Assignment3

Yuki Joyama (yj2803)

Problem 1

$X \sim N(\mu, \Sigma)$ where $X = (X_1, ...X_p)$
The conditional distribution $X_i, X_j | X_s = x_s$ is still a Gaussian distribution with mean and covariance given by: $X_i, X_j | X_s = x_s \sim N(\mu_{ij|S}, \Sigma_{ij|S})$

$$\left(\Sigma_{\{ijS\},\{ijS\}}\right)^{-1} = \begin{pmatrix} K_{\{i,j\},\{i,j\}} & K_{\{i,j\},S} \\ K_{S,\{i,j\}} & K_S \end{pmatrix}$$

When $X_i \perp\!\!\!\perp X_j | X_s$ is true, $K_{ij|S} = 0$, which makes $\left(\Sigma_{\{ijS\},\{ijS\}}\right)^{-1}$ diagonal. Therefore, $X_i \perp\!\!\!\perp X_j | X_s \Rightarrow (\Sigma_{\{ijS\},\{ijS\}})^{-1}_{ij} = 0$ holds.
When $(\Sigma_{\{ijS\},\{ijS\}})^{-1}_{ij} = 0$ is true, $K_{ij|S} = (\Sigma_{ij|S})^{-1} = 0$. $(\Sigma_{ij|S})^{-1}$ is the precision matrix for the conditional distribution $X_i, X_j | X_s = x_s$, so X_i and X_j are independent given X_s.
Therefore, $X_i \perp\!\!\!\perp X_j | X_s \Leftrightarrow (\Sigma_{\{ijS\},\{ijS\}})^{-1}_{ij} = 0$.

Problem 2

```
# simulate data from a given MRF independence model
set.seed(123)
K <- cbind(c(10,7,7,0),c(7,20,0,7),c(7,0,30,7),c(0,7,7,40))
data <- as.data.frame(mvrnorm(n=10000,mu=c(0,0,0,0),Sigma=solve(K)))
colnames(data) <- c("X1","X2","X3","X4")
K
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   10    7    7    0
## [2,]    7   20    0    7
## [3,]    7    0   30    7
## [4,]    0    7    7   40
```

In the precision matrix, $K_{ij} = 0$ implies that variable X_i and X_j are conditionally independent given all other variables. Given the precision matrix K, $K_{14} = K_{41} = 0$ and $K_{23} = K_{32} = 0$. Therefore the following conditional independencies are represented by K:
$X_1 \perp\!\!\!\perp X_4 | X \backslash \{X_1, X_4\}$
$X_2 \perp\!\!\!\perp X_3 | X \backslash \{X_2, X_3\}$ The corresponding graph is an undirected graph that has no edges between X_1 and

X_4, and X_2 and X_3. All other pairs of variables are connected by edges.
Now, I will verify the conditional independence constraints by using linear regression.
$X_1 \perp\!\!\!\perp X_4 | X \backslash \{X_1, X_4\}$:

```
# conditional independence of X1 and X4 given X2, X3
m14 = lm(X1 ~ X2 + X3 + X4, data = data)
summary(m14)
```

```
##
## Call:
## lm(formula = X1 ~ X2 + X3 + X4, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.36729 -0.21127  0.00304  0.21389  1.20994
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.001934   0.003141   0.616    0.538
## X2          -0.682729   0.012203 -55.950   <2e-16 ***
## X3          -0.695282   0.015540 -44.741   <2e-16 ***
## X4           0.007927   0.020037   0.396    0.692
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3141 on 9996 degrees of freedom
## Multiple R-squared:  0.4564, Adjusted R-squared:  0.4563
## F-statistic:  2798 on 3 and 9996 DF,  p-value: < 2.2e-16
```

In this linear model, the coefficient of $X4$ turned out to be non-significant with p-value <0.05.

