Project 9

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

2022-05-05

Problem 23: d-separation

1. Write down all the variables that are d-separated from A given {C,D}

A and G are d-separated given $\{C, D\}$ since both the paths A-C-G and A-D-G are blocked.

2. Indicate whether each statement is true or false and explain your choice.

(a) B is conditionally independent of C given D. False–> the path B-D-A-C is still active

(b) G is conditionally independent of E given D. False–> the path G-C-A-D-B-F-E is still active

(c) C is conditionally independent of F given A. True–> true since all paths are inactive

(d) C is conditionally independent of E given its Markov blanket (of C) True –> Generally for any node if conditioned on the markov blanket its going to be independent of the other nodes. Also all the paths between C and E are blocked given the Markov Blanket of C (e.g. $\{A, D, G\}$).

Problem 24: Testing for marginal correlation

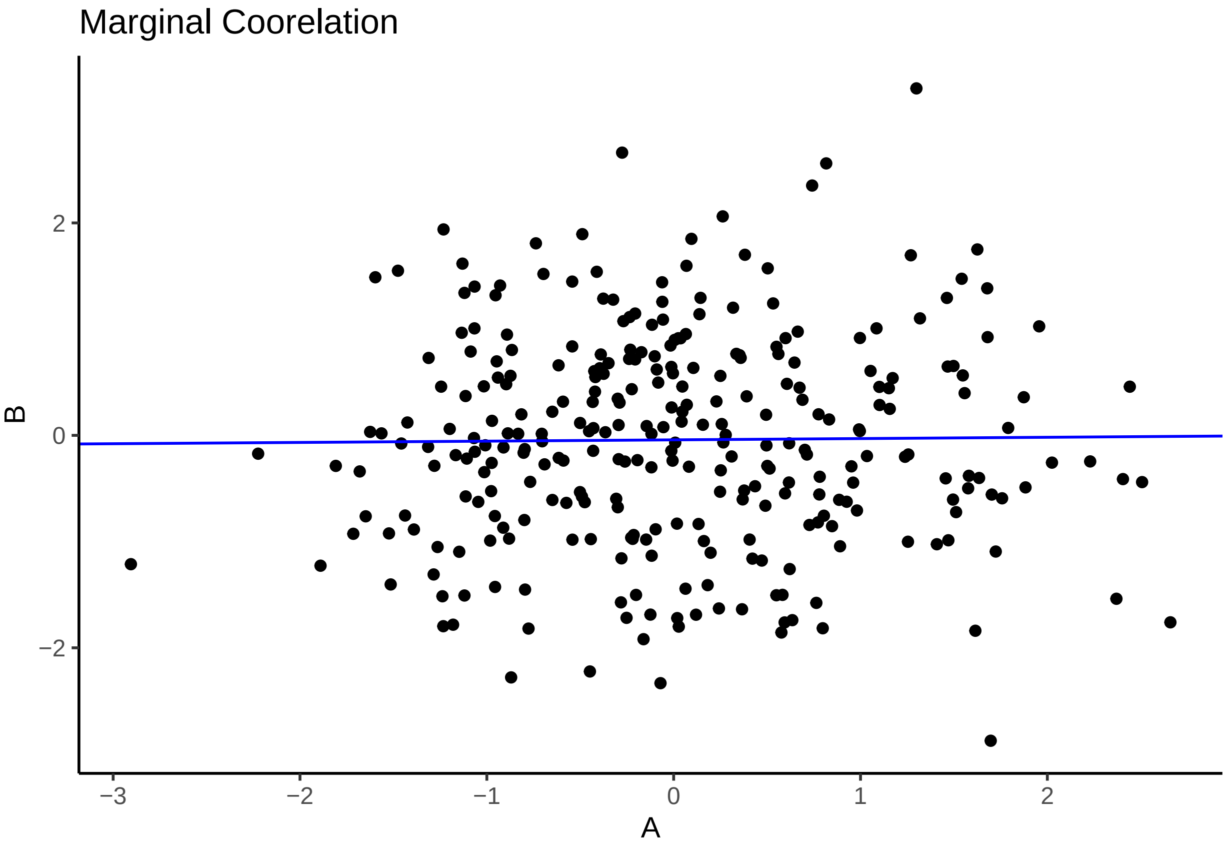
Using the data from MVN DAG.rds2, display the observations of A and B in a scatterplot.

```
library("ggplot2")
library("pcalg")
library("BiDAG")
```

```
#Read in dataset
set.seed(0)
DAG<-readRDS("./MVN_DAG.rds", refhook = NULL)
```

```
#Create the scatterplot

slr.coef<-coef(lm(B~ A,DAG))
qplot(A,B,data=DAG)+theme_classic()+
  geom_abline(intercept =slr.coef[1], slope = slr.coef[2],colour="blue" )+
  ggtitle("Marginal Coorelation")
```



