

団代数と置換

団代数の変異同値な簇集合には、頂点置換簇が含まれる

簇に付随する団代数を考える。

団代数の変異により、簇も変異する。

団代数の変異によって、簇のトポロジーは同じだが、頂点のIDが置換したものが生じることが知られている。

従って、簇の変異同値の交換関係には、頂点ID置換体が含まれることがわかる。

ただし、簇の1頂点に関する変異を行っても、置換は実現できないから、置換は簇変異を繰り返して得られるものであるはず。

団代数の置換行列変異の線形代数表現と行・列置換の線形代数表現の類似性

置換

正方行列 X の行と列との置換は、置換行列 P を用いて

$$PXP^T$$

と表される。

団代数変異

<https://arxiv.org/abs/1703.02225> (<https://arxiv.org/abs/1703.02225>)

団代数の $n \times n$ exchange matrix(skew-symmetric) B に対して、次のような正方行列 W_k (k 番による変異のための行列) を定める

$$W_k = \begin{pmatrix} I_{k-1} & \alpha & \mathbf{0} \\ \mathbf{0} & -1 & \mathbf{0} \\ \mathbf{0} & \beta & I_{n-k} \end{pmatrix}$$

ただし、 I_a は $a \times a$ 単位行列、 $\alpha = (\max(b_{1,k}, 0), \max(b_{2,k}, 0), \dots, \max(b_{k-1,k}, 0))$,
 $\beta = (\max(b_{k+1,k}, 0), \dots, \max(b_{n,k}, 0))$

このとき、変異後の置換行列は

$$W_k B W_k^T$$

となる。

よく似ている。。。

一応、確認しておく。

```

my.B.mut <- function(B, k) {
  new.B <- B
  n <- length(B[, 1])
  new.B[k, ] <- - B[k, ]
  new.B[, k] <- - B[, k]
  for(i in 1:n){
    for(j in 1:n){
      if(i != k){
        if(j != k){
          new.B[i, j] <- B[i, j] + sign(B[i, k]) * max(B[i, k]*B[k, j], 0)
        }
      }
    }
  }
  return(new.B)
}

d <- 5
B <- matrix(sample(1:9, d^2, replace=TRUE), d, d)
B <- B - t(B)

```

B

```

##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    0   -1    2    6   -7
## [2,]    1    0    4    6    0
## [3,]   -2   -4    0   -4    1
## [4,]   -6   -6    4    0    4
## [5,]    7    0   -1   -4    0

```

```

my.mat.mut.B <- function(B, k) {
  b <- B[, k]
  b. <- cbind(b, rep(0, length(b)))
  b. <- apply(b., 1, max)
  b.[k] <- -1
  ret <- diag(rep(1, length(b)))
  ret[, k] <- b.
  return(ret)
}

```

```

W <- my.mat.mut.B(B, 2)
W %*% B %*% t(W)

```

```

##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    0    1    2    6   -7
## [2,]   -1    0   -4   -6    0
## [3,]   -2    4    0   -4    1
## [4,]   -6    6    4    0    4
## [5,]    7    0   -1   -4    0

```

```
my.B.mut(B, 2)
```

```
##      [,1] [,2] [,3] [,4] [,5]  
## [1,]    0    1    2    6   -7  
## [2,]   -1    0   -4   -6    0  
## [3,]   -2    4    0   -4    1  
## [4,]   -6    6    4    0    4  
## [5,]    7    0   -1   -4    0
```

互換と置換

置換は互換の繰り返しで実現できる。

従って、以下では、置換と互換との関係について考えることとする。

置換と互換

順列の互換

順列は鎖状の置換とみなすことができる(A_n 型ルート系ともみなせる)。

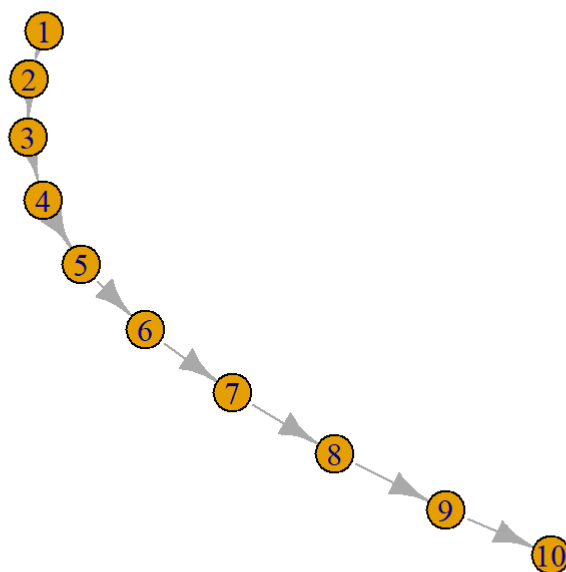
```
library(igraph)
```

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

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

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

```
gs <- list()  
  
d <- 10  
A1 <- diag(rep(1, d))  
  
A1 <- A1[c(2:d, 1), ]  
A1[d, 1] <- 0  
  
B1 <- A1 - t(A1)  
  
gs[[1]] <- graph.adjacency(A1)  
plot(gs[[1]])
```