$X_2 \perp\!\!\!\perp X_3 | X \backslash \{X_2, X_3\}$:

```
# conditional independence of X2 and X3 given X1, X4
m23 = lm(X2 ~ X1 + X3 + X4, data = data)
summary(m23)
```

```
##
## Call:
## lm(formula = X2 ~ X1 + X3 + X4, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90282 -0.15318  0.00188  0.15342  0.85952
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.001141   0.002247   0.508    0.612
## X1          -0.349303   0.006243 -55.950   <2e-16 ***
## X3           0.012316   0.012177   1.011    0.312
## X4          -0.352810   0.013891 -25.398   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2246 on 9996 degrees of freedom
## Multiple R-squared:  0.3841, Adjusted R-squared:  0.3839
## F-statistic:  2078 on 3 and 9996 DF,  p-value: < 2.2e-16
```

In this linear model, the coefficient of $X3$ turned out to be non-significant with p-value <0.05.

Therefore, the conditional independencies are verified.

The list of edges are $X_1 - X_2$, $X_1 - X_3$, $X_2 - X_4$, $X_3 - X_4$.

```
# fit the model (estimate the precision matrix subject to the graph constraints)
glist <- list(
  c("X1", "X2"),
  c("X1", "X3"),
  c("X2", "X4"),
  c("X3", "X4")
)
ddd <- cov.wt(data, method="ML")
fit <- ggmfit(ddd$cov, ddd$n.obs, glist) # Estimate parameters using IPF
fit$K # estimated precision matrix
```

```
##           X1        X2        X3        X4
## X1 10.182411  6.988142  7.140856  0.000000
## X2  6.988142 19.832337  0.000000  7.076402
## X3  7.140856  0.000000 29.394792  6.852069
## X4  0.000000  7.076402  6.852069 40.745105
```

It appears that the model fitting worked because we can see that the estimated precision matrix has $K_{14} = K_{41} = 0$ and $K_{23} = K_{32} = 0$, and everything else non-zero, indicating that the above conditional independencies hold.

Problem 3

```
# Gaussian Bayesian Network model
# covariance matrix
set.seed(123)
```

```r
Sig <- cbind(c(3,-1.4,0,0),c(-1.4,3,1.4,1.4),c(0,1.4,3,0),c(0,1.4,0,3))
data <- as.data.frame(mvrnorm(n=10000,mu=c(0,0,0,0),Sigma=Sig))
colnames(data) <- c("X1","X2","X3","X4")
```

DAG $\mathcal{G}$: $X_1 \to X_2 \leftarrow X_3$ and $X_4 \to X_2$

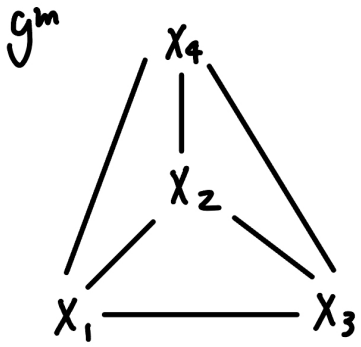
(a) Given the model, correlation constraints are:
$X_1 \perp\!\!\!\perp X_3 | X_2$
$X_1 \perp\!\!\!\perp X_4 | X_2$
$X_3 \perp\!\!\!\perp X_4 | X_2$

```r
# estimate the correlation
cor(data)
```

```
##             X1         X2          X3           X4
## X1  1.00000000 -0.4661188 0.012187930 -0.011504563
## X2 -0.46611880  1.0000000 0.463034923  0.473314198
## X3  0.01218793  0.4630349 1.000000000  0.006392376
## X4 -0.01150456  0.4733142 0.006392376  1.000000000
```

We can see that the correlation between X_1 and X_3, X_1 and X_4, X_3 and X_4 are very close to zero.

(b)



The precision matrix K for $\mathcal{G}^m$ is 4 by 4 matrix with non-zero off-diagonal elements because all the pairs in the moralized graph are connected by edges. This indicates that all the pairs are conditionally dependent given the rest of other variables, meaning that they have non-zero partial correlations.
Even though the covariance matrix Σ shows zero direct covariances between some pairs of variables, the precision matrix K reflects the conditional dependencies imposed by the moralized graph $\mathcal{G}^m$ where all pairs of variables are connected.

