Probabilistic Graphical Models: Problem Set 4 (Solutions)

Svante Linusson, Liam Solus
KTH Royal Institute of Technology

28 September 2021

1. Run the PC-algorithm by hand for the following set of CI-relations on 6 variables.

$$\mathcal{CI} = \{M \perp\!\!\!\perp S \mid F, \quad M \perp\!\!\!\perp H \mid F, \quad M \perp\!\!\!\perp C \mid F, \quad M \perp\!\!\!\perp L \mid F, \\ F \perp\!\!\!\perp H \mid S, \quad F \perp\!\!\!\perp L \mid C, \quad H \perp\!\!\!\perp L \mid C, \quad S \perp\!\!\!\perp C \mid F, H, \quad S \perp\!\!\!\perp L \mid C\}$$

Assume that we know the CI relations $\mathcal{CI}$ hold in the data-generating distribution $\mathbb{P}$ and that $\mathbb{P}$ is faithful to its true causal structure $\mathcal{G}$. Can you identify $\mathcal{G}$ based on the output of the algorithm you just ran?

Solution:

To begin, we start with the complete undirected graph $\mathcal{G} = (V, E)$, where $V = \{M, S, F, H, C, L\}$, and E is the set containing an undirected edge, one for each pair of vertices in V. We first consider conditional independence tests that allow us to determine the skeleton. We are interested in testing statements of the form $i \perp\!\!\!\perp j \mid C$. We start with $|C| = 0$, so we ask if any statements of the form $i \perp\!\!\!\perp j$ hold in P. Since there are no statements of this form in $\mathcal{CI}$, we move onto $|C| = 1$.

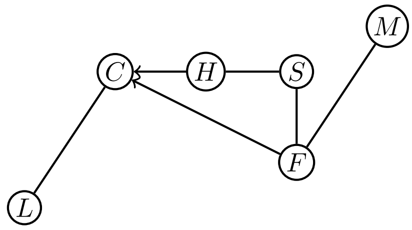
We see that there is an edge in $\mathcal{G}$: $M - S$ and both M and S are connected to F, but $M \perp\!\!\!\perp S \mid F \in \mathcal{CI}$, so we remove this edge and set $\mathcal{G} = (V, E := E \setminus \{M - S\})$. Continuing in the same fashion, we also remove the edges $M - H, M - C, M - L, F - H, F - L, H - L, S - L$, and for each edge removed we store which CI relation we used to remove it:

Edge Removed	CI relation used to remove it
$M - S$	$M \perp\!\!\!\perp S \mid F$
$M - H$	$M \perp\!\!\!\perp H \mid F$
$M - C$	$M \perp\!\!\!\perp C \mid F$
$M - L$	$M \perp\!\!\!\perp L \mid F$
$F - H$	$F \perp\!\!\!\perp H \mid S$
$F - L$	$F \perp\!\!\!\perp L \mid C$
$H - L$	$H \perp\!\!\!\perp L \mid C$
$S - L$	$S \perp\!\!\!\perp L \mid C$

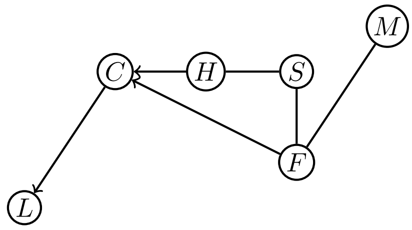
Once we see that no more edges can be removed using $|C| = 1$, we move onto $|C| = 2$, for which we find that $\mathcal{G}$ contains the edge $S - C$, with both S and C adjacent to F and H. Since $S \perp\!\!\!\perp C \mid F, H \in \mathcal{CI}$, we remove this edge resulting in $\mathcal{G} = (V, E)$ where

$$E = \{L - C, C - F, C - H, H - S, M - F\}.$$

This is our skeleton, as no further CI tests remove edges. Using the CI relation used to remove the edge $S - C$ and the CI relations in the above table we then start to add v-structures. We pick a triple path in $\mathcal{G}$ with the two end nodes not connected by an edge, such as $H - S - F$ and check in the relation used to remove the edge $H - F$, namely $F \perp\!\!\!\perp H \mid S$, if S is in the conditioning set. Since it is we do not add a v-structure along this tripe path. For another example, we take the triple path $H - C - F$. Since $H \perp\!\!\!\perp F \mid S$ was the CI relation used to remove the edge and C is not in the conditioning set, we add the v-structure $H \to C \leftarrow F$. We do this for all such triple paths to arrive at the graph



In the last step of the algorithm, we orient any arrows that if oriented the other direction would create v-structures that we did not learn in the previous step. For this graph this amounts to orienting one additional arrow. So the output of the PC algorithm is



Since this graph can be completed to multiple different DAGs without adding in new v-structures, then no we cannot learn $\mathcal{G}$ precisely.

