STAT7730 Final Report

Hyoin An, Yuxuan Xin, and Haozhen Yu

April 24, 2021

1 Introduction

The purpose of this project is to review the paper titled "Bayesian inference of causal effects from observational data in Gaussian graphical models" by Castelletti and Consonni [2].

The Bayesian approach using Directed Acyclic Graphs (DAGs) is one of the most popular approaches in causal studies. Compared to the situation where both experimental and observational data are available, however, performing causal inference based on only observational data is more challenging, because the true underlying DAG is not identifiable from observational data alone. For this reason, one can aim to estimate a Markov equivalence class of graphs instead, which can be represented by a unique graph called Essential graph (EG, [1];[2]). To estimate the equivalence class of DAGs based solely on observational data, Castelletti and Consonni (2020) propose a Bayesian method which not only can learn the structure of the equivalence class of graphs but also infer causality in a way the uncertainty could be estimated.

2 Bayesian networks in Causal inference

Let $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be a DAG, where $\mathcal{V} = \{1, ..., q\}$ is a set of vertices and $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ is a set of directed edges. If there is an edge $u \to \nu$ in $\mathcal{D}$, then we say u is a parent of ν and $pa_{\mathcal{D}}(\nu) = pa(\nu)$ is a parent set, so $u \in pa(\nu)$. The family of ν in $\mathcal{D}$ is $fa_{\mathcal{D}}(\nu) = fa(\nu) = \nu \cup pa(\nu)$.

Consider $(X_1, ..., X_q)$ that respect $\mathcal{D}$. Then, we let $f(\cdot)$ denote the joint probability function of $(X_1, ..., X_q)$ and assume $f(\cdot)$ has the Markov property of the DAG. According to the notation in Pearl (2000), the do-operator $\text{do}(X_i = \widetilde{x}_i)$ denotes an intervention consisting in setting X_i to the value $\widetilde{x}_i$. Then, given the do-operator, the postintervention distribution of $(X_1, ..., X_q)$ is as follows.

$$f(x_1, ..., x_q \mid \text{do}(X_i = \widetilde{x}_i)) = \begin{cases} \prod_{j=1, j \neq i}^{q} f(x_j | \boldsymbol{x}_{pa(j)})|_{x_i = \widetilde{x}_i} & \text{if } x_i = \widetilde{x}_i \\ 0 & \text{otherwise} \end{cases}$$

Let $X_1 = Y$ to explicitly view X_1 as a response, then given the do-operator, the marginal postintervention distribution of Y is obtained as below.

$$f(y \mid \text{do}(X_i = \widetilde{x}_i)) = \int f(y \mid \widetilde{x}_i, \boldsymbol{x}_{pa(i)}) f(\boldsymbol{x}_{pa(i)}) d\boldsymbol{x}_{pa(i)}.$$

Then, the causal effect of $\text{do}(X_i = \widetilde{x}_i)$ is defined as the rate of change of the expectation of postintervention of Y and is denoted by γ_i, that is:

$$\gamma_i = \frac{\partial}{\partial x} \mathbb{E}(Y \mid \text{do}(X_i = \widetilde{x}_i))|_{x_i = \widetilde{x}_i}.$$

Castelletti and Consonni [2] restrict their attention to the case of Gaussian graphical models. Suppose that $\boldsymbol{x} = (X_1, ..., X_q)^T$ is normally distributed, i.e. $\boldsymbol{x}|\boldsymbol{\Sigma} \sim \mathcal{N}_q(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a symmetric positive definite covariance matrix Markov with respect to $\mathcal{D}$. The inverse of $\boldsymbol{\Sigma}$ is a precision matrix, which is denoted by $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$.

