Bayesian Networks: Structure and Directionality

Adam Chen
Stevens Institute of Technology
achen19@stevens.edu

Abstract

It is known that Bayesian networks alone are an insufficient model for causality, as they merely describe the decomposition of a joint probability into conditional probabilities. They fail, in most cases, to capture directionality — a crucial aspect of causality. Furthermore, conditional probability fails to tell us about hidden confounders.

I look at the mathematical formalizations of these two shortcomings, and review the existing corpus of work on their categorization. I implement a tool to analyze when these shortcomings may be an issue and create ambiguities in Bayesian networks. The tool is used to assess potential ambiguities in existing Bayesian networks, and the implications of such ambiguities on causal inference are discussed.

1 Introduction

1.1 Equivalent Bayesian Networks

A Bayesian network is a tuple of a directed acyclic graph (DAG) and a probability distribution over variables corresponding to the vertices of the DAG, such that the probability measure is compatible with the DAG, defined thusly:

Definition 1 (Compatibility (Fong)[5])**.** *Directed acyclic graph G is compatible with probability measure (over the vertices of G) P, if*

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid pa(x_i))$$

where $pa(x_i)$ is the parents of x_i in G.

For the remainder of the paper, I may use "Bayesian network" to refer to the DAG associated with a Bayesian network.

Equivalency Intuitively, two DAGs are equivalent when they are compatible with the same probability distributions. More formally:

Definition 2 (Equivalency (Chickering)[3])**.** *G is equivalent to G' if $\forall P_G : (G, P_G)$ is a Bayesian network $\iff$ (G', P_G) is a Bayesian network. Alternatively, $\forall P_G : G$ is compatible with $P_G \iff G'$ is compatible with P_G.*

For a simple example of equivalent Bayesian networks, consider the DAG of two nodes $A \to B$. The underlying probability distribution satisfies:

$$P(A, B) = P(A) * P(B \mid A) = P(A) * \frac{P(A, B)}{P(A)} = P(A, B)$$

In other words, we have no additional constraints (information) on how A and B are distributed. Notably, this is the same constraint for the graph of $A \leftarrow B$. Thus the two DAGs are equivalent.

1.2 Motivating Example

Consider the Bayesian network shown in Figure 1, modified with two proposed changes from one presented in slides by Kleinberg[6].

We know that in simple examples, as shown before, we may be able to reverse the direction of an arrow and have an equivalent Bayesian network. Is it the case that we can reverse the arrow labeled with "?" and have an equivalent network? More generally, what are all of ways we can change the edges (be it reversing, removing, or adding edges) and have an equivalent network? I refer to this as the "Same-Variable Equivalency Problem".

We also know that many cases, there is some underlying cause (say X), that causes two other variables, thus creating an apparent correlation (in the example, smoking and lung cancer). In this example, can we add the node X and maintain compatibility with the joint probability distributions (excluding X) described by the original network? More generally, how can we add (or remove) nodes while maintaining

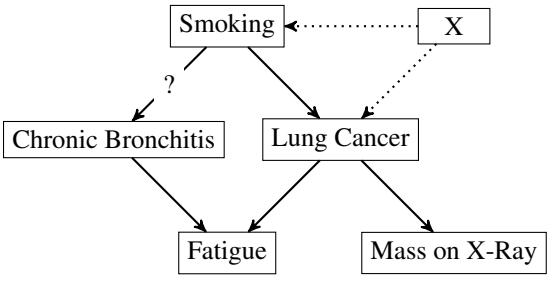


Figure 1: Example of a Bayesian network. Are we able to reverse the "?" arrow? Are we able to add the X node (and dotted edges)?

compatibility on the restricted domain? I refer to this as the "Different-Variable Equivalency Problem".

1.3 Goals

In Sections 2 and 3. I will formalize the two problems briefly described. Then, for each, I will present the theory for the resolution to the problems. The resolutions will be implemented as described in Section 4, and the implementation will be used to evaluate several Bayesian networks (causal and probabilistic) from research to analyze their validity and robustness. Finally, in Section 5, I discuss the implications of these problems on inferring causal relationships from observational data.

2 Same-Variable Equivalency Problem

Verma & Pearl[9] give a necessary and sufficient condition for testing the equivalence of Bayesian networks. Specifically, and phrased more simply by Chickering[3], two Bayesian networks are equivalent if they have the same:

1. Skeleton: set of vertices and undirected edges.
2. V-Structures: set of all unshielded colliders (i.e. triples of the form $X \to Y \leftarrow Z$ with X not adjacent to Z).

