Project 8 Nested Effect Models

Team K - Minghang Li

April 25, 2023

Problem 20: Classical NEMs

Subproblem 1

Construct transitive closure and define Φ

Define a function to make any Φ matrix transitive closed by powering it up until convergence.

```
transitive_closify <- function(phi) {
  old_phi <- phi
  while (TRUE) {
    new_phi <- old_phi %*% phi
    new_phi[new_phi > 0] <- 1
    if (isTRUE(all.equal(new_phi, old_phi))) {
      break
    }
    old_phi <- new_phi
  }
  return(new_phi)
}
```

Construct Φ for Model (a).

```
phi_a <- array(
  dim = c(5, 5),
  dimnames = list(c("S1", "S2", "S3", "S4", "S5"),
                  c("S1", "S2", "S3", "S4", "S5"))
)
#                   S1, S2, S3, S4, S5
phi_a["S1", ] <- c( 1,  0,  1,  1,  0)
phi_a["S2", ] <- c( 0,  1,  0,  0,  1)
phi_a["S3", ] <- c( 0,  0,  1,  1,  1)
phi_a["S4", ] <- c( 0,  0,  0,  1,  1)
phi_a["S5", ] <- c( 0,  0,  0,  0,  1)
```

```
phi_a <- transitive_closify(phi_a)
phi_a
```

```
##    S1 S2 S3 S4 S5
## S1  1  0  1  1  1
## S2  0  1  0  0  1
## S3  0  0  1  1  1
## S4  0  0  0  1  1
## S5  0  0  0  0  1
```

Construct Φ for Model (b).

```
phi_b <- array(
  dim = c(5, 5),
  dimnames = list(c("S1", "S2", "S3", "S4", "S5"),
                  c("S1", "S2", "S3", "S4", "S5"))
)
#                  S1, S2, S3, S4, S5
phi_b["S1", ] <- c( 1,  0,  0,  1,  0)
phi_b["S2", ] <- c( 0,  1,  0,  0,  1)
phi_b["S3", ] <- c( 1,  0,  1,  1,  1)
phi_b["S4", ] <- c( 0,  0,  0,  1,  1)
phi_b["S5", ] <- c( 0,  0,  0,  0,  1)
```

```
phi_b <- transitive_closify(phi_b)
phi_b
```

```
##    S1 S2 S3 S4 S5
## S1  1  0  0  1  1
## S2  0  1  0  0  1
## S3  1  0  1  1  1
## S4  0  0  0  1  1
## S5  0  0  0  0  1
```

Define Θ

Define Θ for Model (a).

```
theta_a <- array(
  dim = c(5, 6),
  dimnames = list(c("S1", "S2", "S3", "S4", "S5"),
                  c("E1", "E2", "E3", "E4", "E5", "E6"))
)
#                    E1, E2, E3, E4, E5, E6
theta_a["S1", ] <- c( 0,  0,  0,  0,  0,  0)
theta_a["S2", ] <- c( 0,  0,  0,  1,  0,  1)
theta_a["S3", ] <- c( 1,  1,  0,  0,  0,  0)
theta_a["S4", ] <- c( 0,  0,  1,  0,  0,  0)
theta_a["S5", ] <- c( 0,  0,  0,  0,  1,  0)
theta_a
```

```
##    E1 E2 E3 E4 E5 E6
## S1  0  0  0  0  0  0
## S2  0  0  0  1  0  1
```

```
## S3   1  1  0  0  0  0
## S4   0  0  1  0  0  0
## S5   0  0  0  0  1  0
```

Define Θ for Model (b).

```
theta_b <- array(
  dim = c(5, 6),
  dimnames = list(c("S1", "S2", "S3", "S4", "S5"),
                  c("E1", "E2", "E3", "E4", "E5", "E6"))
)
#                     E1, E2, E3, E4, E5, E6
theta_b["S1", ] <- c( 1,  1,  0,  0,  0,  0)
theta_b["S2", ] <- c( 0,  0,  0,  1,  0,  1)
theta_b["S3", ] <- c( 0,  0,  0,  0,  0,  0)
theta_b["S4", ] <- c( 0,  0,  1,  0,  0,  0)
theta_b["S5", ] <- c( 0,  0,  0,  0,  1,  0)
theta_b
```

```
##     E1 E2 E3 E4 E5 E6
## S1   1  1  0  0  0  0
## S2   0  0  0  1  0  1
## S3   0  0  0  0  0  0
## S4   0  0  1  0  0  0
## S5   0  0  0  0  1  0
```

Determine the corresponding expected effect patterns (F)

```
F_a <- phi_a %*% theta_a
F_a
```

```
##     E1 E2 E3 E4 E5 E6
## S1   1  1  1  0  1  0
## S2   0  0  0  1  1  1
## S3   1  1  1  0  1  0
## S4   0  0  1  0  1  0
## S5   0  0  0  0  1  0
```

```
F_b <- phi_b %*% theta_b
F_b
```

```
##     E1 E2 E3 E4 E5 E6
## S1   1  1  1  0  1  0
## S2   0  0  0  1  1  1
## S3   1  1  1  0  1  0
## S4   0  0  1  0  1  0
## S5   0  0  0  0  1  0
```

Subproblem 2

If we assume no noise (no false positives and false negatives)… then the D matrix is simply the F matrix transpose.

```
D_a <- t(F_a)
D_a
```

```
##    S1 S2 S3 S4 S5
## E1  1  0  1  0  0
## E2  1  0  1  0  0
## E3  1  0  1  1  0
## E4  0  1  0  0  0
## E5  1  1  1  1  1
## E6  0  1  0  0  0
```

```
D_b <- t(F_b)
D_b
```

```
##    S1 S2 S3 S4 S5
## E1  1  0  1  0  0
## E2  1  0  1  0  0
## E3  1  0  1  1  0
## E4  0  1  0  0  0
## E5  1  1  1  1  1
## E6  0  1  0  0  0
```

Given the discrete data D_a and D_b (sorry for the different notation from the exercise pdf) it's not possible to tell apart the two models because they are identical.

```
all.equal(D_a, D_b)
```

```
## [1] TRUE
```

Subproblem 3

Calculate the marginal log-likelihood ratio (network score) given the data by setting the false positive rate to be 5% and the false negative rate to be 1%.

```
network_score_a <- scoreAdj(D_a,
                            adj = phi_a,
                            method = "disc",
                            fpfn = c(0.05, 0.01)
                            )$score
network_score_a
```

```
## [1] 60.304
```

```r
network_score_b <- scoreAdj(D_b,
                            adj = phi_b,
                            method = "disc",
                            fpfn = c(0.05, 0.01)
                            )$score
network_score_b
```

```
## [1] 60.304
```

Problem 21: Hidden Markov NEMs

Subproblem 1

Subproblem 2

Problem 22: Mixture NEMs

Subproblem 1

Subproblem 2

(a) Compute expected effect pattern $(\rho^T \phi_k \theta_k)^T$

(b)

Subproblem 3