

L24 LAB 10 (MD)

AnaGSanjuanM

2023-02-22

Hecho con gusto por Carla Carolina Pérez Hernández (UAEH)

T1_04_REDES COMPLEJAS(1) - Visión de redes complejas - parte 1

Para dibujar el espacio-producto

Hace uso de datos complejos (matrices muy amplias con más de 100 nodos)

Objetivo: Estimar el Maximum Spanning Tree -árbol de expansión máxima- (asegurar una visión clara del espacio-producto).

Red troncal: Estructura general de la red: vamos a poder v. redes complejas

REGLAS DE ACCIÓN

Regla 1: Mantener n-1 conexiones como máximo, es decir, si tenemos 200 nodos, las conexiones serán 199.

Regla 2: Quitar las conexiones con el peso más bajo, nos vamos quedar con las del peso máximo (menos conexiones) por lo que se limpiarán los nodos, dando prioridad a los nodos con muchas más conexiones (aristas adyacentes al nodo)

Regla 3: No crear nodos aislados

En este ejercicio vamos a:

1. Usar un matriz hipotetica de datos
2. Graficar sus próximos adyacentes

LABORATORIO 10

Alumna: Ana Grisel Sanjuan Merida

Visión clara del espacio-producto: CASO HIPOTÉTICO

¿Cómo crear una visión clara del espacio-producto: árbol de expansión máxima (MST)

Paquete: Balland, P.A. (2017) Economic Geography in R: Introduction to the EconGeo Package, Paper in Evolution

1)La visualización del espacio - producto sea una red conectada: evitar islas de productos aislados.

2)PROBLEMA: tratar de visualizar demasiados enlaces puede crear una complejidad visual innecesaria donde se obstruirán las conexiones más relevantes.

Calculamos el árbol de expansión máxima (MST) de la matriz de proximidad. MST es el conjunto de enlaces que conecta todos los nodos de la red utilizando un número mínimo de conexiones y la suma máxima posible de proximidades. Calculamos el MST usando el algoritmo de Kruskal: Básicamente, el algoritmo clasifica los valores de la matriz de proximidad en orden descendente y luego incluye enlaces en el MST si y solo si conectan un producto aislado. Por definición, el MST incluye todos los productos, pero el número de enlaces es el mínimo posible. Después de seleccionar los enlaces utilizando los criterios mencionados anteriormente, construimos una visualización utilizando un algoritmo de diseño dirigido por la fuerza.

Crear una matriz aleatoria de 200*200.Es una matriz con datos aleatorios conformada por 200 columnas y 200 renglones.

```
M <- matrix(runif(200*200, min=0, max=100), ncol=200)
diag(M) <- 0
head (M[,1:6])
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,]  0.00000 12.557413 57.83207  9.7919999 16.32347 13.3240423
## [2,] 89.59532  0.000000 11.96606  0.1953115 47.28787 56.9845792
## [3,] 32.18517 76.305115  0.00000 64.8215276 67.89941 58.4324453
## [4,] 11.19717 20.723567 68.33390  0.0000000 97.02925  0.1487969
## [5,] 32.50747 62.133908 96.44509 74.9388888  0.00000  2.7714915
## [6,] 94.60943  4.990183 30.22748 31.3301105 62.33856  0.0000000
```

```
dim (M)
```

```
## [1] 200 200
```

Para graficar

```
library (igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##   decompose, spectrum
```

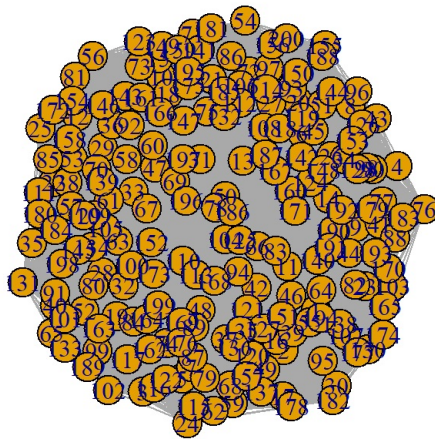
```
## The following object is masked from 'package:base':
##
##   union
```

Graficar la matriz de adyacencias. Crear objeto llamado g, que es la gráfica de adyacencia. El input es la matriz que se acaba de generar. Es una red no dirigida y el peso es una variable a tomar en cuenta.

```
g <- graph.adjacency(M, mode="undirected" , weighted = TRUE)
```

El plot tiene muchas aristas y es difícil de interpretar

```
plot (g)
```



Aplicación de la función Árbol de Expansión Máxima. Para asegurar que las tres reglas se cumplan, se multiplica la matriz original por -1 para hacerla negativa e identificar los máximos. Se aprecia en el Environment.

```
M <- -M
head (M[,1:6])
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,]  0.00000 -12.557413 -57.83207  -9.7919999 -16.32347 -13.3240423
## [2,] -89.59532  0.000000 -11.96606  -0.1953115 -47.28787 -56.9845792
## [3,] -32.18517 -76.305115  0.00000 -64.8215276 -67.89941 -58.4324453
## [4,] -11.19717 -20.723567 -68.33390  0.0000000 -97.02925  -0.1487969
## [5,] -32.50747 -62.133908 -96.44509  -74.9388888  0.00000  -2.7714915
## [6,] -94.60943  -4.990183 -30.22748 -31.3301105 -62.33856  0.0000000
```

En la paquetería de igraph se tiene la función de calcular el árbol de expansión mínima, por lo que hacerla negativa me permite obtener el Árbol de Expansión Máxima. Transforma la matriz en en NEGATIVA y para identificar los máximos.