Then, under the gaussian assumption, the expectation of postintervention of Y becomes

$$\mathbb{E}(Y \mid \text{do}(X_i = \widetilde{x}_i), \boldsymbol{\Sigma})|_{x_i = \widetilde{x}_i} = \gamma_i \widetilde{x}_i + \int \boldsymbol{\gamma}_{pa(i)}^T \boldsymbol{x}_{pa(i)} d\boldsymbol{x}_{pa(i)}.$$

We can observe the relationship between the causal effect γ_i and $\boldsymbol{\Sigma}$. Note that the conditional expectation of Y

$$\mathbb{E}(Y \mid \boldsymbol{x}_{fa(i)}, \boldsymbol{\Sigma}) = [\boldsymbol{\Sigma}_{Y, fa(i)}](\boldsymbol{\Sigma}_{fa(i), fa(i)})^{-1} \boldsymbol{x}_{fa(i)}.$$

Then, since $fa(i) = i \cup pa(i)$, we have

$$\gamma_i = \left[[\boldsymbol{\Sigma}_{Y, fa(i)}](\boldsymbol{\Sigma}_{fa(i), fa(i)})^{-1} \right]_1,$$

where subscript 1 corresponds to the first entry of the vector.

3 Bayesian Inference of DAG Model Parameters

Castelletti and Consonni [2] makes Bayesian inference on $\boldsymbol{\Sigma}$ and each γ_i. Then, the choice of priors on the parameters for each model is very important. Geiger and Heckerman (G&H, [4]) propose a method to construct priors for comparison of DAG-models that ensures identical marginal likelihoods for DAGs belonging to the same equivalence class. Their method assumes two conditions, which are prior modularity and global parameter independence.

- **Prior Modularity** Given two distinct DAG models with same set of parents for vertex j, the prior for θ_j must be the same under both models, i.e. $p(\theta_j|\mathcal{D}_h) = p(\theta_j|\mathcal{D}_k)$ for any pair of DAGS $\mathcal{D}_h$ and $\mathcal{D}_k$.

- **Global Parameter Independence** For any DAG model $\mathcal{D}$, the parameters θ_j should be a priori independent, i.e. $p(\theta|\mathcal{D}) = \prod_{j=1}^q p(\theta_j|\mathcal{D})$

Based on that settings, all parameter priors are completely determined by a unique prior and all DAG-models within the same equivalence class will be scored equally. Then, following the procedure of G&H, it provides a prior for precision matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ which is based on the Cholesky parameterization of a DAG model. Then, it can make an inference on Σ by obtaining the posterior distribution on the DAG Cholesky parameters (D, L).

If we assume a Gaussian model $x|\boldsymbol{\Sigma} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, we can re-parameterize it through the structural equation model $L^\top x = \epsilon$, where L is a (q, q) lower triangular matrix of coefficients.

It implies $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{D})$, where $\mathbf{D} = diag(\sigma^2)$ and $\mathbf{\Sigma} = L^{-\top} D L^{-1} = \mathbf{\Omega}^{-1}/$ Hence, (D, L) can be represented by $\mathbf{\Sigma}$, respectively.$D_{jj} = \mathbf{\Sigma}_{jj|pa(j)},\ \ L_{\prec j]} = -\mathbf{\Sigma}_{\prec j \succ} \mathbf{\Sigma}_{\prec j]}$.

Based on the settings, we can firstly start from a prior on the unrestricted precision matrix for a complete DAG. It assumes to assign a non-informative prior $p^N(\mathbf{\Omega}) \propto |\mathbf{\Omega}|^{-1}$, then the posterior of $\mathbf{\Omega}|X$ will follow a Wishart Distribution $W_q(n+q-1, X^\top X)$. Next, it will induce a complete DAG Wishart posterior on (D, L). Through the procedure of G & H, we can recover the posterior distribution on (D, L), whose density is shown below. Then, marginal and conditional distributions of D_{jj} and $L_{\prec j]}$ can also be determined.

$$p(D, L) \propto exp[-\frac{1}{2} tr(L D^{-1} L^\top) U] \prod_{j=1}^{q} D_{jj}^{-a_j/2}$$

$$D_{jj} \sim I - Ga(\frac{a_j}{2} - \frac{|pa(j)|}{2} - 1, \frac{1}{2} U_{jj|\prec j \succ});$$

$$L_{\prec j]} | D_{jj} \sim \mathcal{N}_{pa(j)}(-U_{\prec j \succ}^{-1} U_{\prec j]}, D_{jj} U_{\prec j \succ}^{-1}).$$

Next, it can generate posterior draws of the Cholesky parameters D_{jj} and $L_{\prec j]}$,$j = 1, ..., q$, which provides posterior draws from $\mathbf{\Sigma}$. Then, one can finally obtain the posterior draws of the causal effect parameters γ_i for any given DAG model.

