A User's Guide to Network Analysis in R

Author:Douglas A. Luke

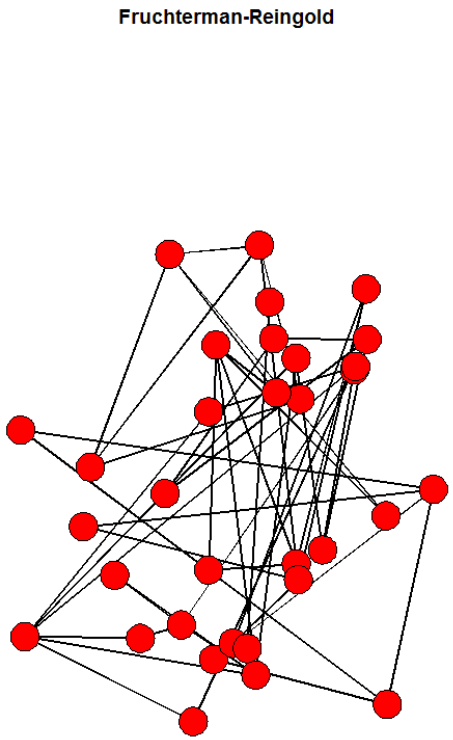
Reporter:萧伯任

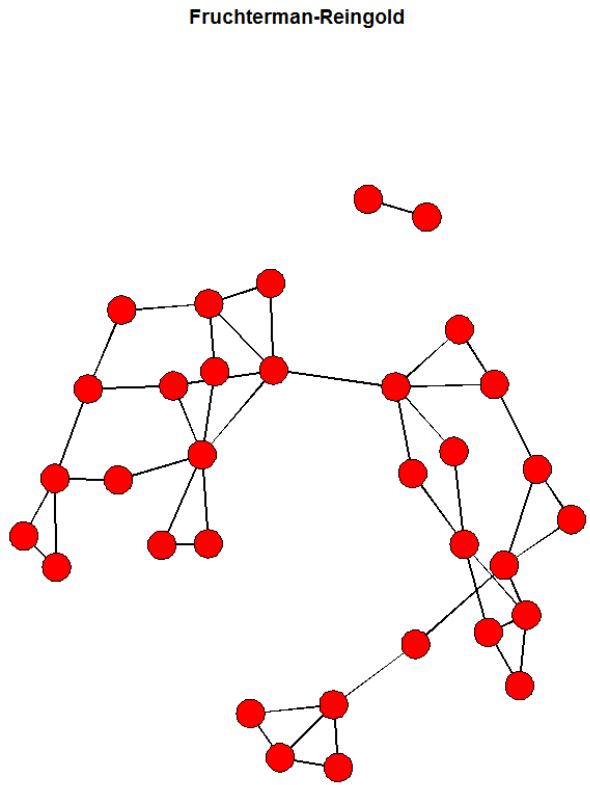
2016-11-9

Contents

1. Chapter 4 Basic Network Plotting and Layout
   - 4.1 The Challenge of Network Visualization
   - 4.2 The Aesthetics of Network Layouts
   - 4.3 Basic Plotting Algorithms and Methods

4.2 The Aesthetics of Network Layouts





Network graphics are easier to understand if they follow the following five guidelines: (3m,2M)

1. Minimize edge crossings.
2. Maximize the symmetry of the layout of nodes.
3. Minimize the variability of the edge lengths.
4. Maximize the angle between edges when they cross or join nodes.
5. Minimize the total space used for the network display.

One general class of algorithms, called force-directed, has proven to be a flexible and powerful approach to automatic network layouts. These algorithms work iteratively to minimize the total energy in a network