(c)

```r
# estimate precision matrix K
estK = ggmfit(cov.wt(data, method = "ML")$cov, cov.wt(data, method = "ML")$n.obs, glist = glist)$K

# output estimated K
estK
```

```
##              X1         X2           X3           X4
## X1  0.427104764  0.2001358 -0.004798841  0.000000000
## X2  0.200135811  0.5291716  0.000000000 -0.202133735
## X3 -0.004798841  0.0000000  0.337174079 -0.003159849
## X4  0.000000000 -0.2021337 -0.003159849  0.418846284
```

```
# take the inverse of K
solve(estK)
```

```
##             X1          X2          X3          X4
## X1  2.99172214 -1.38708078  0.03630902 -0.66912636
## X2 -1.38708078  2.95998158 -0.00635506  1.42842869
## X3  0.03630902 -0.00635506  2.96652480  0.01931305
## X4 -0.66912636  1.42842869  0.01931305  3.07701107
```

```
# output the true covariance matrix
Sig
```

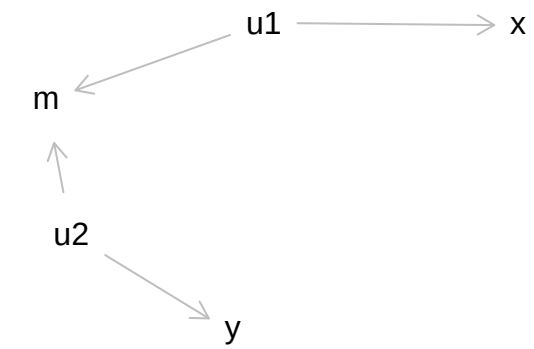
```
##      [,1] [,2] [,3] [,4]
## [1,]  3.0 -1.4  0.0  0.0
## [2,] -1.4  3.0  1.4  1.4
## [3,]  0.0  1.4  3.0  0.0
## [4,]  0.0  1.4  0.0  3.0
```

The estimated covariance matrix is mostly similar to true covariance matrix but differ slightly on X_2 and X_3, and X_1 and X_4.

Problem 4

```
library(dagitty)

# simulate 10000 observations from the following graph
g <- dagitty( "dag{ x <- u1; u1 -> m <- u2 ; u2 -> y }" )
plot(g)
```



```r
# simulate data based on the DAG
set.seed(123)
n = 10000
u1 = rnorm(n)
u2 = rnorm(n)
x = u1 + rnorm(n)
m = u1 + u2 + rnorm(n)
y = u2 + rnorm(n)

data <- data.frame(x = x, m = m, y = y)

# estimate the effect of x on y adjusting for m in a linear regression
m1 = lm(y ~ x + m, data = data)
summary(m1)
```

```
##
## Call:
## lm(formula = y ~ x + m, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6875 -0.8515  0.0020  0.8471  4.6296
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.011052   0.012631   0.875    0.382
## x           -0.192179   0.009655 -19.905   <2e-16 ***
## m            0.411101   0.008010  51.320   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.263 on 9997 degrees of freedom
## Multiple R-squared:  0.2086, Adjusted R-squared:  0.2084
## F-statistic:  1317 on 2 and 9997 DF,  p-value: < 2.2e-16
```

```r
# 95% CI for the effect of x on y adjusting for m
confint(m1, "x", level = 0.95)
```

```
##        2.5 %     97.5 %
## x -0.2111049 -0.1732535
```

The estimate of X on Y adjusted for M is -0.19 with a 95% CI of (-0.21, -0.17), which is statistically significant.

```r
# identify the sufficient adjustment set from dagitty
adjustmentSets(g, exposure = "x", outcome = "y")
```

```
##  {}
```

The empty set was returned, which makes sense because M is a collider in the path X to Y and should not be adjusted.
The new estimate and 95% CI can be obtained by the following:

```r
# estimate the effect of x on y adjusting for nothing in a linear regression
m2 = lm(y ~ x, data = data)
summary(m2)
```

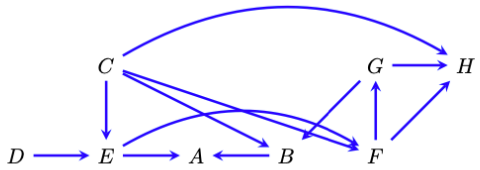
```
##
## Call:
## lm(formula = y ~ x, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1477 -0.9357  0.0027  0.9579  5.3572
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.006057   0.014196   0.427    0.670
## x           0.006232   0.009944   0.627    0.531
##
## Residual standard error: 1.42 on 9998 degrees of freedom
## Multiple R-squared:  3.928e-05,  Adjusted R-squared:  -6.074e-05
## F-statistic: 0.3927 on 1 and 9998 DF,  p-value: 0.5309
```

```r
# 95% CI for the effect of x on y adjusting for nothing
confint(m2, "x", level = 0.95)
```

```
##         2.5 %     97.5 %
## x -0.01326051 0.02572401
```

