Project 8

Santiago Castro Dau, June Monge, Rachita Kumar, Sarah Lötscher

Problem 20: Classical NEMs

1. For each model, construct the transitive closure (by adding edges) and define the corresponding adjacency matrices Φ and Θ, which represent the signalling pathways and the E-gene attachments. Determine the corresponding expected effect patterns (F).

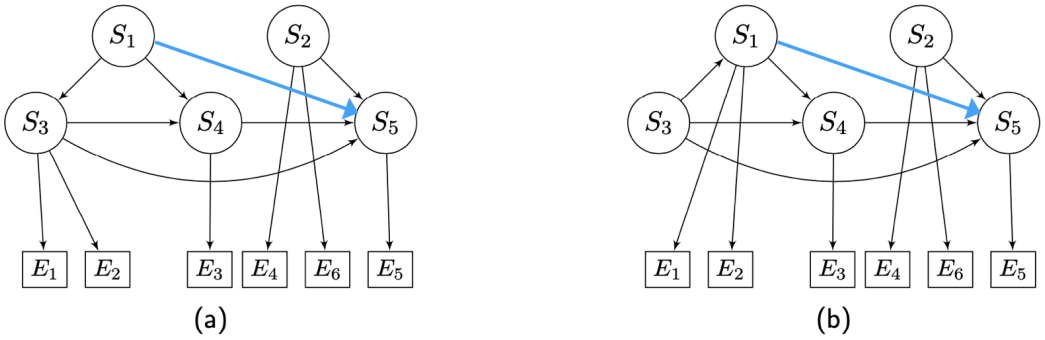

Figure 1: New edges drawn in blue.

In the following adjacency matrices Φ a non zero the Φ_{ij} element in the matrix indicates that node S_i is connected to S_j and that this edge is directed towards S_j.

$$\Phi_a = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Phi_b = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Fpr the following matrices Θ, $\Theta_{ij} = 1$, if E-gene j is regulated by S-gene i.

$$\Theta_a = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\Theta_b = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The expected effect pattern is given by the expression $F = \Phi\Theta$.

```
phi_a = matrix(data = c(1, 0, 1, 1, 1,
                        0, 1, 0, 0, 1,
                        0, 0, 1, 1, 1,
                        0, 0, 0, 1, 1,
                        0, 0, 0, 0, 1),
               ncol = 5,
               nrow = 5,
               byrow = TRUE,
               dimnames = list(c("S1", "S2", "S3", "S4", "S5"),
                               c("S1", "S2", "S3", "S4", "S5")))

phi_b = matrix(data = c(1, 0, 0, 1, 1,
                        0, 1, 0, 0, 1,
                        1, 0, 1, 1, 1,
                        0, 0, 0, 1, 1,
                        0, 0, 0, 0, 1),
               ncol = 5,
               nrow = 5,
               byrow = TRUE,
               dimnames = list(c("S1", "S2", "S3", "S4", "S5"),
                               c("S1", "S2", "S3", "S4", "S5")))

theta_a = matrix(data = c(0, 0, 0, 0, 0, 0,
                          0, 0, 0, 1, 0, 1,
                          1, 1, 0, 0, 0, 0,
                          0, 0, 1, 0, 0, 0,
                          0, 0, 0, 0, 1, 0),
                 ncol = 6,
                 nrow = 5,
                 byrow = TRUE,
                 dimnames = list(c("S1", "S2", "S3", "S4", "S5"),
                                 c("E1", "E2", "E3", "E4", "E5","E6")))

theta_b = matrix(data = c(1, 1, 0, 0, 0, 0,
                          0, 0, 0, 1, 0, 1,
                          0, 0, 0, 0, 0, 0,
                          0, 0, 1, 0, 0, 0,
                          0, 0, 0, 0, 1, 0),
                 ncol = 6,
                 nrow = 5,
```

```
                byrow = TRUE,
                dimnames = list(c("S1", "S2", "S3", "S4", "S5"),
                                c("E1", "E2", "E3", "E4", "E5","E6")))

# F_a
phi_a %*% theta_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
```

```
##    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
```

2. Assuming no noise, determine the discrete data D_1 and D_2 from both models. Given only the data, can you tell apart the two models?