More generally, since each Bayesian network belongs to a finitely-sized equivalence class of networks completely determined by skeleton and v-structures, we may characterize all networks equivalent to a given network by those factors.

Notably, since it is the case that equivalent networks must have the same skeleton, we may limit our consideration to only how we assign directions to the edges of a network, not changing the structure or number of the edges themselves. In the case of whether a single edge is reversible (while preserving equivalence), Chickering[3] presents a necessary and sufficient condition: the edge $X \to Y$ is reversible if and only if the parents of Y are exactly X and the parents of X. (Moreover, they show that every equivalent network may be reached by a sequence of such edge reversals.)

Revisiting the Motivating Example Recall Figure 1. Observe that the sole parent of the Chronic Bronchitis node is the Smoking node, and that the Smoking node has no parents. In other words, Chickering's condition is satisfied, so we may conclude that that edge is reversible.

More generally, we may want to know what are *all* the possible assignments of directions to arrows that maintain equivalency. To do so, we note that the only v-structure in the graph is Chronic Bronchitis → Fatigue ← Lung Cancer. Thus we may erase the arrows on all other edges, and we obtain Figure 2.

However, we may not assign directions to the undirected edges however we please ($2^3 = 8$ ways). The new assignment of directions must not introduce any new v-structures, and it must not create any cycles. These constraints limit the choice of direction to four options:

1. The graph as shown in Figure 1.
2. Chronic Bronchitis → Smoking → Lung Cancer → Mass on X-Ray.

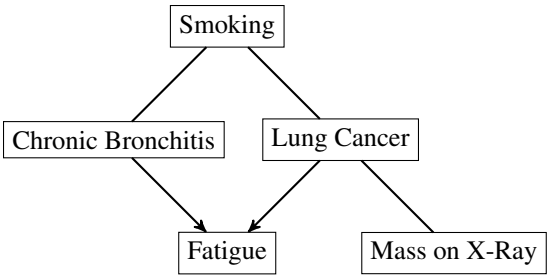


Figure 2: Replacing every undirected edge with an edge in an arbitrary direction, without introducing any new v-structures, yields an equivalent Bayesian network.

3. Mass on X-Ray → Lung Cancer → Smoking → Chronic Bronchitis.
4. Chronic Bronchitis ← Smoking ← Lung Cancer → Mass on X-Ray.

Any other choice on arrows would create a new v-structure, either on (Chronic Bronchitis, Smoking, Lung Cancer), or on (Smoking, Lung Cancer, Mass on X-Ray). In this example, creating a cycle is impossible.

3 Different-Variable Equivalency Problem

In Section 2, the case of changing the edge-structure of a DAG was investigated. Now, consider the cases where we change the number of nodes in a graph. On a practical level, node addition may be done if it is believed that an unmeasured variable is truly responsible for observed dependencies. Node removal may be done mostly for simplifying computation, but it may also be seen as the inverse process of node addition. That is, to ask the question of "when can we add a node to a DAG", one may ask the dual question of "are there DAGs while yield this one once a node is removed".

Formally, we look at two forms of node removal: marginalization and conditioning.

Node Removal To give a semi-formal treatment of the problem statement, consider that we have measure (probability) spaces represented by each variable: $X_1, \ldots, X_n$. Suppose we have a probability measure P on the product space $X_1 \times \cdots \times X_n$. We would like to, from this measure, derive a measure for the subspace $X_1 \times \cdots \times X_{k-1} \times X_{k+1} \times \cdots \times X_n$, essentially "removing" X_k. There are two canonical (and useful) ways to do so. Take $x_i \in X_i$:

Definition 3 (Marginalization (Barber)[2])**.** *The marginalization of probability measure $P(x_1, \ldots, x_n)$ over x_k is*

$$P(x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n) = \sum_{x_k} P(x_1, \ldots, x_n)$$

Definition 4 (Conditioning)**.** *The conditioning of probability measure $P(x_1, \ldots, x_n)$ on x_k is*

$$P(x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n \mid x_k)$$

$$= \frac{P(x_1, \ldots, x_n)}{P(x_k)}$$

Note that the expression in the denominator is a marginalization.

