Assignment_1

Jieran Sun, HJ, Gummi

2023-02-27

Question 1

For the network models a) and b), case a) holds the statement of $A \perp B \mid C$, whereas b) holds the statement of $A \perp B$.

For case a):

$$P(A, B \mid C) = \frac{P(A, B, C)}{P(C)} \tag{1}$$

$$= \frac{P(A \mid C)P(B \mid C)P(C)}{P(C)} \tag{2}$$

$$= P(A \mid C)P(B \mid C) \tag{3}$$

Hence the conditional independence.

For case b):

$$P(C \mid A, B) = \frac{P(A, B, C)}{P(A, B)} \tag{4}$$

$$= \frac{P(C \mid A, B)P(B)P(A)}{P(A, B)} \tag{5}$$

$$p(A, B) = P(A)P(B) \tag{6}$$

Question 2

Markov blanket $MB(D)$ is the set of nodes composed of the parents, co-parents and children of D. In this case, $MB(D) = \{B, F, C, G, E\}$. Given that

$$P(D \mid MB(D), A) = \frac{P(D, MB(D), A)}{P(A, MB(D))} \tag{7}$$

$$= \frac{P(A \mid D, MB(D))P(D, MB(D))}{P(A \mid MB(D))P(MB(D))} \tag{8}$$

$$= \frac{P(A)P(D, MB(D))}{P(A)P(MB(D))} \tag{9}$$

$$= \frac{P(D, MB(D))}{P(MB(D))} \tag{10}$$

$$= P(D \mid MB(D)) \tag{11}$$

Question 3

a)

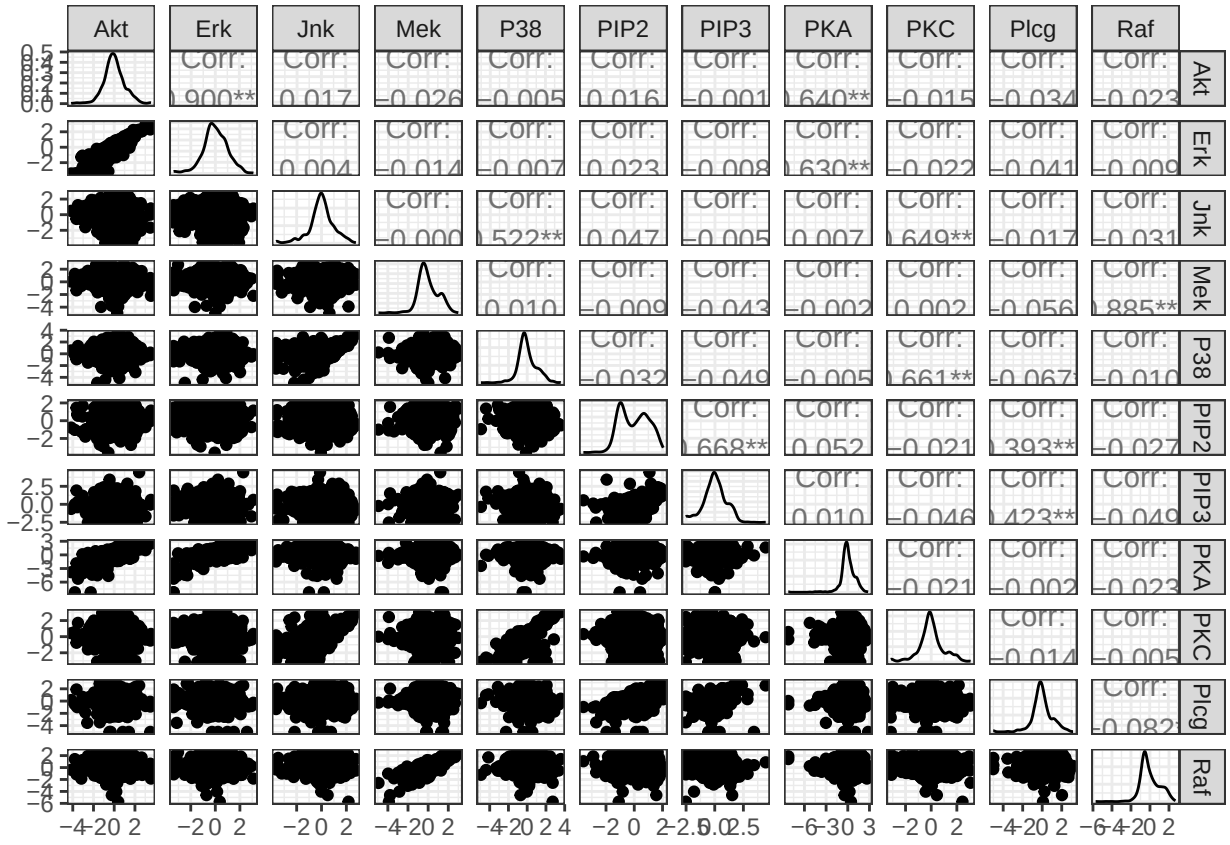
```
data <- read.csv("https://raw.githubusercontent.com/felixleopoldo/benchpress/master/resources/data/myda

set.seed(2023)
ind <- sample(1:nrow(data), as.integer(0.8*nrow(data)), replace = FALSE)
train_data <- data[ind,]
test_data <- data[-ind,]

cat(sprintf("Number of variables n: %s, the number of observations N: %s.", ncol(data), nrow(data)))
```

```
## Number of variables n: 11, the number of observations N: 902.
```

```
ggpairs(data) + theme_bw()
```



