
[12,]  Inf 0   Inf Inf Inf Inf
[13,]  2   2   2   2   2   3   2   3   1   3   2   Inf 0   2   1   1
[14,]  2   2   2   4   3   3   3   4   1   1   4   Inf 2   0   3   2
[15,]  3   3   2   1   1   4   2   3   2   4   1   Inf 1   3   0   2
[16,]  2   2   2   2   3   3   1   2   1   3   3   Inf 1   2   2   0
```

[1] ~ [16] までの数字はそれぞれ表 1 の貴族名を表しています。まず最初に目につくのは、\$gdist における [12] の行がすべて Inf になっていることです。この意味するところは、このノードには無向辺が存在しない、言い換えると 12 番目の貴族はどこにも婚姻関係を結んでいないということを表しています。12 番目の貴族とはプッチ家のことでした。その通りになっていることを図 3 を見て確認して下さい。

また、気になるのは当時の有力貴族であったメディチ家（9番）のことなのですが、ここでは距離が3の最も遠い位置にいるビスケリ家（4番）、ランバーテスキ家（8番）、ペルッツィ家（11番）について考察してみましょう。まずはビスケリ家ですが、この家は元々は有力貴族だったにもかかわらず、火事を起こし、財産は一夜にして灰になりました。こうして経済的に大打撃を受けたビスケリ家はフィレンツェを去ったそうです。同じくペルッツィ家も、フィレンツェ最古の家系の1つであり、もともとは五家の一家であることは疑う余地のないほど有力な金融業者だったそうです。しかし、100年戦争の初期イギリスに多額の金の貸し付けを行い、その後イングランド王エドワード3世にこの戦争ローンを踏み倒され、破産したそうです。また、ランバーテスキ家に関しては詳細が分からないほどマイナーな貴族です。メディチ家の台頭は、これら古い家系の没落があった後の話です。距離が遠いのも仕方のないことなのですね。なお、ビスケリ家はフィレンツェを去った後、しばらくしてからグアダーニ家（7番）と名前を変えてフィレンツェに戻ってきたそうです。メディチ家とグアダーニ家の間の距離を見てみると2となっており、権力者に近づいています。なかなか『したたか』ではありませんか。

3.5 ブロックモデリング

次に構造同値の関係から、ネットワークを縮約できないかを見るために、ブロックモデリングを実行してみましょう。まずは隣接行列のクラスター分析からです。

```
clust <- equiv.clust(flo, method="euclidean", mode="graph",
                    cluster.method="ward")
plot(clust)
```

上記のようにユークリッド距離関数、ウォード法を用いてクラスター分析を行うと図4のようになります。しかし、ノードを表すのが番号のままでは少し見にくいですね。そこで以下のように貴族名のベクトルを作成し、それを `plot()` 関数の引数に食わせてみましょう。

```
# ラベルとする貴族名のオブジェクトを作成
fname <- c("Acciaiuoli", "Albizzi", "Barbadori", "Bischeri",
           "Castellani", "Ginori", "Guadagni", "Lamberteschi",
           "Medici", "Pazzi", "Peruzzi", "Pucci", "Ridolfi",
           "Salviati", "Strozzi", "Tornabuoni")
# クラスター分析の結果をラベル付きでプロット
plot(clust, labels=fname)
```

上記の結果を図5に示します。デンドログラムの下に貴族名が表示されており、図4よりも見やすいですね。

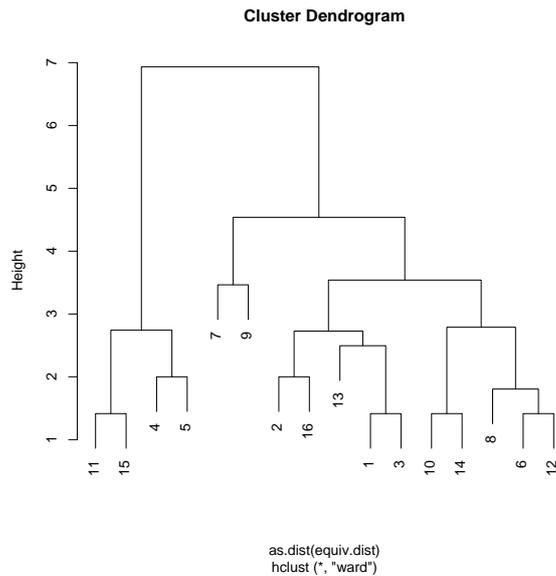


図4 ラベルなしデンドログラム

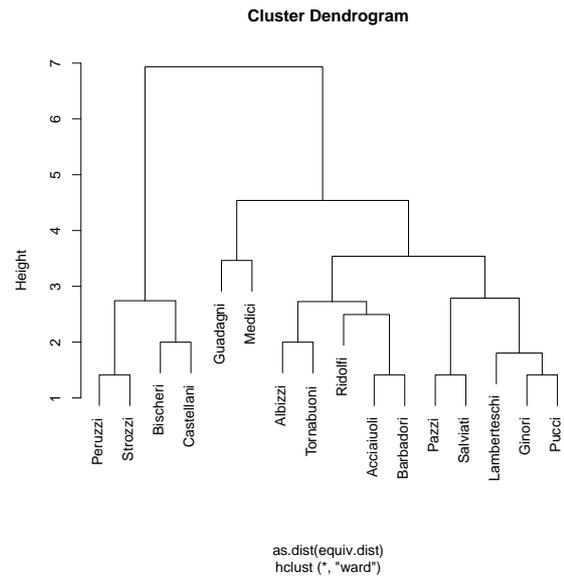


図5 ラベルありデンドログラム

さて、この結果を用いてブロックモデリングを行います。結果は以下の通りです。なお、資料に入りきらないため若干編集してあります。また【Block membership:】の並び順に関しても、見やすくするために変更してあります。

```
> blockmodel(flo, ec=clust, k=3, mode="graph")
```

Network Blockmodel:

Block membership:

Acciaiuoli	Albizzi	Barbadori	Ginori	Lamberteschi
1	1	1	1	1
Pazzi	Pucci	Ridolfi	Salviati	Tornabuoni
1	1	1	1	1
Bischeri	Castellani	Peruzzi	Strozzi	
2	2	2	2	
Guadagni	Medici			
3	3			

Reduced form blockmodel:

	Block 1	Block 2	Block 3
Block 1	0.06666667	0.0500000	0.450
Block 2	0.05000000	0.8333333	0.125
Block 3	0.45000000	0.1250000	0.000

まず【Block membership:】を見てみると、Block3にはメディチ家とグアダーニ家が分類されています。先ほども述べたように、グアダーニ家はビスケリ家がフィレンツェを

追い出され、再度戻ってくるときに名乗った名前です。このようにメディチ家と同じ役割のブロックに所属していることから分かるようにグアダーニ家（旧ビスケリ家）の処世のうまさが見て取れます。

また Block2 には、破産した家であるビスケリ家とペルツィ家が分類されています。このことから推察するに、Block2 はメディチ家が台頭する前に有力だった家、あるいはメディチ家とは相容れない家の集まりなのでしょう。そして Block1 には権力者メディチ家と上手く結びつくことができた家が分類されているというわけです。

このことは【Reduced form blockmodel:】を見ても明らかです。Block3（権力者）と関連が深いのは Block1 であり（値は 0.45）、Block2（没落）との値は 0.125 と小さくなっています。なかなか興味深い結果となっていますね。