- Fruchterman-Reingold algorithm ' s layout

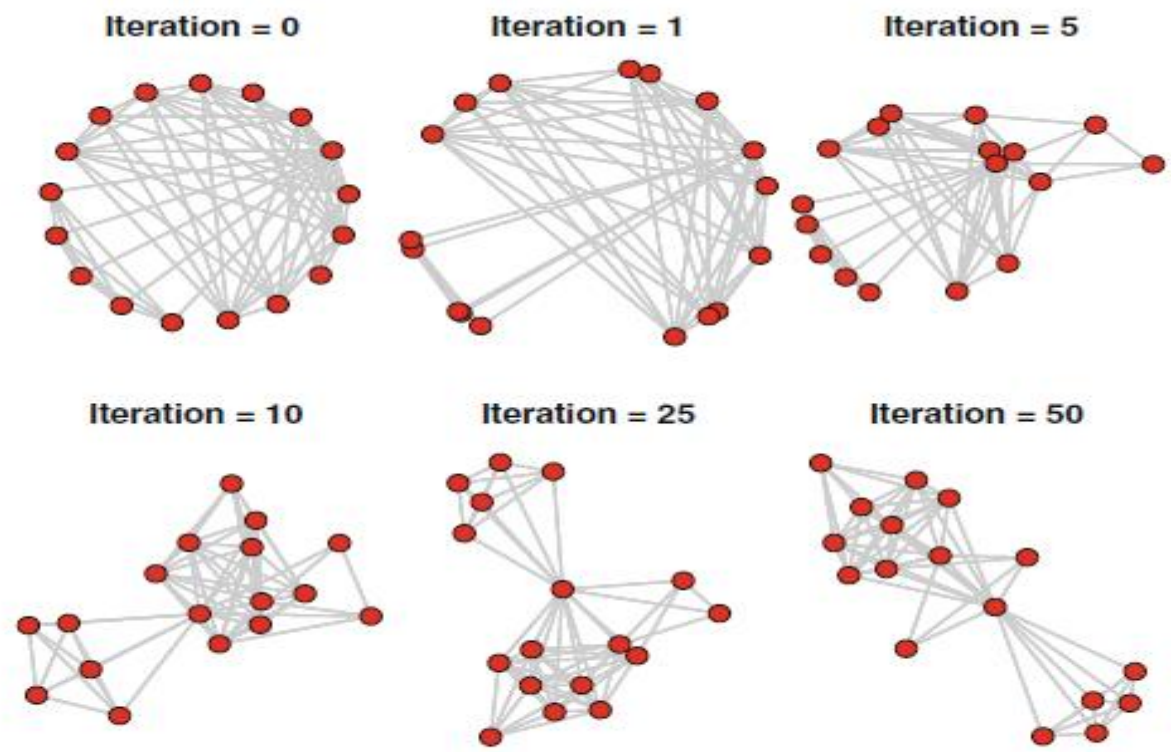
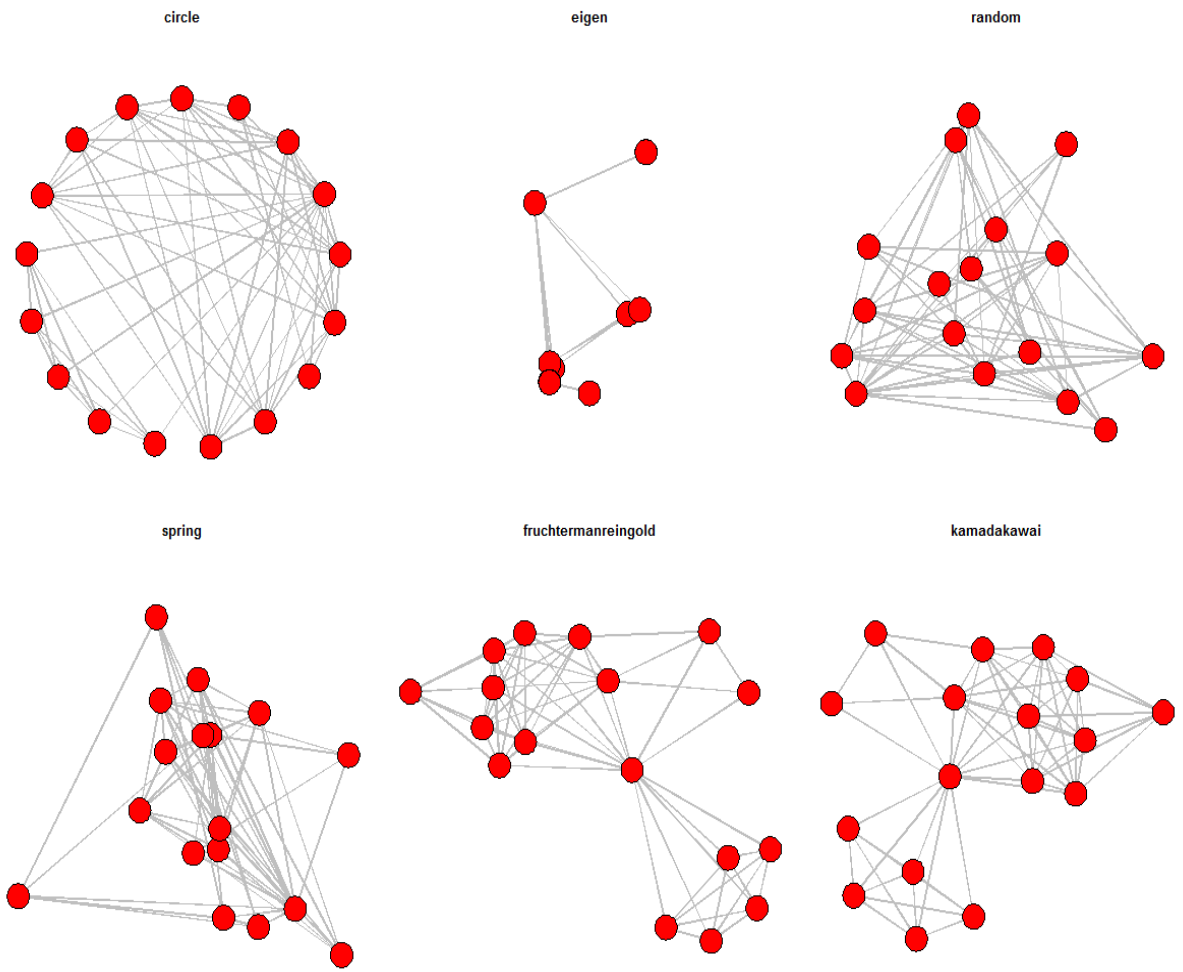