隣接 2 頂点 v_k, v_{k+1} に関する変異 μ_k, μ_{k+1} の交互 5 連続変異

天下り式になるが、鎖状簇の歪対称行列 B に対して以下の変異を行うことで

$$B_M = \mu_k \mu_{k+1} \mu_k \mu_{k+1} \mu_k B$$

$$M = (\mu_k, \mu_{k+1}, \mu_k, \mu_{k+1}, \mu_k)$$

B_M は、 B の行と列とについて、 $k, k+1$ を交換したものになっている。

$$B_M = s_{k,k+1}(B)$$

```

As <- list()
As[[1]] <- A1
Bs <- list()
Bs[[1]] <- B1
Ws <- list()

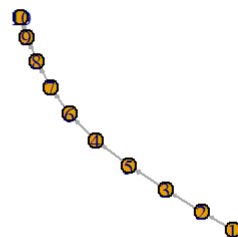
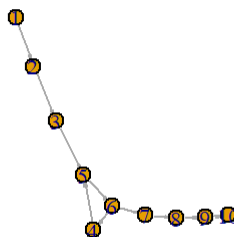
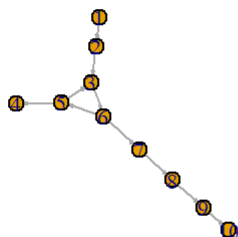
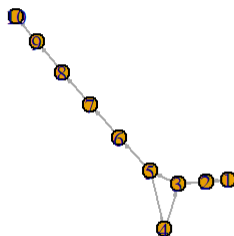
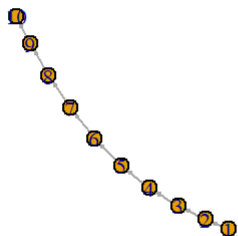
k <- 4
n.mut <- 5
ks <- c(k, k+1, k, k+1, k)

for(i in 1:n.mut){
  k <- ks[i]
  B1 <- my.B.mut(B1, k)
  Ws[[i]] <- my.mat.mut.B(B1, k)
  A1 <- B1
  A1[which(A1 < 0)] <- 0
  gs[[i+1]] <- graph.adjacency(A1)
  As[[i+1]] <- A1
  Bs[[i+1]] <- B1
}

par(mfrow=c(2, 3))

for(i in 1:(n.mut+1)){
  plot(gs[[i]], edge.arrow.size=0.1)
}

```



B を直接、行置換・列置換した行列が、変異を5回繰り返して作成した行列と一致することの検算。

```
P <- diag(rep(1, d))
s <- 1:d
s[k+1] <- k
s[k] <- k+1
P <- P[s,]

P %*% Bs[[1]] %*% t(P) - Bs[[6]]
```

```
##      [, 1] [, 2] [, 3] [, 4] [, 5] [, 6] [, 7] [, 8] [, 9] [, 10]
## [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
## [7,]    0    0    0    0    0    0    0    0    0    0
## [8,]    0    0    0    0    0    0    0    0    0    0
## [9,]    0    0    0    0    0    0    0    0    0    0
## [10,]   0    0    0    0    0    0    0    0    0    0
```

```
range(P %*% Bs[[1]] %*% t(P) - Bs[[6]])
```

```
## [1] 0 0
```