4 Bayesian Inference on Causal Effects

As mentioned in Section 1, Markov equivalent DAGs can be collected into equivalence classes and represented by their essential graphs. For EG model selection, Bayesian methods generate a posterior distribution on the graph space via MCMC.

We address the uncertainty that concerns the structure of data-generating mechanism using the Objective Bayes (OB) methodology developed in Castelletti et al.[3]. Let $\mathcal{S}_q$ be the set of all EGs on q nodes. Starting from an EG $\mathcal{G}$, we assign a Bernoulli-beta prior $p(\mathcal{G})$ to the adjacency matrix of the skeleton of $\mathcal{G}$, then the posterior distribution of $\mathcal{G}$ is

$$p(\mathcal{G} \mid \boldsymbol{X}) = \frac{m_{\mathcal{G}}(\boldsymbol{X}) p(\mathcal{G})}{\sum_{\mathcal{G} \in \mathcal{S}_q} m_{\mathcal{G}}(\boldsymbol{X}) p(\mathcal{G})}.$$

Next we focus on the evaluation of the causal effect of X_i on Y, i.e. γ_i. Given an equivalence class of DAGs represented by their essential graph $\mathcal{G}$, let $\{\gamma_l(\mathcal{G}); l = 1, ..., L_{\mathcal{G}}\}$ be the collection of $L_{\mathcal{G}}$ distinct causal effects of X_i on Y (dropping subscript i for simplicity). A natural choice for an overall measure of γ conditional on $\mathcal{G}$ is the average conditional causal effect

$$\gamma_{\text{avg}}(\mathcal{G}) = \frac{1}{L_{\mathcal{G}}} \sum_{l=1}^{L_{\mathcal{G}}} \gamma_l(\mathcal{G}).$$

An estimate of $\gamma_{\text{avg}}(\mathcal{G})$ is the corresponding posterior conditional expectation

$$\bar{\gamma}_{\text{avg}}(\mathcal{G}; \boldsymbol{X}) = \frac{1}{L_{\mathcal{G}}} \sum_{l=1}^{L_{\mathcal{G}}} \mathbb{E}\left\{\gamma_l(\mathcal{G}) \mid \boldsymbol{X}, \mathcal{G}\right\}.$$

The estimate $\bar{\gamma}_{\text{avg}}\left(\mathcal{G};\boldsymbol{X}\right)$ depends on the unknown EG $\mathcal{G}$. Under the Bayesian setting, we account for the model uncertainty concerning $\mathcal{G}$ through its posterior distribution on the EG space. There are two distinct strategies to deal with it.

The first strategy employs the full posterior distribution on the EG space. Our MCMC visits a collection of EGs $\{\mathcal{G}_k, k = 1, ..., K\}$, and the posterior probability of each $\mathcal{G}_k$ is approximately

$$p\left(\mathcal{G}_k \mid \boldsymbol{X}\right) \approx \frac{m_{\mathcal{G}_k}(\boldsymbol{X}) p\left(\mathcal{G}_k\right)}{\sum_{k=1}^{K} m_{\mathcal{G}_k}(\boldsymbol{X}) p\left(\mathcal{G}_k\right)}.$$

Combining the idea of Bayesian Model Averaging (BMA) and the posterior distribution on the graph space, the OB-MA estimate of γ is

$$\bar{\gamma}_{OB-MA}(\boldsymbol{X}) = \sum_{\mathcal{G}_k} \mathbb{E}\left\{\gamma_{\text{avg}}\ \left(\mathcal{G}_k\right) \mid \boldsymbol{X}, \mathcal{G}_k\right\} p\left(\mathcal{G}_k \mid \boldsymbol{X}\right).$$

The second one relies on a single summary of the posterior distribution on the EG space. We construct a graph by only including those edges with marginal posterior probability of inclusion exceeds a threshold, say, 0.5, and denote $\mathcal{G}^*$ as the resulting graph. Given $\mathcal{G}^*$, we can construct a set of distinct causal effects $\{\gamma_l(\mathcal{G}^*); l = 1, ..., L_{\mathcal{G}^*}\}$, and then obtain $\bar{\gamma}_{\text{avg}}\left(\mathcal{G}^*;\boldsymbol{X}\right)$ as discussed before. This estimate is referred to as OB-MED.