Fig. 4.4 Iterative Fruchterman-Reingold algorithm

4.3 Basic Plotting Algorithms and Methods

```
as.sociomatrix(Bali)
op <- par(mar=c(0,0,4,0),mfrow=c(2,3))
gplot(Bali,gmode="graph",edge.col="grey75",
      vertex.cex=1.5,mode='circle',main="circle")
gplot(Bali,gmode="graph",edge.col="grey75",
      vertex.cex=1.5,mode='eigen',main="eigen")
gplot(Bali,gmode="graph",edge.col="grey75",
      vertex.cex=1.5,mode='random',main="random")
gplot(Bali,gmode="graph",edge.col="grey75",
      vertex.cex=1.5,mode='spring',main="spring")
gplot(Bali,gmode="graph",edge.col="grey75",
      vertex.cex=1.5,mode='fruchtermanreingold',main='fruchtermanreingold')
gplot(Bali,gmode="graph",edge.col="grey75",
      vertex.cex=1.5,mode='kamadakawai',
      main='kamadakawai')
par(op)
```


circle
eigen
random
spring
fruchtermanreingold
kamadakawai

4.3.1 Finer Control Over Network Layout

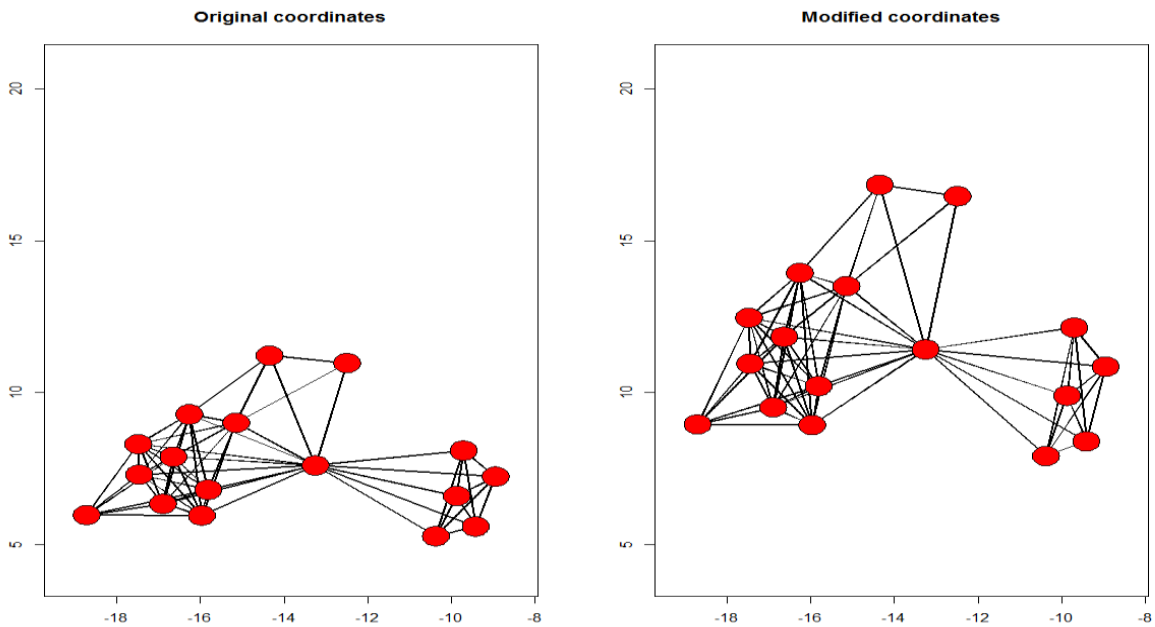
```
mycoords1 <- gplot(Bali,gmode="graph",
                   vertex.cex=1.5)
mycoords2 <- mycoords1
mycoords2[,2] <- mycoords1[,2]*1.5

mycoords1
mycoords2
```

```
> mycoords1
                x          y
 [1,] -0.4230975  0.3163096
 [2,] -0.5066442 -2.6028000
 [3,] -1.0966784  2.8334580
 [4,]  1.6666642  0.4618159
 [5,] -1.8973319  2.2458946
 [6,] -0.9905590 -0.3751369
 [7,]  1.0651642 -3.1226218
 [8,] -2.1040211  0.7644136
 [9,] -0.1653383  2.1003249
[10,]  4.5656928 -0.8711486
[11,]  5.5654128 -0.8870595
[12,]  5.7262530  0.2976781
[13,]  4.7871597  1.7959726
[14,]  5.7357085  1.4324933
[15,] -0.2364330  3.0832271
[16,] -2.8288392  3.8177842
[17,] -2.3625703  1.4749702
```