Assuming one perturbation experiment for each S-gene, the binarized data matrix D with entries $e_{ji} = 1$ if S-gene i had an effect on E-gene j, and $e_{ji} = 0$ otherwise.

```
D1 = array(dim = c(6, 5), dimnames = list(c("E1", "E2", "E3", "E4", "E5","E6"),
                                          c("S1", "S2", "S3", "S4", "S5")))

D1["E1",] = c(1,0,1,0,0)
D1["E2",] = c(1,0,1,0,0)
D1["E3",] = c(1,0,1,1,0)
D1["E4",] = c(0,1,0,0,0)
D1["E5",] = c(1,1,1,1,0)
D1["E6",] = c(0,1,0,0,1)

D2 = array(dim = c(6, 5), dimnames = list(c("E1", "E2", "E3", "E4", "E5","E6"),
                                          c("S1", "S2", "S3", "S4", "S5")))

D2["E1",] = c(1,0,1,0,0)
D2["E2",] = c(1,0,1,0,0)
D2["E3",] = c(1,0,1,1,0)
D2["E4",] = c(0,1,0,0,0)
D2["E5",] = c(1,1,1,1,0)
D2["E6",] = c(0,1,0,0,1)

D1
```

```
##    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  0
## E6  0  1  0  0  1
```

```
D2
```

```
##    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  0
## E6  0  1  0  0  1
```

Since the Data matrices D1 and D2 are identical, we cannot tell the two models apart.

3. Take D_1 and D_2 from the previous question. For each model, 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%.

```
library(mnem)
```

```
## Registered S3 methods overwritten by 'RcppEigen':
##   method               from
##   predict.fastLm       RcppArmadillo
##   print.fastLm         RcppArmadillo
##   summary.fastLm       RcppArmadillo
##   print.summary.fastLm RcppArmadillo
```

```
scoreAdj(D = D1, adj = phi_a, method="disc", fpfn=c(0.05,0.01))$score
```

```
## [1] 51.68914
```

```
scoreAdj(D = D2, adj = phi_b, method="disc", fpfn=c(0.05,0.01))$score
```

```
## [1] 51.68914
```

Problem 21: Hidden Markov NEMs

```
u = t(array(c(c(1,1,1,0),
              c(0,1,1,1),
              c(0,0,1,1),
              c(0,0,0,1)),
              dim = c(4, 4), dimnames = list(c("S1", "S2", "S3", "S4"),
                                             c("S1", "S2", "S3", "S4"))))
v1 = t(array(c(c(1,1,1,0),
              c(0,1,1,1),
              c(0,0,1,0),
              c(0,0,0,1)),
              dim = c(4, 4), dimnames = list(c("S1", "S2", "S3", "S4"),
                                             c("S1", "S2", "S3", "S4"))))
v2 = t(array(c(c(1,0,0,0),
              c(1,1,1,0),
              c(1,0,1,0),
              c(1,0,0,1)),
              dim = c(4, 4), dimnames = list(c("S1", "S2", "S3", "S4"),
```

```
                                              c("S1", "S2", "S3", "S4"))))

lambdas = seq(0.1, 0.9, by=0.1)

s_uv1 = sum(u!=v1)
s_uv2 = sum(u!=v2)

Trn = array(dim = c(9,2), dimnames = list(lambdas,c("v1", "v2")))
models = mnem:::enumerate.models(4,
                                 name=c("S1", "S2", "S3", "S4"),
                                 trans.close = TRUE,
                                 verbose=FALSE)

for(lambda in lambdas){
  C = 0

  for (model in models) {
    C = C + (1-lambda)^sum(u != model)
  }

  C = C

  Trn[as.character(lambda),"v1"] = (1/C)*((1-lambda)^s_uv1)
  Trn[as.character(lambda),"v2"] = (1/C)*((1-lambda)^s_uv2)
}

Trn
```

```
##              v1           v2
## 0.1 0.004554634 2.420519e-03
## 0.2 0.007483512 1.961758e-03
## 0.3 0.012503102 1.470977e-03
## 0.4 0.021261524 9.919777e-04
## 0.5 0.036854418 5.758503e-04
## 0.6 0.065292226 2.674370e-04
## 0.7 0.118810462 8.661283e-05
## 0.8 0.224199580 1.434877e-05
## 0.9 0.447865533 4.478655e-07
```

2. Plot the transition probabilities for v_1 and v_2 as a function of λ. Describe the transition probabilities as a function of λ.

```
library(reshape2)
library(ggplot2)
library(RColorBrewer)
```

```
## Warning: package 'RColorBrewer' was built under R version 4.0.5
```

```
data = data.frame(melt(Trn))
colnames(data)<-c("lambda","v","T")
plot<-ggplot(data,aes(x=lambda,y=T,color=v))+
  geom_point()+
  theme_classic()+
  ylab("Transition probaility")+
```

```
  labs(title="Transition probabilities as a function of lambda")
plot
```