The estimate of X on Y adjusted for nothing is 0.006 with a 95% CI of (-0.013, 0.025), which is not statistically significant.
The first case implies that there is a causal effect of X on Y, and the second case implies the opposite. According to the DAG, adjusting for notation would be more appropriate when examining the effect of X on Y, and we confirmed that X has no causal effect on Y from the linear regression. The first case tells us that misadjustment can potentially create a non-existent causal effect between two variables and should be evaluated cautiously.

Problem 5




```
# construct the above as a daggity object
g <- dagitty( "dag{
               D -> E -> A <- B <- G -> H <- F;
               F <- E <- C -> H;
               B <- C -> F -> G
}" )

# simulate 10000 observations
set.seed(123)
sim = simulateSEM(g, N = 10000)
head(sim)
```

```
##              A           B          C          D          E          F
## 1 -0.296489126 -0.66420987 -0.4821885  2.3765598 -0.8687467  0.1416106
## 2  0.710138090  0.81157636  1.5104003  0.7692584  0.9370280  1.8130595
## 3  0.303585431  0.28045377  0.1677099 -1.1840606  2.0868906  0.5514855
## 4  0.536824952  0.05972787  1.1252700 -0.9752382  0.4386372 -0.5726999
## 5 -0.006931758 -1.70142449 -1.1669347  0.7692714  0.2457463 -0.2883801
## 6  0.808882754 -0.55947177 -0.2054906 -0.1735509 -0.5435384  0.0426904
##            G          H
## 1  0.8724668  0.4270816
## 2 -1.0372098  0.9486101
## 3  0.2454710 -0.3545600
## 4 -0.3382455  0.7674376
## 5 -0.9355281  0.4053085
## 6 -0.7233592  0.8459079
```

```
# identify the sufficient adjustment set from dagitty
adjustmentSets(g, exposure = "E", outcome = "F")
```

```
## { C }
```

```
adjustmentSets(g, exposure = "B", outcome = "A")
```

```
## { E }
## { C, F }
## { C, G }
```

Given the sufficient adjustment set, I will investigate the effect of E on F adjusting for C and the effect of B on A adjusting for {E}, {C, F}, {C, G} using the linear regression.

```
# linear models
# E on F
mEF = lm(F ~ E + C, data=sim)
summary(mEF)
```

```
##
## Call:
## lm(formula = F ~ E + C, data = sim)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1224 -0.5303  0.0064  0.5225  3.2563
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.003953   0.007791   0.507    0.612
## E           0.470705   0.007786  60.458   <2e-16 ***
## C           0.444186   0.007815  56.840   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.779 on 9997 degrees of freedom
## Multiple R-squared:  0.3834, Adjusted R-squared:  0.3833
## F-statistic:  3108 on 2 and 9997 DF,  p-value: < 2.2e-16
```

```
# B on A
mBA_E = lm(A ~ B + E, data=sim)
summary(mBA_E)
```

```
##
## Call:
## lm(formula = A ~ B + E, data = sim)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8005 -0.6565  0.0164  0.6377  3.6255
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.004713   0.009592  -0.491    0.623
## B           -0.243989   0.009511 -25.653  < 2e-16 ***
## E            0.048336   0.009531   5.071 4.02e-07 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9591 on 9997 degrees of freedom
## Multiple R-squared:  0.06451,    Adjusted R-squared:  0.06433
## F-statistic: 344.7 on 2 and 9997 DF,  p-value: < 2.2e-16
```

```
mBA_CF = lm(A ~ B + C + F, data=sim)
summary(mBA_CF)
```

```
##
## Call:
## lm(formula = A ~ B + C + F, data = sim)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7038 -0.6547  0.0120  0.6427  3.6614
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.004607   0.009596  -0.480  0.63115
## B           -0.242145   0.010167 -23.816  < 2e-16 ***
## C           -0.029881   0.010964  -2.725  0.00643 **
## F            0.042920   0.010553   4.067  4.8e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9595 on 9996 degrees of freedom
## Multiple R-squared:  0.06382,    Adjusted R-squared:  0.06354
## F-statistic: 227.2 on 3 and 9996 DF,  p-value: < 2.2e-16
```

```
mBA_CG = lm(A ~ B + C + G, data=sim)
summary(mBA_CG)
```

```
##
## Call:
## lm(formula = A ~ B + C + G, data = sim)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7570 -0.6590  0.0129  0.6381  3.6725
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.004237   0.009603  -0.441    0.659
```

```
## B           -0.247460   0.012719 -19.456   <2e-16 ***
## C           -0.011443   0.010495  -1.090    0.276
## G            0.011299   0.011919   0.948    0.343
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9603 on 9996 degrees of freedom
## Multiple R-squared:  0.06236,    Adjusted R-squared:  0.06208
## F-statistic: 221.6 on 3 and 9996 DF,  p-value: < 2.2e-16
```