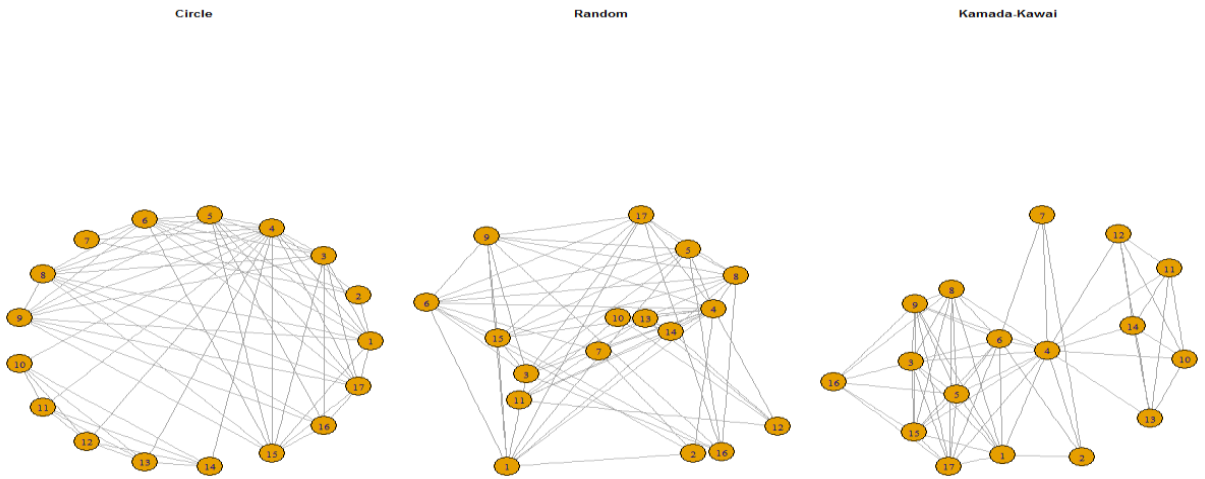
```
> mycoords2
                x          y
 [1,] -0.4230975  0.4744644
 [2,] -0.5066442 -3.9041999
 [3,] -1.0966784  4.2501869
 [4,]  1.6666642  0.6927238
 [5,] -1.8973319  3.3688420
 [6,] -0.9905590 -0.5627054
 [7,]  1.0651642 -4.6839327
 [8,] -2.1040211  1.1466204
 [9,] -0.1653383  3.1504873
[10,]  4.5656928 -1.3067228
[11,]  5.5654128 -1.3305892
[12,]  5.7262530  0.4465172
[13,]  4.7871597  2.6939589
[14,]  5.7357085  2.1487399
[15,] -0.2364330  4.6248406
[16,] -2.8288392  5.7266762
[17,] -2.3625703  2.2124552
```

```
op <- par(mar=c(4,3,4,3),mfrow=c(1,2))
gplot(Bali,gmode="graph",coord=mycoords1,
      vertex.cex=1.5,suppress.axes = FALSE,
      ylim=c(min(mycoords2[,2])-1,max(mycoords2[,2])+1),
      main="Original coordinates")
gplot(Bali,gmode="graph",coord=mycoords2,
      vertex.cex=1.5,suppress.axes = FALSE,
      ylim=c(min(mycoords2[,2])-1,max(mycoords2[,2])+1),
      main="Modified coordinates")
par(op)
```



4.3.2 Network Graph Layouts Using igraph

```
library(igraph)
library(intergraph)
iBali <- asIgraph(Bali)
op <- par(mar=c(0,0,3,0),mfrow=c(1,3))
plot(iBali,layout=layout_in_circle,
     main="Circle")
plot(iBali,layout=layout_randomly,
     main="Random")
plot(iBali,layout=layout_with_kk,
     main="Kamada-Kawai")
par(op)
```



Thank You for your Attention