

使用MDMR包进行多变量距离矩阵回归

Alex / 2023-08-03 / free_learner@163.com / AlexBrain.cn

记录使用R环境下MDMR包进行多变量距离矩阵回归（multivariate distance matrix regression, MDMR）的基本方法。这里的内容来源于MDMR包作者写的[教程](#)。多变量矩阵回归适用于当自变量和因变量都是多变量的情形，而我们常用的多元线性回归因变量要求是单变量（univariate）。关于MDMR的数学原理，我还不是特别清楚，初步理解就是检验因变量的距离与自变量的关系，具体的请参考：

McArtor, D. B., Lubke, G. H., & Bergeman, C. S. (2017). Extending multivariate distance matrix regression with an effect size measure and the asymptotic null distribution of the test statistic. *Psychometrika*, 82, 1052-1077.

MDMR在MRI脑成像数据分析中的应用可以参考下面的论文：

Shehzad, Z., Kelly, C., Reiss, P. T., Craddock, R. C., Emerson, J. W., McMahon, K., ... & Milham, M. P. (2014). A multivariate distance-based analytic framework for connectome-wide association studies. *Neuroimage*, 93, 74-94.

一、安装MDMR包

```
install.packages('MDMR')
library(MDMR)
```

二、样例数据

MDMR包含一个样例数据集 `mdmrdata`，包含500个样本，自变量包含3个变量，因变量包含10个变量。所有变量都进行了标准化（均值为0，方差为1）。

```
data(mdmrdata)
head(X.mdmr)
head(Y.mdmr)
```

三、计算距离矩阵

可以使用 `dist()` 函数计算常用的距离指标，也可以采用其他距离指标，比如前面文献（Shehzad et al., 2014）里使用的距离指标。

```
## using Manhattan distance
D <- dist(Y.mdmr, method = "manhattan")
## using distance measure defined in Shehzad2014
D2 <- sqrt(2 * (1 - cor(t(Y.mdmr))))
```

四、多变量距离回归

```
mdmr.res <- mdmr(X = X.mdmr, D = D)
summary(mdmr.res)
```

	Statistic	Numer.DF	Pseudo.R2	Analytic.p.value
(Omnibus)	0.1514	3	0.1315	< 1e-20 ***
X1	0.0629	1	0.0546	< 1e-13 ***
X2	0.0249	1	0.0217	3.2038e-06 ***
X3	0.0685	1	0.0594	< 1e-13 ***

MDMR分析的结果包含4行4列，4行分别表示模型整体和每个自变量的效应，4列分别表示统计量、自由度、伪R²和p值。类似于对多元线性回归结果的解读。

五、计算效应量

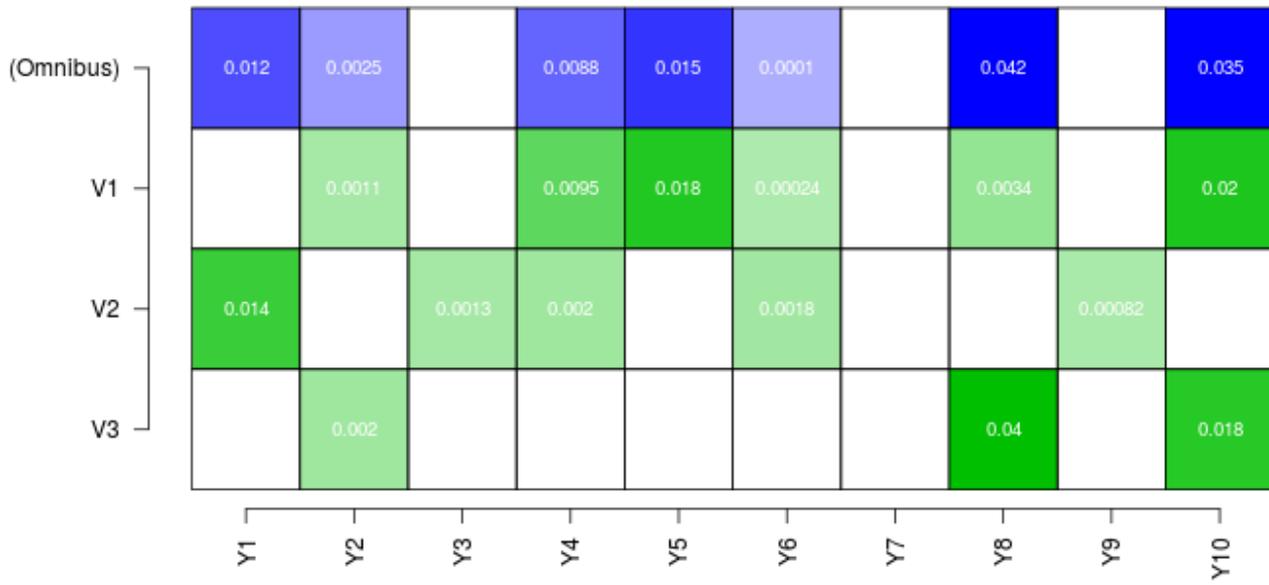
这里效应量指的是因变量中每个变量与自变量的关联的大小，计算思路是随机打乱某个因变量的顺序，重新进行前面的分析，将伪R²的减少程度作为效应量的度量。

1. 如果距离矩阵的计算是使用 `dist()` 函数，可以使用下面的方法获得每个因变量的效应量：

```
delta(X = X.mdmr, Y = Y.mdmr, dtype = "manhattan", plot.res = T, niter = 10, seed = 12345)
```

选项 `dtype` 用于指定距离指标，`plot.res` 如果设置为TRUE，会生成一个热图（如下图），`niter=10` 表示重复打乱过程10次，最终的结果取10次的平均。

MDMR Effect Sizes



2. 如果其他方法计算的的距离矩阵，可以使用下面的方法计算效应量：

```

## define a function to calculate distance
distFunc <- function(x){
  N <- nrow(x)
  distMat <- matrix(0, nrow=N, ncol=N)
  for (idx in c(1:N)){
    for (jdx in c(1:N)){
      distMat[idx, jdx] <- sum(abs(x[idx,] - x[jdx,]))
    }
  }
  return(distMat)
}
## calculate the G matrix
D <- distFunc(Y.mdmr)
G <- gower(D)
## calculate the G matrix after shuffling each column of Y
q <- ncol(Y.mdmr)
G.list <- lapply(1:q, FUN = function(k) {
  set.seed(k)
  Y.shuf <- Y.mdmr
  Y.shuf[,k] <- sample(Y.shuf[,k])
  gower(distFunc(Y.shuf))
})
names(G.list) <- colnames(Y.mdmr)
## calculate the effect size
delta(X = X.mdmr, G = G, G.list = G.list, plot.res = F)

```

这里假设不用 `dist()` 函数来计算距离，而是使用自定义函数的方式。注意这里为了简洁只打乱了1次，而第一种方法里打乱了10次取均值。