To see how these operations affect a Bayesian network, consider the simple examples of how parents and children of nodes are affected. Refer to Figure 3a. Suppose we wanted the marginalization over A. By the Bayesian network, we know

$$P(A, B, C) = P(A)P(B \mid A)P(C \mid A)$$

$$\sum_A P(A, B, C) = \sum_A P(A)P(B \mid A)P(C \mid A)$$

Which is not, in general, reducible to $P(B)P(C)$. Thus after marginalization, B and C are dependent, so there must be an edge between them. By contrast, consider the case in Figure 3b.

$$P(A, B, C) = P(A \mid B, C)P(B)P(C)$$

$$\begin{aligned} \textstyle\sum_A P(A, B, C) &= \textstyle\sum_A P(A \mid B, C)P(B)P(C) \\ &= P(B)P(C) \textstyle\sum_A P(A \mid B, C) \\ &= P(B)P(C) \end{aligned}$$

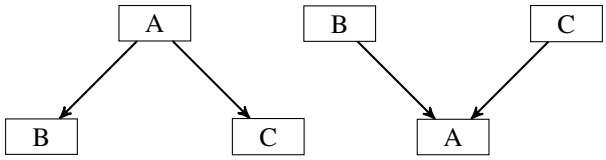
And so, B and C are independent and the Bayesian network after marginalization is the graph of B and C and no edges.

In general, it appears that marginalization does not introduce new edges between parents. It appears to extend edges from each parent to each child, and create a directed clique among the children. I could not find a source proving this claim for DAGs in particular, but processes for variable elimination (usually taken to mean marginalization) involve the moralization of the (undirected) hidden Markov model corresponding to the DAG[7].

For simple cases of conditioning, we find the opposite effect as marginalization (very much suggesting that it is a dual process). Barber[2] creates the same example graphs, and asserts that in Figure 3a, conditioning on A creates an edge between B and C, and in Figure 3b, conditioning on A results in a graph with no edges. More generally, the case of conditioning may be more well understood through the mechanism of d-separation. Recall that two nodes x and y are dependent conditioned on a node z (more generally a set of nodes) if z d-separates x and y. We can use this condition to ascertain the structure of the graph resulting from conditioning on a variable. In particular, removing a node z necessarily creates a dependency between every pair of nodes d-separated by z (i.e. there exists a path between them that does not contain a collider).

Once again, I could not locate a source proving this result, so I leave it as an unproven conjecture: conditioning on a node z yields a Bayesian network whose DAG is the original DAG, but with an additional edge between each node d-separated by z (thus forming a directed clique).

Node Addition I now arrive at the opposite problem, one that is more relevant to causal inference: the addition of new nodes. There is fundamental difficulty in that one cannot know a priori whether there is an underlying cause. First, allow me to semi-formally state the problem:



(a) B and C conditioned on A. (b) A conditioned on B and C.

Figure 3: DAGs illustrating the simplest non-trivial cases of parents and children.

Given a DAG G, we would like to:

> Characterize G' such that:
> Vertices(G') = Vertices$(G) \cup \{x\} \wedge$
> $\forall P, P' : P$ is marginalization of P' over x $\implies$
> $((G, P)$ is a Bayesian network $\iff$
> (G', P') is a Bayesian network$)$.

where x is a node not in G (mutatis mutandis for conditioning). There are clearly trivial cases: marginalizing on leaves, conditioning on root nodes.

Since the problem is phrased in terms of the result of a node removal, operations that leave clear effects on the graph, we can look for such effects. Namely, they tend to form complete bipartite sub-graphs and cliques. Elidan et. al[4] formalize this intuition; when such a Bayesian network is constructable, then indeed, the probability measures satisfy the same independence constraints, and furthermore, the new graph is minimal (in some sense).

The authors go on to describe that in inference contexts, the completely strict clique requirement is too much, and that semi-cliques are good indicators to try to infer hidden variables. They propose a heuristic-based algorithm for the detection of semi-cliques for the purposes of inference.

Examples Berkson's paradox is a well-known statistical paradox, where a sample skewed by thresholding admits an apparent (usually negative) correlation. It may be modeled in the realm of Bayesian networks by conditioning on a collider (e.g. conditioning on A in Figure 3b). In the resulting observations, we see two variables that appear to be correlated, but in actuality are independent; a study's design may just be flawed in a way (apparent or not readily apparent) such that the conditioning occurs prior to data analysis.

In the motivating example (ref. Figure 1), we wondered if a node X could be added above Smoking and Lung Cancer. It is clear that a node that is condition on could be added; this is a trivial case. The interesting case is whether there is an underlying cause (i.e. a node that we would marginalize on). It may be possible, as there is the necessary dependency between Smoking and Lung Cancer. As Smoking is a root node, there's no way to a priori (to collection of additional data) know whether there is an underlying cause; at the very least, in the view of Elidan et. al, the relative simplicity of the graph does not give us reason to believe that there should be one.