5 A Simulation study

The performances of OB-MA and OB-MED methods are evaluated via simulation. We consider different number of nodes $q \in \{5, 10, 20\}$ and sample sizes $n \in \{50, 100, 200\}$. Castelletti and Consonni [2] provide the R code for both generating observational data from DAGs and calculating the OB-MA and OB-MED estimate for the observational data. For each fixed q and n, we randomly generate 40 DAGs and then for each DAG $\mathcal{D}$, we generate n independent and identically distributed observations from the following mechanism

$$X_{i,j} = \mu_j + \sum_{k \in \text{pa}_D(j)} \beta_{k,j} X_{i,k} + \varepsilon_{i,j},$$

where $\varepsilon_{i,j} \overset{iid}{\sim} \mathcal{N}(0,1)$, and $\mu_j = 0$, with regression coefficients $\beta_{k,j} \overset{iid}{\sim} \mathcal{U}(1,2)$. For each DAG $\mathcal{D}$, its representative EG $\mathcal{G}$ is determined and a target node $i \in \{2, ..., q\}$ is randomly selected. Also, its distinct causal effects of $\text{do}(X_i = \widetilde{x}_i)$ on $Y \equiv X_1$ are summarized by $\bar{\gamma}_{true}$.

To see the performance of the proposed methods, Structural Hamming Distance (SHD) is calculated, which is a distance between the true graph and the estimated graph, and smaller SHD indicates better performance in estimating graphs. It turns out that the SHD value tends to increase for larger q for OB-MED method, which is expected since a graph gets complicated quickly as the number of nodes increases. We compare this value of OB-MED of $q = 10$ case to the PC algorithm [5] with three different significance level $\alpha \in \{0.1, 0.05, 0.01\}$, which is shown in the Table 1. The result for cases of $q = 5$ and $q = 20$ are omitted here to save the space. The performance of OB-MED is similar to or better than the performance of PC algorithms especially when n is larger.

	n = 50	n = 100	n = 200
OBMED	6.25 (2.82)	4.70 (2.54)	4.53 (2.32)
PC0.1	6.15 (2.65)	5.68 (1.46)	5.52 (1.75)
PC0.05	5.68 (2.96)	5.03 (1.44)	5.10 (1.88)
PC0.01	5.50 (1.88)	4.55 (1.74)	4.56 (1.97)

Table 1: Mean and Standard deviations of SHD ($q = 10$)

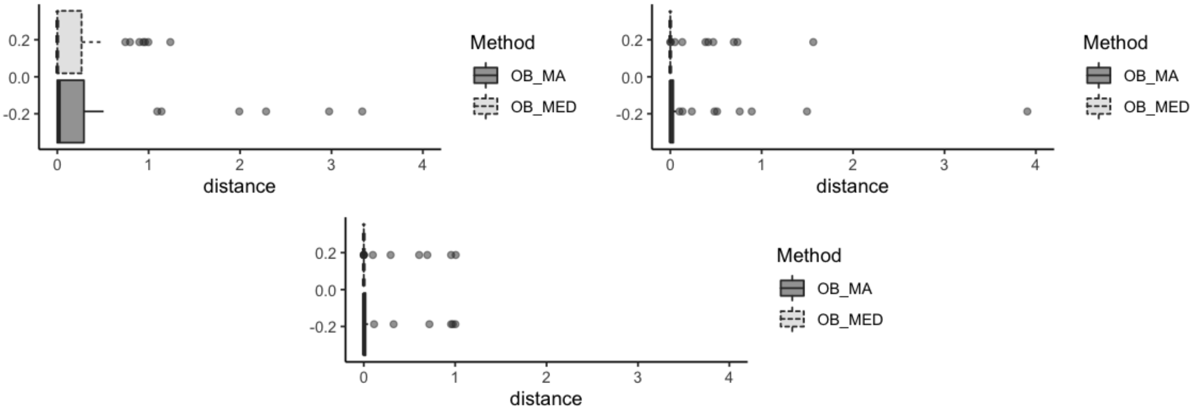

Figure 1: Absolute distance for $q = 10$, $n = 50, 100$, (top-left, top-right) and $n = 200$ (bottom)