What does the plot suggest about their (marginal) correlation? There seems to be no trend in the plot which suggests no correlation. Does it agree with Figure 3? The results match the Figure since A and B are marginally independent and, thus, their correlation should be 0.

```r
#Coorelation Test

cor.test(DAG$A,DAG$B)
```

```
##
##  Pearson's product-moment correlation
##
## data:  DAG$A and DAG$B
## t = 0.20194, df = 298, p-value = 0.8401
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1016784  0.1247727
## sample estimates:
##        cor
## 0.01169715
```

Use the function cor.test() to test the null hypothesis of no correlation between A and B What is your conclusion?

We can not reject the hypothesis that A and B are marginally uncorrelated as it gives us a p-value of p-value = 0.8401. Therefore, A and B are not correlated.

Problem 25: Testing for partial correlation

Compute the partial correlation $\rho_{A,B|C}$ to assess the association between A and B given C as follows:

1. Linearly regress A on C (that is, with A as the response variable and C as the explanatory variable). Compute and store the residuals.

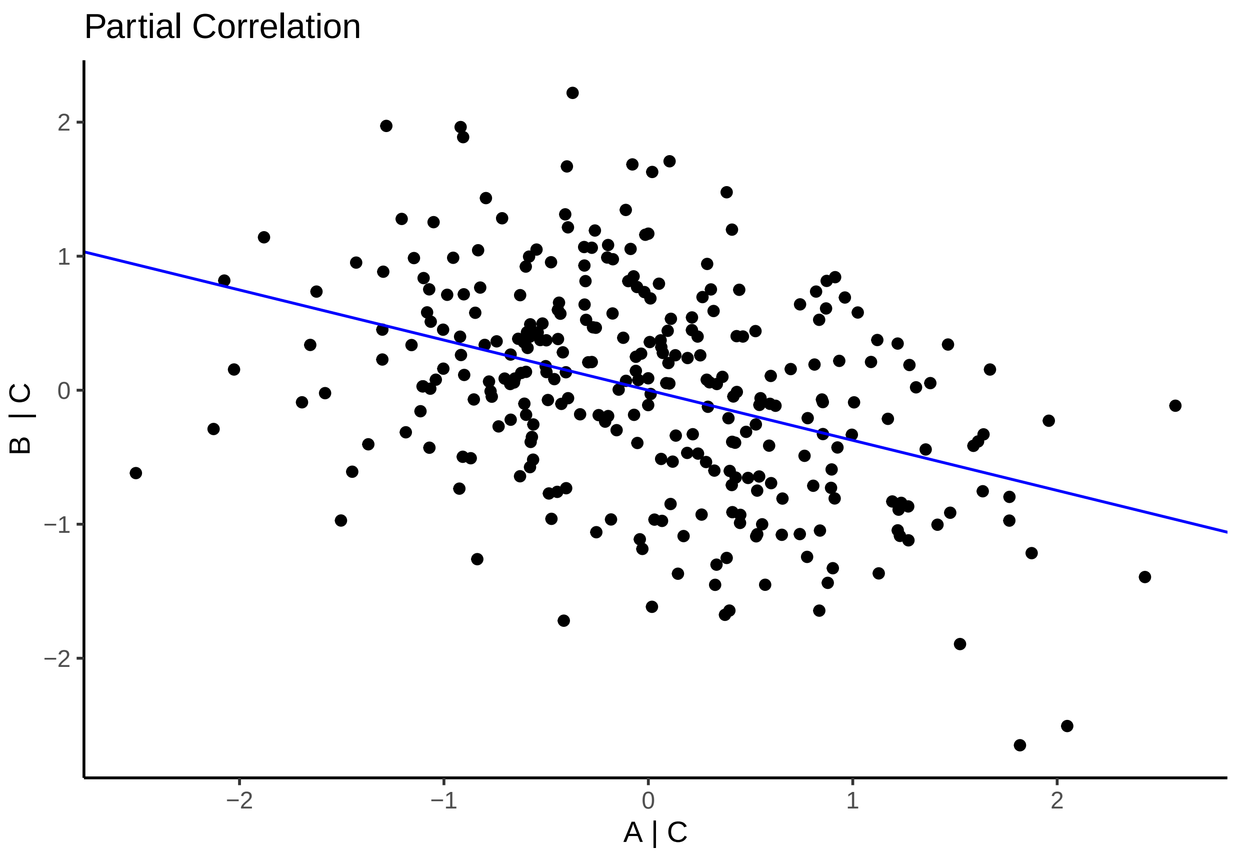
```
regA_C<-lm(A~ C,as.data.frame(DAG))
residualA_C<-residuals(regA_C)
```