Graficar nueva con matriz negativa, ver el espacio-producto, tecnológico.

```
g <- graph.adjacency(M, mode="undirected" , weighted = TRUE)
```

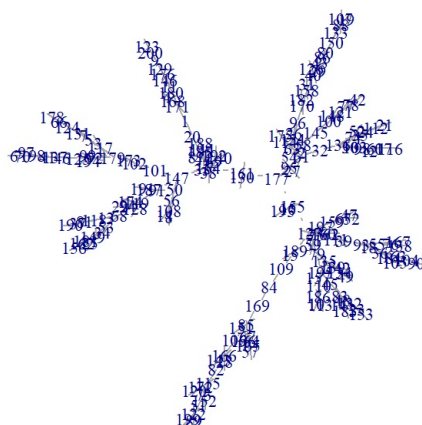
Calcular árbol de expansión mínima

```
MST <- minimum.spanning.tree(g)
```

Graficar lo que se acaba de hacer

```
plot (MST, vertex.shape="none", vertex.label.cex=.7)
```

```
## Warning in v(graph): Non-positive edge weight found, ignoring all weights
## during graph layout.
```



Matriz de adyacencias (nuevo relacionamiento)

```
A <- get.adjacency(MST, sparse = F)
```

Ver las aristas que hay entre los nodos

head(A)

[illegible]

```
## [3,] 0 0 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 0 0 0 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 1 0 0 0 0
##      [,75] [,76] [,77] [,78] [,79] [,80] [,81] [,82] [,83] [,84] [,85] [,86]
## [1,] 0 0 0 0 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 0 0 0 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0 0 0
##      [,87] [,88] [,89] [,90] [,91] [,92] [,93] [,94] [,95] [,96] [,97] [,98]
## [1,] 0 0 0 0 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 0 0 0 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0 0 0
##      [,99] [,100] [,101] [,102] [,103] [,104] [,105] [,106] [,107] [,108]
## [1,] 0 0 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 0 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0
##      [,109] [,110] [,111] [,112] [,113] [,114] [,115] [,116] [,117] [,118]
## [1,] 0 0 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 1 0
## [4,] 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 0 0 0 0 0 0 1 0
## [6,] 0 0 0 0 0 0 0 0 0 0
##      [,119] [,120] [,121] [,122] [,123] [,124] [,125] [,126] [,127] [,128]
## [1,] 0 0 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 0 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0
##      [,129] [,130] [,131] [,132] [,133] [,134] [,135] [,136] [,137] [,138]
## [1,] 0 0 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 1 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0
##      [,139] [,140] [,141] [,142] [,143] [,144] [,145] [,146] [,147] [,148]
## [1,] 0 0 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 0 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0
##      [,149] [,150] [,151] [,152] [,153] [,154] [,155] [,156] [,157] [,158]
## [1,] 0 0 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 0 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0
##      [,159] [,160] [,161] [,162] [,163] [,164] [,165] [,166] [,167] [,168]
## [1,] 0 0 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 1
## [5,] 0 0 0 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0
##      [,169] [,170] [,171] [,172] [,173] [,174] [,175] [,176] [,177] [,178]
## [1,] 0 0 1 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 0 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0
##      [,179] [,180] [,181] [,182] [,183] [,184] [,185] [,186] [,187] [,188]
## [1,] 0 0 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0
```

```
## [5,]      0      0      0      0      0      0      0      0      0      0
## [6,]      0      0      0      0      0      0      0      0      0      0
##      [,189] [,190] [,191] [,192] [,193] [,194] [,195] [,196] [,197] [,198]
## [1,]      0      0      0      0      0      0      0      0      0      0
## [2,]      0      0      0      0      0      0      0      0      0      0
## [3,]      0      0      0      0      0      0      0      0      0      0
## [4,]      0      0      0      0      0      0      0      0      0      0
## [5,]      0      0      0      0      0      0      0      0      0      0
## [6,]      0      0      0      0      0      0      0      0      0      0
##      [,199] [,200]
## [1,]      0      0
## [2,]      0      0
## [3,]      0      0
## [4,]      0      0
## [5,]      0      0
## [6,]      0      0
```

Exportar

Ver las aristas (pesos). nodos n-1

```
write.graph(MST,file="g.gml", format="gml")
```

Formato legible para cualquier paquetería especializada. Se guarda en la carpeta específica.

Exportar resultados para cytoscape o gephi.

```
write.csv(A, file="Adyacentes.csv")
```