For $q = 10$ case, the difference between true causal effect and estimated causal effect at the targeted node is calculated in Figure 1, indicating that the performance seems to be improved for larger n. On the paper [2], they compared these values to other methods in literature as well and showed that their methods perform better since they considered the model uncertainty.

6 Discussion

In this project, we review Castelletti and Consonni [2] that propose a Bayesian methodology for causal effects estimation of observational data for Gaussian graphical models. When only observational data is available, a DAG is identifiable only up to the EG, and an Objective Bayesian (OB) framework is proposed. The proposed methods estimate the causal effects using Bayesian approach and MCMC algorithm to sample the essential graph and include the model uncertainty in the model. The performance of the proposed methods were shown to better in the paper and also was improved for larger n in our simulation study. The computation of these methods, however, was a lot slower than PC algorithm due to the sampling procedure. Furthermore, from the simulation, the proposed methods seemed to perform relatively accurately to estimate the existence of the causal relationship, but the methods were not as good in estimating the magnitude of the causal effect, especially with the small n. Lastly, estimation of γ_i depends on the inverse calculation of the covariance matrix, but the inverse matrix did not appear to be stable in the simulation, which might explain the few points with large absolute value distance.

References

[1] Anderson, S. A., Madigan, D. & Perlman, M. D. (1997) A characterization of Markov equivalence classes for acyclic diagraphs, *The Annals of Statistics*, 25(2), 505-541.

[2] Castelletti, F., & Consonni, G. (2020) Bayesian inference of causal effects from observational data in Gaussian graphical models, *Biometrics*, 277(1), 136-149.

[3] Castelletti, F., Consonni, G., Della Vedova, M. & Peluso, S. (2018) Learning Markov equivalence classes of directed acyclic graphs: an objective Bayes approach, *Bayesian Analysis*, 13, 1231-1256.

[4] Geiger, D & Heckerman, D. (2002) Parameter priors for directed acyclic graphical models and the characterization of several probability distributions, *The Annals of Statistics*, 30(5), 1412-1440.

[5] Maathuis, M.H., Kalisch, M. & Bühlmann, P. (2009) Estimating high-dimensional intervention effects from observational data, *The Annals of Statistics*, 37(6A), 3133–3164.