2. Linearly regress B on C. Compute and store the residuals.

```
regB_C<-lm(B~ C,as.data.frame(DAG))
residualB_C<-residuals(regB_C)
```

3. Plot the residuals of A (regressed on C) against the residuals of B (regressed on C).

```
slr.coef<-coef(lm(residualB_C~ residualA_C))
qplot(residualA_C,residualB_C)+
  theme_classic()+
  geom_abline(intercept =slr.coef[1], slope = slr.coef[2],colour="blue" )+
  ggtitle("Partial Correlation ")+labs(y= "B  | C", x = "A | C")
```



What do you see? There seems to be a trend in the plot which suggests that there could be a partial correlation.

4. Use the function cor.test() to test the null hypothesis of no correlation between the residualsof A (regressed on C) and the residuals of B (regressed on C)

```
#Coorelation Test

cor.test(residualA_C,residualB_C)
```

```
##
##  Pearson's product-moment correlation
##
## data:  residualA_C and residualB_C
## t = -7.5173, df = 298, p-value = 6.6e-13
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4903245 -0.2995546
## sample estimates:
##        cor
## -0.3992521
```

What is your conclusion? With the p-value of 6.6e-13 we can reject the Null Hypothesis which states that there is no partial correlation. We can say that A and B are not conditionally independent given C. Does this agree with your expectation based on the underlying DAG in Figure 3? It agrees with the Figure, as it corresponds to the /explaining away/ scenario in which A and B are parent nodes of C and, thus, given C, they cannot be independent.

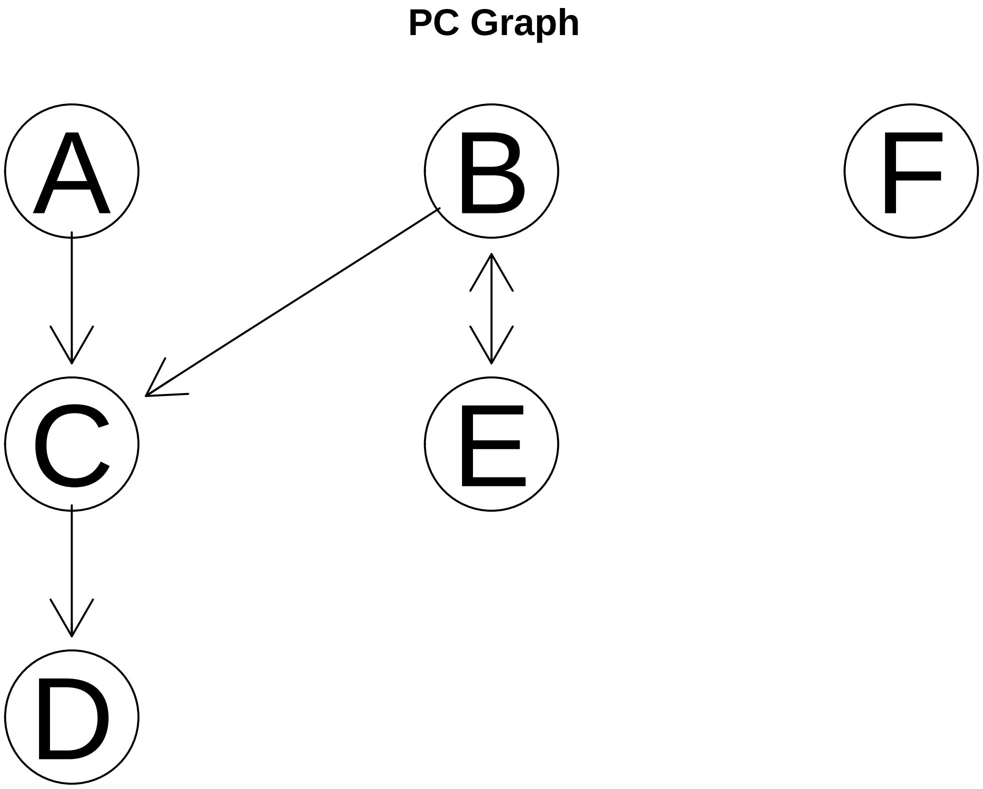
Problem 26: Running the PC algorithm

Use the function pc() to run the PC algorithm on the data in MVN DAG.rds, and plot the result.

```
sSt <- list(C = cor(DAG),n= nrow(DAG))

PCrun<-pc(suffStat=sSt,indepTest = gaussCItest,labels = colnames(DAG),
          alpha = 0.01)

plot(PCrun,main="PC Graph")
```