---

2. Suppose we sample data $\mathbb{D}$ from a distribution P over four variables (X_1, X_2, X_3, X_4). Based on conditional independence tests done with $\mathbb{D}$ we estimate that P entails exactly two CI relations

$$\mathcal{CI} = \{X_1 \perp\!\!\!\perp X_3 \mid X_4, \quad X_2 \perp\!\!\!\perp X_4 \mid X_3\}.$$

Can you use the PC algorithm to learn a DAG representation of $\mathbb{P}$?

Solution:

No, we cannot. This is because the PC algorithm would learn the skeleton $\mathcal{G} = ([4], E)$, where

$$E = \{1-2, 2-3, 3-4, 1-4\},$$

where the edges $1-3$ and $2-4$ were removed by observing the CI relations $X_1 \perp\!\!\!\perp X_3 \mid X_4$ and $X_2 \perp\!\!\!\perp X_4 \mid X_3$, respectively. Hence, in the second part of the algorithm, we would consider the triple path $1-2-3$, and since the edge $1-3$ was removed by the existence of $X_1 \perp\!\!\!\perp X_3 \mid X_4$ where X_2 is not in the conditioning set, we would add a v-structure $1 \to 2 \leftarrow 3$. However, we would then also consider the triple path $2-1-4$, where the edge between 2 and 4 was removed according to $X_2 \perp\!\!\!\perp X_4 \mid X_3$, whose conditioning set does not contain X_1. Hence, we would want to also add the v-structure $2 \to 1 \leftarrow 4$, which is impossible. So the PC algorithm could not return a graph that is complete-able to a DAG representation of the data. This sort of edge like $1-2$ is often called a *conflict edge*. It's presence could represent that our data-generating distribution is not faithful to any DAG, or that our learned CI relations are incorrect, possibly due to issues such as small sample size that could lead to errors in our hypothesis tests for conditional independence.

---

3. We say that a distribution $\mathbb{P}$ satisfies **restricted faithfulness** with respect to a DAG $\mathcal{G} = ([m], E)$ if $\mathbb{P}$ satisfies the global Markov property with respect to $\mathcal{G}$ and

   - For all edges $i \to j \in E$, $X_i \not\perp\!\!\!\perp X_j \mid X_S$ for all $S \subset [m] \setminus \{i, j\}$, and
   - For all paths $\langle i, j, k\rangle$ in $\mathcal{G}$ with i and k not adjacent in $\mathcal{G}$ and for all subsets $C \subset [m] \setminus \{i, k\}$ such that i and k are d-connected given C in $\mathcal{G}$, $X_i \not\perp\!\!\!\perp X_k \mid X_C$.

   (a) Show that if $\mathbb{P}$ is faithful to $\mathcal{G}$ then it also satisfies restricted faithfulness with respect to $\mathcal{G}$.

(b) Suppose $\mathbb{P}$ is satisfies restricted faithfulness with respect to $\mathcal{G}$. Show that, when given a conditional independence test that perfectly answers any query for $\mathbb{P}$, any acyclic orientation of the output of the PC algorithm that does not introduce new v-structures will be Markov equivalent to $\mathcal{G}$.

Solution:

a) Suppose that $\mathbb{P}$ is faithful to $\mathcal{G}$. Then A and B are d-separated given C in $\mathcal{G}$ if and only if $A \perp\!\!\!\perp B \mid C$ in $\mathbb{P}$. If $i \to j \in E$, we know that i and j are not d-separated given any $C \subset [m] \setminus \{i,j\}$. Hence, $i \not\perp\!\!\!\perp j \mid C$ for all $C \subset [m] \setminus \{i,j\}$, by the faithfulness condition. Thus, $\mathbb{P}$ satisfies condition (1). If $\langle i,j,k \rangle$ is a path in $\mathcal{G}$ with i and k not adjacent, then by faithfulness, we know that $i \not\perp\!\!\!\perp k \mid C$ whenever i and k are d-connected given C in $\mathcal{G}$. Hence, condition (2) is also satisfies by $\mathbb{P}$, and we conclude that $\mathbb{P}$ satisfies restricted faithfulness.

