SPHARM2

SPHARM方式で解くための準備

以下を解く。 \[C = (Z^T Z)^{-1} Z^T V \] Vは頂点数ｘ３の行列で、各頂点の(x,y,z)座標。 Zは、各頂点を単位球面にマップしたときの、球面上の点における、球面調和関数の値。行数＝頂点数、列数＝球面調和関数の数。 Cは、推定される球面調和関数係数行列。行数＝球面調和関数の数、列数＝３。

必要とするパッケージ

```
## Warning: package 'rgl' was built under R version 3.4.4
```

```
## Warning: package 'RFOC' was built under R version 3.4.4
```

```
## Warning: package 'igraph' was built under R version 3.4.4
```

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

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

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

```
## Warning: package 'knitr' was built under R version 3.4.4
```

```
## Warning: package 'tagcloud' was built under R version 3.4.4
```

```
## Loading required package: Rcpp
```

```
## Warning: package 'Rcpp' was built under R version 3.4.4
```

```
## Warning: package 'e1071' was built under R version 3.4.4
```

```
## Warning: package 'sets' was built under R version 3.4.3
```

```
##
## Attaching package: 'sets'
```

```
## The following object is masked from 'package:igraph':
##
##     %>%
```

```
## The following object is masked from 'package:rgl':
##
##     %>%
```

```
## Loading required package: polynom
```

```
knit_hooks$set(webgl = hook_webgl)
```

\(C = (Z^T Z)^{-1} Z^T V\)を解くための関数

球面調和関数の扱い

関数 sph() のthetaは、z軸の角度、phiはxy平面の角度

SPHARMの係数推定のパフォーマンス実験計画

観察データの条件を変えて、SPHARMの係数推定法(のR実装)のパフォーマンスを評価することとする。

実験条件

実験ID	形作成次数	頂点数	乱雑項	観察偏り	観察不均一
1	低	多	0	0	0
2	低	多	1	0	0
3	低	多	0	1	0
4	低	多	0	0	1
5	低	多	1	1	1
6	低	少	0	0	0
7	低	少	1	0	0
8	低	少	0	1	0
9	低	少	0	0	1
10	低	少	1	1	1

- (条件１)比較的低次の球面調和関数が滑らかな形を決めている場合。頂点数は多く、単位球面上の頂点分布は均一な場合。ただし、この形が球面調和関数で決まる、というのは、「球面上の点の半径を球面調和関数で長短に変えた上で、アフィン変換で歪ませたような形であるので、球面に張り付いた(x,y,z)座標が球面調和関数で表されているというような単純なものではない。
- (条件２) 条件１であって、頂点座標に乱雑項を加えた場合
- (条件３) 条件１であるものの、観察点が単位球面上で特定の範囲に偏っている場合
- (条件４) 条件１であるものの、観察点が単位球面上で粗密がある場合
- (条件５) (条件２, ３, ４)を併せたもの
- (条件６－１０)は頂点数を多から少へ変更したもの

実験条件を指定してデータを作るための関数

まず、形の基本情報(次数・頂点数を指定)を作り、 それに対して、実験条件(乱雑項・観察点不均一)に沿って値を変えることとする。

```
# d 次数
# N 頂点数の多少を決めるパラメタ
# 帰り値
## X 3次元座標
## E エッジリスト
## tp 単位球面上の角座標
## g グラフオブジェクト(igraph)
## w エッジの長さ
my.cell.shape <- function(d,N){
  # 形の凹凸・複雑さをコントロールするパラメタ、n,k
  n <- d
  k <- 5
  # メッシュのノード数をコントロールするパラメタ
  n.mesh <- N # 色々試すなら、32くらいにしておくのが無難。送ったhtmlファイルはn.mesh=64
  # 形を球面調和関数係数ベクトルで指定する
  A. <- matrix(runif(n^2), n, n)
  A.[1, 1] <- k
  B <- matrix(rnorm(n^2), n, n)
  # 閉曲面オブジェクトを作る
  xxx <- my.spherical.harm.mesh(A = A., B = B, n = n.mesh)

  #xxx$v <- xxx$v + rnorm(length(xxx$v))*r

  g <- graph.edgelist(xxx$edge,directed=FALSE)
  vname <- paste("",1:length(V(g)),sep="")
  g <- set_vertex_attr(g,"name",value=vname)
  # edge lengths
  w <- sqrt(apply((xxx$v[xxx$edge[,1],]-xxx$v[xxx$edge[,2],])^2,1,sum))

  # thetas,phis
  tmp <- my_sphere_tri_mesh(n.mesh)
  xyz <- tmp$xyz
  tosp <- TOSPHERE(xyz[, 1], xyz[, 2], xyz[, 3])
  tp <- cbind(tosp[[1]]/360 * 2 * pi, tosp[[2]]/360 * 2 * pi)

  return(list(X = xxx$v,E = xxx$edge, angles = cbind(tp[,2],tp[,1]),g=g,w=w ))
}
# 上の出力を3Dプロットする関数
my.plot.shape <- function(shape){
  plot3d(shape$X)
  segments3d(shape$X[c(t(shape$E)), ])
}
```