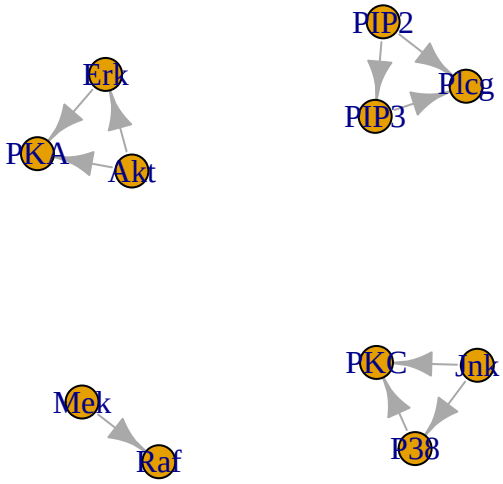

```
Para_init <- BiDAG::scoreparameters(scoretype = "bge",
                                    train_data)
```

```
Bayes_network <- BiDAG::iterativeMCMC(scorepar = Para_init)
```

```
## maximum parent set size is 2
```

```
## core space defined, score table are being computed
## score tables completed, iterative MCMC is running
```

```
dag_bayes <- Bayes_network$CPDAG %>% graph_from_adjacency_matrix(mode = "directed")
plot(dag_bayes)
```




```
eval_bayes <- BiDAG::scoreagainstDAG(scorepar = Para_init,
                                     incidence = getDAG(Bayes_network),
                                     datatoscore = test_data)
cat(sprintf("The average score is %.3f", mean(eval_bayes)))
```

```
## The average score is -12.593
```

```
amtuning <- function(am_v){
  # Reproducibility
  set.seed(2023)

  log_score_sum <- 0
  edge_sum <- 0

  for (i in 1:100){

    # Split the data
    ind <- sample(1:nrow(data), as.integer(0.8*nrow(data)), replace = FALSE)
```

```
    train_data <- data[ind,]
    test_data <- data[-ind,]

    # Initialize the parameters
    Para_init <- BiDAG::scoreparameters(scoretype = "bge",
                                        bgepar = list(am = am_v, aw = NULL, edgepf = 1),
                                        train_data)

    # Network construction
    Bayes_network <- BiDAG::iterativeMCMC(scorepar = Para_init)
    eval_mean <- mean(BiDAG::scoreagainstDAG(scorepar = Para_init, incidence = getDAG(Bayes_network),
                                       datatoscore = test_data))
    edge_mean <- sum(getDAG(Bayes_network))
    log_score_sum <- log_score_sum + eval_mean
    edge_sum <- edge_sum + edge_mean
  }
  c(score = log_score_sum/100, edge = edge_sum/100)
}

am_list <- c(1e-3, 1e-1, 1, 10, 1e2)

registerDoParallel(5)
results_df <- foreach (am = am_list, .combine=rbind) %dopar% {
  amtuning(am)
}
stopImplicitCluster()
```

```
am_list <- c(1e-3, 1e-1, 1, 10, 1e2)
results_df <- readRDS("results.rds")
results_df_prez <- as.data.frame(results_df, row.names = as.character(am_list))
knitr::kable(results_df_prez)
```

	score	edge
0.001	-12.89803	7.01
0.1	-12.87736	9.32
1	-12.87260	10.21
10	-12.87824	12.94
100	-12.99792	15.57

The final average log score is taken through the average of the test dataset.

```
OptAM <- am_list[which.max(results_df[,"score"])]
cat(sprintf("The am value corrresponding to the highest score is %s", OptAM))
```

```
## The am value corrresponding to the highest score is 1
```

```
set.seed(2023)
# build the final product parameter
Para_final <- BiDAG::scoreparameters(scoretype = "bge",
                                     bgepar = list(am = OptAM, aw = NULL, edgepf = 1),
```

```
                                    data = data)

# build the network
Bayes_network_final <- BiDAG::iterativeMCMC(scorepar = Para_final)
```

```
## maximum parent set size is 3
## core space defined, score table are being computed
## score tables completed, iterative MCMC is running
```

```
# Plot the DAG
Bayes_network_final$DAG %>% graph_from_adjacency_matrix(mode = "directed") %>% igraph::plot.igraph()
```