b) First, we want to check that the PC algorithm would learn the correct skeleton of $\mathcal{G}$. We use the fact that if $\mathbb{P}$ satisfies the global Markov property with respect to $\mathcal{G}$ then two nodes i and j are adjacent in $\mathcal{G}$ if and only if i and j are d-separated given either $\mathrm{pa}_{\mathcal{G}}(i)$ or $\mathrm{pa}_{\mathcal{G}}(j)$. By assumption (1) of restricted faithfulness, there is no CI relation in our distribution that will cause us to remove an edge in E, the set of edges in $\mathcal{G}$. However, if i and j are not adjacent in $\mathcal{G}$, then since $\mathcal{G}$ satisfies the global Markov property with respect to $\mathcal{G}$, then either i and j are d-separated given $\mathrm{pa}_{\mathcal{G}}(i)$ or they are d-separated given $\mathrm{pa}_{\mathcal{G}}(j)$. Hence, for $|C| = \min\{|\mathrm{pa}_{\mathcal{G}}(i)|, |\mathrm{pa}_{\mathcal{G}}(j)|\}$, the PC algorithm will remove the edge between i and j. It follows that the PC algorithm learns the correct skeleton.

To see that it learns the correct v-structures, suppose that $i - j - k$ is triple path in the skeleton of $\mathcal{G}$ with i and k not adjacent, and that the PC algorithm removed the edge between i and k based on the learned CI relation $i \perp\!\!\!\perp k \mid C$. If there is a v-structure in $\mathcal{G}$ $i \to j \leftarrow k$, then if $j \in C$, i and k would be d-connected given C, violating condition (2) of restricted faithfulness. Hence, it must be that $j \notin C$, and so the PC algorithm would learn the presence of the v-structure. If there is no v-structure along $i - j - k$ in $\mathcal{G}$, then if $j \notin C$, there would be a d-connecting path $\langle i,j,k \rangle$ between i and k in $\mathcal{G}$ given C, which would also violate condition (2) of restricted faithfulness. Hence, it must be that $j \notin C$ in this case, and so the PC algorithm would not add a v-structure, learning the correct thing.

From this, we see that PC algorithm learns an element of the Markov equivalent class of $\mathcal{G}$ by the theorem of Verma and Pearl.

---

4. Let $\mathbb{P}$ be a distribution over $(X_1, \dots, X_m)$, and suppose that $\mathbb{P}$ is faithful to $\mathcal{G}$ and $\mathcal{H}$ is a minimal I-MAP of $\mathbb{P}$.

   (a) Show that $\mathcal{G} \leq \mathcal{H}$.

   (b) Are $\mathcal{G}$ and $\mathcal{H}$ are Markov equivalent?

Solution:

a) If $\mathbb{P}$ is faithful to $\mathcal{G}$ and $\mathcal{H}$ is a minimal I-MAP of $\mathbb{P}$, then $\mathbb{P}$ satisfies the global Markov property with respect to $\mathcal{H}$, and hence $A \perp\!\!\!\perp B \mid C$ whenever A and B are d-separated given C in $\mathcal{H}$. It follows that $\mathcal{CI}(\mathcal{H}) \subset \mathcal{CI}(\mathbb{P})$. Since $\mathbb{P}$ is faithful to $\mathcal{G}$, then $\mathcal{CI}(\mathcal{G}) = \mathcal{CI}(\mathbb{P})$, so we conclude that $\mathcal{G} \leq \mathcal{H}$.

b) It is possible that $\mathcal{CI}(\mathcal{H})$ is a strict subset of $\mathcal{CI}(\mathcal{G})$. For example, consider $\mathcal{G} = ([3], E_{\mathcal{G}})$ and $\mathcal{H} = ([3], E_{\mathcal{H}})$ where

$$E_{\mathcal{G}} = \{1 \to 2, 2 \to 3\}, \qquad E_{\mathcal{H}} = \{1 \to 2, 1 \to 3, 3 \to 2\}.$$