The estimates when adjusting for all other variables in the graph are as follows:

```
# E on F
mEF_all = lm(F ~ E + A + B + C + D + G + H, data=sim)
summary(mEF_all)
```

```
##
## Call:
## lm(formula = F ~ E + A + B + C + D + G + H, data = sim)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0812 -0.5288  0.0041  0.5191  3.1953
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.003287   0.007770   0.423    0.672
## E            0.468390   0.009274  50.505  < 2e-16 ***
## A            0.013820   0.008101   1.706    0.088 .
## B            0.012633   0.010484   1.205    0.228
## C            0.461653   0.009636  47.908  < 2e-16 ***
## D            0.001400   0.009229   0.152    0.879
## G            0.038185   0.009653   3.956 7.68e-05 ***
## H           -0.044818   0.009035  -4.960 7.15e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7767 on 9992 degrees of freedom
## Multiple R-squared:  0.3873, Adjusted R-squared:  0.3869
## F-statistic: 902.3 on 7 and 9992 DF,  p-value: < 2.2e-16
```

```
# B on A
mBA_all = lm(A ~ B + E + C + D + F + G + H, data=sim)
summary(mBA_all)
```

```
##
## Call:
## lm(formula = A ~ B + E + C + D + F + G + H, data = sim)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7478 -0.6549  0.0145  0.6403  3.6220
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.004824   0.009594  -0.503 0.615122
## B           -0.247376   0.012707 -19.467  < 2e-16 ***
## E            0.043587   0.012823   3.399 0.000679 ***
## C           -0.013032   0.013194  -0.988 0.323308
## D            0.011461   0.011395   1.006 0.314557
## F            0.021071   0.012351   1.706 0.088031 .
## G            0.008458   0.011928   0.709 0.478288
## H           -0.003955   0.011170  -0.354 0.723280
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9591 on 9992 degrees of freedom
## Multiple R-squared:  0.06502,    Adjusted R-squared:  0.06436
## F-statistic: 99.26 on 7 and 9992 DF,  p-value: < 2.2e-16
```

Effect of E on F

The point estimate for E on F when adjusting for the set {C} is 0.4707 with a standard error of 0.0078, while adjusting for all other variables yields a point estimate of 0.4684 with a standard error of 0.0093. The point estimates are very similar, but the variance is slightly higher in the model adjusting for all other variables, likely due to increased noise from including unnecessary variables, which adds complexity without improving the estimate.

Effect of B on A The point estimate for B on A when adjusting for the set {E} is -0.2440 with a standard error of 0.0096. When adjusting for the set {C, F}, the point estimate is -0.2421 with a standard error of 0.0102. Adjusting for the set {C, G} gives a point estimate of -0.2475 with a standard error of 0.0127. Finally, when adjusting for all other variables, the point estimate is -0.2474 with a standard error of 0.0127. All the point estimates are quite similar across the different adjustment sets. However, the variance is slightly higher in the model that adjusts for all other variables compared to the models using the sufficient adjustment sets, likely for the same reasons discussed earlier. Additionally, the model adjusted for the set {E} yields the lowest variance, likely because {E} is the minimal sufficient adjustment set for estimating the effect of B on A according to the DAG. As a result, it introduces less noise compared to the other models that adjust for more variables.