Does the algorithm successfully learn the structure of the data-generating graph in Figure 3? Yes the PC algorithm has successfully estimated the correct graph when α =0.01, because it matches both the skeleton and all the colliders. Note that the arrow direction of B & E can not be estimated as it is indistinguishable from that of the true graph: the underlying skeleton is identical for any possible edge orientation that does not form a collider. How is the result affected by the significance level α for the conditional independence tests? The result change with different α values. The higher alpha is, more edges can get accepted so the graph becomes more dense.

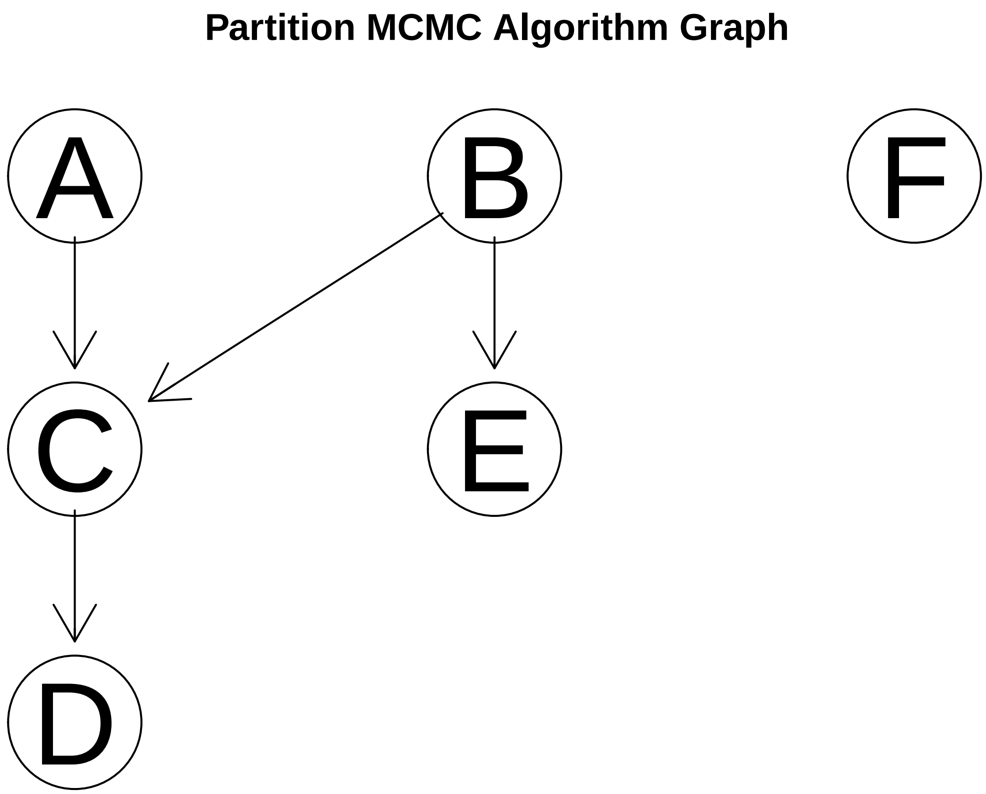
Problem 27: Running the partition MCMC algorithm

Run the partition MCMC algorithm using the function partitionMCMC() and plot the result.

```r
param<-scoreparameters(scoretype="bge", DAG)

partmcmc<-partitionMCMC(param)

plot(m2graph(partmcmc$DAG), main="Partition MCMC Algorithm Graph")
```



How is the result affected by the hyper-parameter α_μ? The colliders are preserved for every value of α_μ while the rest of the skeleton/structure gets lost for smaller values of α_μ, where we observe a sparse network with several disconnected components.

Our intuition behinds comes from the fact that the BGe score is a decompose likelihood function that assumes that the data is normal with mean vector μ, and this vector μ has at the same time a prior with mean ν and variance $\alpha_\mu W$, where α_μ controls how narrow we believe our data do be distributed a priori. Then the algorithm, for a reason not so clear to us, scores sparse networks with higher likelihoods when the variance is lower (smaller values of α_μ). We also observe this in homework 1, where there is more edges for higher values of α_μ.