CS7290 Causal Modeling in Machine Learning: Homework 2

Submission guidelines

Use a Jupyter notebook and/or R Markdown file to combine code and text answers. Compile your solution to a static PDF document(s). Submit both the compiled PDF and source files. If you use Google Collab, send the link as well as downloaded PDF and source files.

Recall the survey DAG discussed in the previous homework. Use **survey.txt** and the DAG structure to answer Question 1 and Question 2.

- **Age (A):** It is recorded as *young* (**young**) for individuals below 30 years, *adult* (**adult**) for individuals between 30 and 60 years old, and *old* (**old**) for people older than 60.
- **Sex (S):** The biological sex of individual, recorded as *male* (**M**) or *female* (**F**).
- **Education (E):** The highest level of education or training completed by the individual, recorded either *high school* (**high**) or *university degree* (**uni**).
- **Occupation (O):** It is recorded as an *employee* (**emp**) or a *self employed* (**self**) worker.
- **Residence (R):** The size of the city the individual lives in, recorded as *small* (**small**) or *big* (**big**).
- **Travel (T):** The means of transport favoured by the individual, recorded as *car* (**car**), *train* (**train**) or *other* (**other**)

We use the following directed acyclic graph (DAG) as our basis for building a model of the process that generated this data.

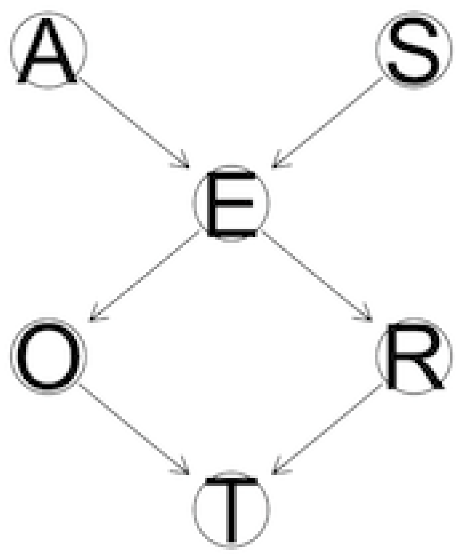


Figure 1: "survey.png"

Build the DAG and name it `net`.

First, run the following code block to create the `d_sep` function .

```
# This is the same as the bnlearn's `dsep` function but
# avoids some type checking which would throw errors in this homework.
d_sep <- bnlearn:::dseparation
```

The following code evaluates the d-separation statement "A is d-separated from E by R and T". This statement is false.

```r
d_sep(bn = net, x = 'A', y = 'E', z = c('R', 'T'))
```

```
## [1] FALSE
```

We are going to do a brute-force evaluation of every possible d-separation statement for this graph.

```r
vars <- nodes(net)
pairs <- combn(x = vars, 2, list)
arg_sets <- list()
for(pair in pairs){
  others <- setdiff(vars, pair)
  conditioning_sets <- unlist(lapply(0:4, function(.x) combn(others, .x, list)), recursive = F)
  for(set in conditioning_sets){
    args <- list(x = pair[1], y = pair[2], z = set)
    arg_sets <- c(arg_sets, list(args))
  }
}
```

For each pair of variables in the DAG, we want to evaluated if they are d-separated by the other nodes in the DAG. The code above does a bit of combinatorics to grab all pairs of variables from that DAG, and then for each pair, calculates all subsets of size 0, 1, 2, 3, and 4 of other variables that are not in that pair. List `arg_sets` contains all the pairs with all the combinations of other variables for each pair.

Question 1: Markov Property (12 points)

A joint distribution $P_{\mathbb{X}}$ is said to satisfy **Markov property** with respect to DAG G if for all disjoint node sets A, B, C satisfy $A \perp_G B|C \Rightarrow A \perp_P B|C$. In other words, every true d-separation statement in DAG G corresponds to a true conditional independence statement in joint probability distribution P. In this question, we will evaluate if Markov property holds for our survey DAG and dataset **survey.txt**. We don't have the true underlying joint probability distribution that generated this data, but we can do statistical tests for conditional independence on the data we have.