Appendix: R codes

```
#### 1. OB methods (most codes are adopted from [2]) ####
##############################################################################

# q : number of nodes (including the response Y)
# n : number of observations

library(pcalg)
library(extraDistr)
library(mvtnorm)
library(tidyverse)
library(ggplot2)
library(gridExtra)

set.seed(123)
source("OBES_mcmc.r")
source("bayes_causal.r")
load("datan100q10p01distinct.RData")

isugMAT_ <- function(A_) {
  .Call('_gRbase_isugMAT_', PACKAGE = 'gRbase', A_)
}

shd.new <- function(m1, m2){
  shd <- 0
  s1 <- m1 + t(m1)
  s2 <- m2 + t(m2)
  s1[s1 == 2] <- 1
  s2[s2 == 2] <- 1
  ds <- s1 - s2
  ind <- which(ds > 0)
  m1[ind] <- 0
  shd <- shd + length(ind)/2
  ind <- which(ds < 0)
  m1[ind] <- m2[ind]
  shd <- shd + length(ind)/2
  d <- abs(m1 - m2)
  shd + sum((d + t(d)) > 0)/2
}

q = 5
n = 50
M    = 20
T    = 600
burn = 100
S = 50
```

```
p.star = 0.1
graphnum<-40

OBMA_Int_eff.list<-rep(0,graphnum)
OBMED_Int_eff.list<-rep(0,graphnum)
Int_eff.list<-rep(0,graphnum)
Int_eff_true.list<-rep(0,graphnum)
SHD.list<-rep(0,graphnum)

for(d in 1:graphnum){

  true.dag = randomDAG(n = q, prob = p.star)
  # plot(true.dag)
  true.dag.mat = t(as(true.dag, "matrix"))

  true.beta = true.dag.mat*matrix(runif(q*(q-1), -2, -1), q, q)
  diag(true.beta) = 1

  sigma.cond = diag(rep(1, q))
  Sigma = t(solve(true.beta))%*%sigma.cond%*%solve(true.beta)

  Omega = solve(Sigma)

  mu_0 = c(rep(0, q))
  X = rmvnorm(n, mu_0, Sigma)

  m = colMeans(X)
  s = apply(X = X, FUN = sd, MARGIN = 2)

  X = t((t(X) - m)/s)

  out = OBES_mcmc_chain(Y = X, M, T, verbose = FALSE)
  MCMC_chain = out[[1]]

  post_probs = matrix((rowMeans(MCMC_chain[,(burn + 1):(T)])),q,q)

  rownames(post_probs) = colnames(post_probs) = 1:q
  median_graph = (post_probs > 0.5)*1

  #proj_median_graph = dag2essgraph(median_graph)
  proj_median_graph = as(median_graph, "matrix")*1

  x.pos = rdunif(1,2,q)

  OBMA_output = OBMA_function(X, MCMC_chain, M, x.pos, y.pos = 1, S)

  OBMA_Int_eff.list[d]<-OBMA_output$OBMA_average
  EG_estimate = proj_median_graph

  OBMED_output = OBMED_function(X, EG_estimate, x.pos, y.pos = 1, S = S)

  OBMED_Int_eff.list[d]<-OBMED_output$OBMED_average

  fa_i = fa(x.pos, true.dag)
  Int_eff_true.list[d] = (Sigma[1, fa_i]%*%solve(Sigma[fa_i, fa_i]))[1,1]

  Distinc_graph<-distinct_causal(x.pos, y.pos = 1,  true.dag)
  Gammas = lapply(FUN = posterior_gamma, X = Distinc_graph,
  Y = X, x.pos = x.pos, y.pos = 1, S = S)
  average_Gammas = lapply(FUN = mean, X = Gammas)
  Int_eff.list[d] = mean(unlist(average_Gammas))

  SHD.list[d]<-shd.new(true.dag.mat, EG_estimate)

  print(d)

}

dist_OBMA<-abs(OBMA_Int_eff.list-Int_eff_true.list)
dist_OBMED<-abs(OBMED_Int_eff.list-Int_eff_true.list)

save.image("datan50q5p01distinct.RData")
##############################################################################


#### 2. PC Algorithm ####
##############################################################################
PC.graph <- function(niter = 40, a=0.1, q=10, n=50){
  PC <- NULL; PC.graph <- list()
  set.seed(123)
  for(i in 1:niter){
    true.dag = randomDAG(n = q, prob = p.star)
    # plot(true.dag)
    true.dag.mat = t(as(true.dag, "matrix"))

    true.beta = true.dag.mat*matrix(runif(q*(q-1), -2, -1), q, q)
    diag(true.beta) = 1

    sigma.cond = diag(rep(1, q))