If $\mathbb{P}$ is faithful to $\mathcal{G}$ then $\mathcal{CI}(\mathbb{P}) = \mathcal{CI}(\mathcal{G}) = \{1 \perp\!\!\!\perp 3 \mid 2\} \supset \emptyset = \mathcal{CI}(\mathcal{H})$. Hence, $\mathcal{H}$ is an I-MAP of $\mathbb{P}$. To see that $\mathcal{H}$ is a minimal I-MAP of $\mathbb{P}$ notice that removing the edge $1 \to 3$ from $\mathcal{H}$ would make 1 and 3 d-separated given $\emptyset$, but $1 \perp\!\!\!\perp 3 \notin \mathcal{CI}(\mathbb{P})$. Similarly, removing the edge $1 \to 2$ would make 1 and 2 d-separated given 3 but $1 \perp\!\!\!\perp 2 \mid 3 \notin \mathcal{CI}(\mathbb{P})$, and removing $3 \to 2$ would make 2 and 3 d-separated given 1, but $2 \perp\!\!\!\perp 3 \mid 1 \notin \mathcal{CI}(\mathbb{P})$. Hence, $\mathcal{H}$ is a minimal I-MAP of $\mathbb{P}$, but $\mathcal{H}$ and $\mathcal{G}$ cannot be Markov equivalent as they do not have the same skeleton.

---

5. In this problem, we will set up and use an R package that allows us to apply the PC algorithm and GES to real data. The package is called `pcalg`, and you should take the following steps to install it:

   - Download R and R Studio if you have not done so already. This can be done by following the download links at `https://www.rstudio.com/`.
   - Open R Studio and type:

     ```
     install.packages("pcalg")
     ```

   - The `pcalg` package requires a number of packages that are only available through `bioconductor`. To download and install these packages, type the following into the R terminal:

     ```
     if (!requireNamespace("BiocManager", quietly = TRUE))
     install.packages("BiocManager")
     BiocManager::install(c("graph", "RBGL", "ggm", "Rgraphviz"))
     ```

   - We can then start experimenting with the `pcalg` package! It has some built in data sets we will experiment with. In the data set we will use, we have $m = 8$ continuous variables with Gaussian noise and $n = 5000$ observations (samples). To analyze this data set we first load the `pcalg` package and the data set by typing the following:

     ```
     library("pcalg")
     library("Rgraphviz")
     data("gmG")
     ```

     The object gmG contains two components, gmG$x, which contains the samples, and gmG$g, which contains the true graph used to generate the data. To see the true graph type the following. The first command sets up the viewing window so that we can display four different figures. The second command plots the true graph:

     ```
     par(mfrow=c(2,2))
     plot(gmG$g, main="True Graph")
     ```

     To start, we will use only the first 1000 samples in the data set. To take this subset type:

     ```
     dataSub<-gmG$x[c(1:1000),]
     ```

     To use the PC algorithm to try and learn this graph from the data `dataSub`, we first generate a sufficient statistic from the data that will be used by our conditional independence test:

     ```
     suffStat <- list(C = cor(dataSub), n = nrow(dataSub))
     ```

     We then run the PC algorithm on the sufficient statistic. Aside from the sufficient statistic, we also give a choice for the conditional independence test, the number of variables, and a significance level α for the test. The package pcalg includes several conditional independence tests (Gaussian, discrete and binary) for standard data types. We use the Gaussian test as we know our data is drawn from a multivariate normal model:

     ```
     pc.fit <- pc(suffStat, indepTest = gaussCItest, p=8, alpha = 0.01)
     ```

     To see the graph learned by the PC algorithm we type:

     ```
     plot(pc.fit, main="PC 1000 Samples")
     ```

     How does the result compare with the true graph? (Notice that R represents a undirected edge with a bidirect edge in its plots.) Alternatively, we can use to GES to estimate the causal structure. To do so, we first define the score (BIC) object associated to the data, and then apply the function `ges`. (Note that the character after the capital “L” when defining the object `score` is a zero and not a capital O, whereas the character after “pen” is a capital O.)

     ```
     score <- new("GaussL0penObsScore", dataSub)
     ges.fit <- ges(score)
     plot(ges.fit$essgraph, main="GES 1000 Samples")
     ```

How do the two results compare? If they are different, what do you think lead to the differences? How does each compare with the true graph? Now try running the PC algorithm on the full data set `gmG$x`. How does the result compare with the other graphs? How might you explain the similarities/differences?

Now experiment with your own data! (real or simulated!) This intro is drawn from the paper *Causal Inference Using Graphical Models with the R Package pcalg* (2012) by Kalisch et. al. For a more detailed exploration of the package, please see this paper or the R package documentation.

Solution:

Simply follow the above instructions.