1.1

Create a new list. Iterate through the list of argument sets and evaluate if the d-separation statement is true. If a statement is true, add it to the list. Show code. What is the number of true d-separation statements? (1 point)

1.2

Consider a pair of nodes (X, Y), assuming they are not connected by a direct edge. For a set Z that makes X and Y d-separate (i.e. $X \perp Y|Z$), if removing any element from Z would break the d-separation between X and Y (i.e. X and Y becomes dependent), then we consider $X \perp Y|Z$ a nonredundant d-separation statment. Write two d-separation statements, one redundant and the other nonredundant, for a pair of nodes (A, T). (2 points)

1.3

List all the nonredundant d-separation statements for each pair of nodes that are not connnected by a directed edge. (3 points)

1.4

Based on this understanding of redundancy, how can you make this algorithm for finding true d-separation statements more efficient? (2 points)

1.5

The `ci.test` function in `bnlearn` does statistical tests for conditional independence. Using 0.05 as threshold, when $p < 0.05$, null hypothesis of conditional independence is rejected, and we conclude the pair (x,y) are dependent. When $p \geq 0.05$, null hypothesis can't we rejected, and we conclude the pair (x, y) are independent. Evaluate the global Markov property assumption by doing conditional independence test on each true d-separation statement. What is the proportion of true d-separation statements that are also true conditional independence statements? (2 points)

1.6

If we only consider nonredundant true d-separation statements, what is the proportion of them that are true conditional independence statements? (1 point)

1.7

Based on the results, how well does Markov property assumption hold up with this DAG and dataset? (1 point)

Question 2: Faithfulness (6 points)

A joint distribution P is **faithful** to DAG $\mathbb{G}$ if all disjoint node sets A, B, C satisfy $A \perp_P B|C \Rightarrow A \perp_G B|C$. In other words, every true conditional independence statement about the joint distribution corresponds to a true d-separation statement in the DAG. In this question, we will evaluate if faithfulness holds for our survey DAG and dataset **survey.txt**.

2.1

Iterate through the `arg_sets` list, run `ci.test` for each argument set in the list, creating a new list of sets where you conclude the conditional independence statement is true. What is the number of true conditional independence statements? (2 point)

2.2

Evaluate faithfulness assumption by doing d-separation test on each true conditional independence statement. What is the proportion of true conditional independence statements that are also true d-separation statements? (2 point)

2.3

If we only consider non-redundant d-separation statements, what is the proportion of true conditional independence statements that are also true nonredundant d-separation statements? (1 point)

2.4

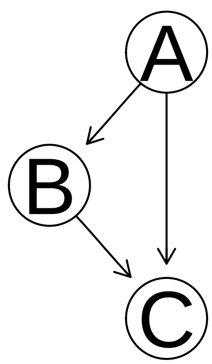
Based on the results, how well does faithfulness assumption hold up with this DAG and dataset? (1 point)

Question 3: Intervention as graph mutilation (12 points)

Run the following code to build a simple three node graph.

```r
net <- model2network('[A][B|A][C|B:A]')
alias <- c('off', 'on')
cptA <- matrix(c(0.5, 0.5), ncol=2)
dimnames(cptA) <- list(NULL, alias)
cptB <- matrix(c(.8, .2, .1, .9), ncol=2)
dimnames(cptB) <- list(B = alias, A = alias)
cptC <- matrix(c(.9, .1, .99, .01, .1, .9, .4, .6))
dim(cptC) <- c(2, 2, 2)
dimnames(cptC) <-  list(C = alias, A = alias, B = alias)
model <- custom.fit(net, list(A = cptA, B = cptB, C = cptC))
graphviz.plot(model)
```

```
## Loading required namespace: Rgraphviz
```



3.1

Given this model, use Bayes rule to calculate by hand $P(A = on|B = on, C = on)$. Show process (3 points)

3.2

Estimate this probability using *rejection sampling*. To do this, use the `rbn` function in `bnlearn` (use `?rbn` to learn about it) to create a dataframe with a large number of sampled values from the model. Remove the rows where B and C are not both 'on'. Estimate the $P(A = on|B = on, C = on)$ as the proportion of rows where A == 'on'. (Pro tip: Try the `filter` function in the package `dplyr`). (3 points)

3.3

Use `mutilated` to create a new graph under the intervention do($B = on$). Plot the new graph. (1 point)

3.4

Calculate by hand $P(A = on|\text{do}(B = on), C = on)$. Show process (3 points)

3.5

Estimate $P(A = on|\text{do}(B = on), C = on)$ using rejection sampling. (2 points)

Question 4: Implement intervention in Pyro (9 points)

4.1

Implement the model in Question 3 in `pyro`. (3 points)

4.2

Compute $P(A = on|B = on, C = on)$ using `pyro.condition` and an inference algorthm. (3 points)

4.3

Compute $P(A = on|\text{do}(B = on), C = on)$ using `pyro.do` and an inference algorthm. (3 points)