実験条件

```
# r 乱雑項(0はなし、正の数がその程度)
# h1 偏り1(heterogeneity) (0-1の値。0はすべて使う。割合1-h1の点を特定の領域に集中させる)
# h1をh2!=0と組み合せて使うときはh1>0とすること
# h2 偏り2。観察点の粗密を入れる。0は特定の領域に集中させる
# h2が大きくなると、集中した場所を複数個所作る。h2<=1
# h2.fracは使う頂点の割合
# h2k 偏り2をコントロールする係数(デフォルト値を使うことを原則とする)
# 返り値
## X 乱雑項を加えた座標
## E エッジ
## obs.id 観察点のID
my.spharm.jikken <- function(shape,r=0,h1=0,h2=0,h2.frac=1,h2k=4){
  X <- shape$X + rnorm(length(shape$X))*r
  n <- length(X[,1])
  if(h2==0){
    obs.n <- ceiling(n * (1-h1))
    tmp <- c(1:n,1:n)
    s <- sample(1:n,1)
    obs.id <- tmp[(1:obs.n)+s]
  }else{
    obs.n <- ceiling(n *(1-h1))
    obs.n2 <- obs.n * 2
    tmp <- c(1:n,1:n,1:n,1:n)
    s <- sample(1:n,1)
    obs.id <- tmp[(1:obs.n2)+s]
    #plot(obs.id)
    tmpn <- length(obs.id)
    pr <- h2k*(1 + cos((1:tmpn)/tmpn * pi * (1+h2*100)))
    obs.id <- sample(obs.id,ceiling(n *h2.frac),prob=pr)
  }
  return(list(X=X,E=shape$E,angles=shape$angles,obs.id=obs.id))
}
# 上の実験観測点をハイライトしてプロットする関数
my.plot.ex <- function(ex,obscol=2,r=0.05){
  plot3d(ex$X)
  segments3d(ex$X[c(t(ex$E)), ])
  spheres3d(ex$X[ex$obs.id,],color=obscol,radius=r)
}
# 上の実験観測点の分布を、単位球面上での粗密でプロットする関数
my.plot.ex.sphere <- function(ex,obscol=2,r=0.05){
  z <- cos(ex$angles[,1])
  x <- sin(ex$angles[,1]) * cos(ex$angles[,2])
  y <- sin(ex$angles[,1]) * sin(ex$angles[,2])
  XX <- cbind(x,y,z)
  plot3d(XX)
  segments3d(XX[c(t(ex$E)), ])
  spheres3d(XX[ex$obs.id,],color=obscol,radius=r)
}
# オリジナルの形と復元した形とを併せてプロットする関数
# 球面調和関数係数も返す
my.plot.oriANDest <- function(ex,obscol=1,r=0.05,L=15){
  my.plot.ex(ex,obscol=obscol,r=r)
  #spheres3d(ex$X,color=2,radius)
  coef1 <- my.spcoef.est(ex$X[ex$obs.id,],ex$angles[ex$obs.id,],L)
  ex1X.est <- my.coef2xyz(coef1,ex$angles)
  spheres3d(Re(ex1X.est),color=2,radius=r)
  segments3d(Re(ex1X.est)[c(t(ex$E)), ],color=2)
```

```
  return(list(coef=coef1,Xest=ex1X.est))
}
```

実験

実験１－５

形作成

```
d <- 6
N <- 32
shape1 <- my.cell.shape(d,N)
```

```
my.plot.shape(shape1)
#plot3d(shape1$X)
#segments3d(shape1$X[c(t(shape1$E)), ])
```

実験別の座標と観察点

```
ex1 <- my.spharm.jikken(shape1,r=0,h1=0,h2=0)
ex2 <- my.spharm.jikken(shape1,r=0.05,h1=0,h2=0)
ex3 <- my.spharm.jikken(shape1,r=0,h1=0.2,h2=0)
ex4 <- my.spharm.jikken(shape1,r=0,h1=0,h2=0.2,h2.frac=0.8,h2k=10)
ex5 <- my.spharm.jikken(shape1,r=0.05,h1=0.3,h2=0.2,h2.frac=0.8,h2k=10)
```

```
my.plot.ex(ex1)
```

```
my.plot.ex.sphere(ex1)
```

```
my.plot.ex(ex2)
```

```
my.plot.ex.sphere(ex2)
```

```
my.plot.ex(ex3)
```

```
my.plot.ex.sphere(ex3)
```