    Sigma = t(solve(true.beta))%*%sigma.cond%*%solve(true.beta)

    Omega = solve(Sigma)
    mu_0 = c(rep(0, q))
    X = rmvnorm(n, mu_0, Sigma)

    m = colMeans(X)
    s = apply(X = X, FUN = sd, MARGIN = 2)

    X = t((t(X) - m)/s)

    n <- nrow(X)
    V <- as.character(1:ncol(X))

    res <- pcalg::skeleton(list(C = cor(X), n = n), alpha=a, labels = V,
                           indepTest = gaussCItest)
    PC[i] <- shd(true.dag, res@graph)
    PC.graph[[i]] <- res@graph
  }
  return(list(PC.shd=PC, graph=PC.graph))
}

PC.0.1.n50 <- PC.graph(a = 0.1, n=50)
PC.0.1.n100 <- PC.graph(a = 0.1, n=100)
PC.0.1.n200 <- PC.graph(a = 0.1, n=200)

PC.0.05.n50 <- PC.graph(a = 0.05, n=50)
PC.0.05.n100 <- PC.graph(a = 0.05, n=100)
PC.0.05.n200 <- PC.graph(a = 0.05, n=200)

PC.0.01.n50 <- PC.graph(a = 0.01, n=50)
PC.0.01.n100 <- PC.graph(a = 0.01, n=100)
PC.0.01.n200 <- PC.graph(a = 0.01, n=200)


mean(PC.0.1.n50$PC.shd); sd(PC.0.1.n50$PC.shd)
mean(PC.0.1.n100$PC.shd); sd(PC.0.1.n100$PC.shd)
mean(PC.0.1.n200$PC.shd); sd(PC.0.1.n200$PC.shd)

mean(PC.0.05.n50$PC.shd); sd(PC.0.05.n50$PC.shd)
mean(PC.0.05.n100$PC.shd); sd(PC.0.05.n100$PC.shd)
mean(PC.0.05.n200$PC.shd); sd(PC.0.05.n200$PC.shd)

mean(PC.0.01.n50$PC.shd); sd(PC.0.01.n50$PC.shd)
mean(PC.0.01.n100$PC.shd); sd(PC.0.01.n100$PC.shd)
mean(PC.0.01.n200$PC.shd); sd(PC.0.01.n200$PC.shd)

##############################################################################
```

```
#### 3. Result Plots ####
##############################################################################

library(tidyverse)
library(ggplot2)

### Create a boxplot
load("datan50q10p01distinct.RData")
dat <- as.tibble(cbind(OB_MED = dist_OBMED, OB_MA = dist_OBMA))
data<-cbind(c(mean(dist_OBMA,na.rm=T),mean(dist_OBMED,na.rm=T),
mean(SHD.list,na.rm=T)),c(sd(dist_OBMA,na.rm=T),sd(dist_OBMED,na.rm=T),
sd(SHD.list,na.rm=T)))
colnames(data)<-c("mean","Standard Deviation")
rownames(data)<-c("Distance for OBMA","Distance for OBMED","SHD")
ggplot(data=dat %>% gather(key = "Method", value = "distance")) +
  geom_boxplot(aes(distance, fill=Method,linetype=Method), alpha=0.5) +
  scale_fill_grey() + xlim(0, 4)+
  theme_classic()


load("datan100q10p01distinct.RData")
dat <- as.tibble(cbind(OB_MED = dist_OBMED, OB_MA = dist_OBMA))
data<-cbind(c(mean(dist_OBMA,na.rm=T),mean(dist_OBMED,na.rm=T),
mean(SHD.list,na.rm=T)),c(sd(dist_OBMA,na.rm=T),sd(dist_OBMED,na.rm=T),
sd(SHD.list,na.rm=T)))
colnames(data)<-c("mean","Standard Deviation")
rownames(data)<-c("Distance for OBMA","Distance for OBMED","SHD")
ggplot(data=dat %>% gather(key = "Method", value = "distance")) +
  geom_boxplot(aes(distance, fill=Method,linetype=Method), alpha=0.5) +
  scale_fill_grey() + xlim(0, 4)+
  theme_classic()

load("datan200q10p01distinct.RData")
dat <- as.tibble(cbind(OB_MED = dist_OBMED, OB_MA = dist_OBMA))
data<-cbind(c(mean(dist_OBMA,na.rm=T),mean(dist_OBMED,na.rm=T),
mean(SHD.list,na.rm=T)),c(sd(dist_OBMA,na.rm=T),sd(dist_OBMED,na.rm=T),
sd(SHD.list,na.rm=T)))
colnames(data)<-c("mean","Standard Deviation")
rownames(data)<-c("Distance for OBMA","Distance for OBMED","SHD")
ggplot(data=dat %>% gather(key = "Method", value = "distance")) +
  geom_boxplot(aes(distance, fill=Method,linetype=Method), alpha=0.5) +
  scale_fill_grey() + xlim(0, 4)+
  theme_classic()
##############################################################################
```