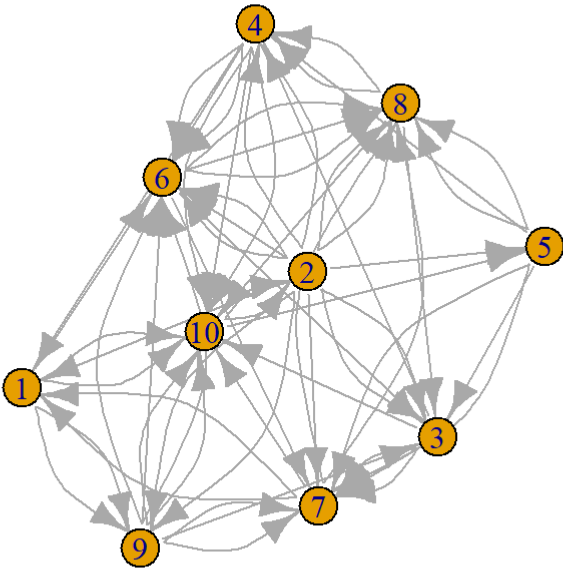
一般化

籠において、2 頂点が 1 本の矢で結ばれているとき、その 2 頂点の互換は、鎖状籠の場合と同じになる(が 2 頂点間に 2 本以上の矢があるときは、互換とならない)。

```
B <- matrix(sample(c(0, 1, 2, 3), d^2, replace=TRUE), d, d)
#B <- matrix(sample(c(0, 2), d^2, replace=TRUE), d, d)
#B <- matrix(sample(c(0, 1), d^2, replace=TRUE), d, d)
B <- B - t(B)
B1 <- B
A1 <- B1
A1[which(A1 < 0)] <- 0

gs[[1]] <- graph.adjacency(A1)
As <- list()
As[[1]] <- A1
Bs <- list()
Bs[[1]] <- B1
Ws <- list()

plot(gs[[1]])
```



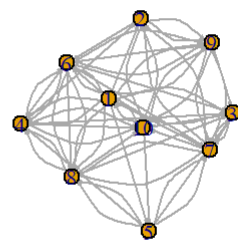
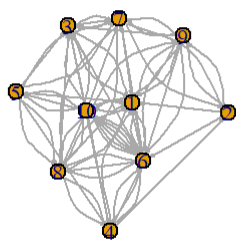
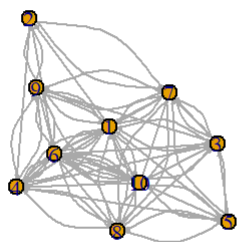
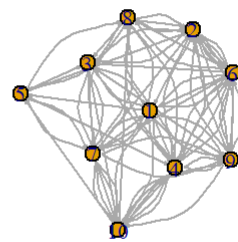
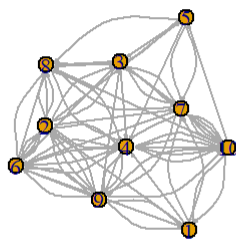
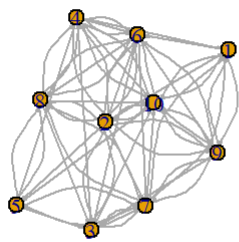
```
add <- which(Bs[[1]] == 1, arr.ind=TRUE)
k <- add[1, 2]
k2 <- add[1, 1]

#tmp <- sample(1:d, 2)
#k <- tmp[1]
#k2 <- tmp[2]
#k <- 4
#k2 <- 5
n.mut <- 5
ks <- c(k, k2, k, k2, k)

#n.mut <- 9*100
#ks <- sample(1:d, n.mut, replace=TRUE)
for(i in 1:n.mut){
  k <- ks[i]
  B1 <- my.B.mut(B1, k)
  Ws[[i]] <- my.mat.mut.B(B1, k)
  A1 <- B1
  A1[which(A1 < 0)] <- 0
  gs[[i+1]] <- graph.adjacency(A1)
  As[[i+1]] <- A1
  Bs[[i+1]] <- B1
}

par(mfrow=c(2, 3))

for(i in 1:(n.mut+1)){
  plot(gs[[i]], edge.arrow.size=0.1)
}
```

```
P <- diag(rep(1,d))
s <- 1:d
s[k2] <- k
s[k] <- k2
P <- P[s,]
```

```
P %*% Bs[[1]] %*% t(P) - Bs[[6]]
```

```
##      [, 1] [, 2] [, 3] [, 4] [, 5] [, 6] [, 7] [, 8] [, 9] [, 10]
## [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
## [7,]    0    0    0    0    0    0    0    0    0    0
## [8,]    0    0    0    0    0    0    0    0    0    0
## [9,]    0    0    0    0    0    0    0    0    0    0
## [10,]   0    0    0    0    0    0    0    0    0    0
```

```
range(P %*% Bs[[1]] %*% t(P) - Bs[[6]])
```

```
## [1] 0 0
```

考察

- 順列と置換、互換は基本的なデータ構造操作である
- 順列を鎖状態とみなすと、順列は叢・団代数の部分集合とみなされ、互換は基本的な操作ではなく、複数のステップを経て達成される複雑な処理であるとみなすこともできる
- 順列叢の変異により、異なる順列叢を作成する変異列には、有向木(半順序)が現れたり、三角形サイクルを含む有向グラフが現れたりする
- この構成に、順序・半順序・サイクルありの有向グラフの間のトポロジーが見いだせないだろうか？