4 Evaluation

4.1 Implementation

A program for reading in directed graphs (interpreted as Bayesian networks) and performing analyses on them was written.

Bayesian Network Equivalence As per the characterization of congruence classes of Bayesian networks presented in [9; 3], testing for the equivalence of Bayesian networks was implemented via analysis on skeletons and v-structures.

Additionally, a recursive, back-tracking assignment algorithm was implemented for enumerating *all* Bayesian networks equivalent to a given network. Indeed, this tool was used to validate the explanation given for the motivating example.

Node Removal/Addition I implemented marginalizing on nodes using the conjectured rules described previously. This was done simply via transformations on the edge set.

After this, the more difficult task of inferring node additions was implemented. Brute-force detection of cliques of a given size was used. From the list of all cliques, the clique with the highest number of parents shared among all clique members was chosen. From the chosen clique, an artificial node was added, and all edges from the shared parents to clique members were removed. Edges from each shared parent to the new node and edges from the new node to each clique member were added. Note that the edges between clique members were not removed; there may often be dependencies between effects.

It must be noted that this process is not necessarily a solution to the problem as presented. Namely, if we must assume causal faithfulness, we must verify coherence with the new network with the underlying data. To satisfy faithfulness, some of the edges added in the conservative theoretical case may have to be removed. However, it is a heuristic for potential alternate DAGs that explain the data with less complexity.

As implemented, the operations of marginalization and inference are one-sided inverses in the sense that the composition of inference and marginalization is idempotent.

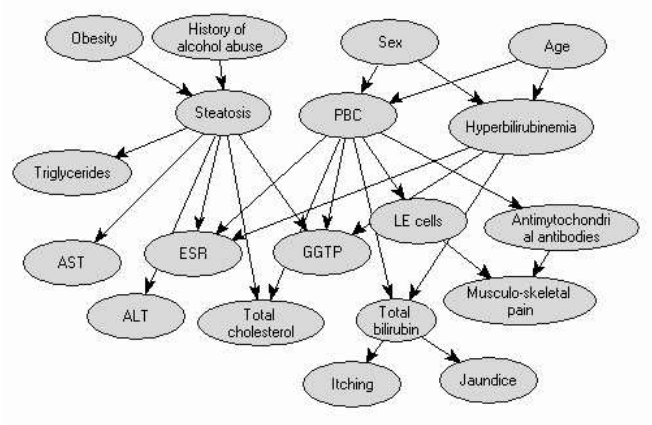
4.2 Analysis of Other Bayesian Networks

Two other Bayesian networks from published papers were used to investigate the qualities of causal as well as non-causal networks in publications. One was an explicitly causal Bayesian network for the diagnosis of liver disorders[10]. The other was a probabilistic (i.e. non-causal) network assessing factors contributing to quality of life (mental and physical) in the Reach for Life cohort[11].

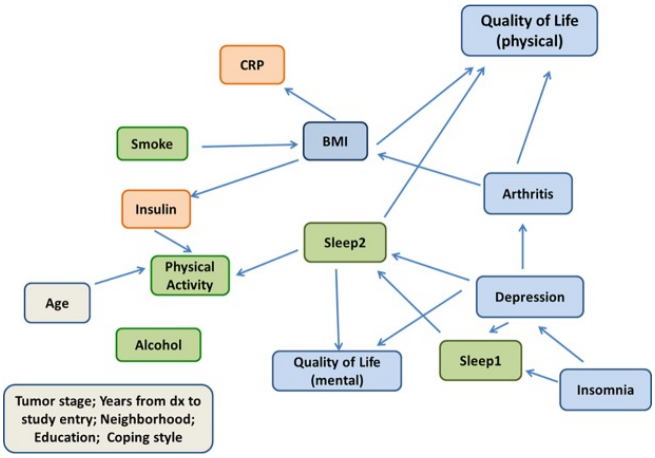
The DAGs are reproduced in Figure 4.

I found that the causal liver disorders network was uniquely determined. That is, no reversals of edges would produce an equivalent Bayesian network.

The same directionality (edge-reversal) analysis was performed on the quality of life network. A total of 12 valid graphs were obtained. This suggests that there are several relations whose directionality cannot be inferred from the data alone. At least once ambiguity arises from a root node which is the sole parent of their children, namely (Insomnia, Depression). Indeed, the authors repeat that their model is merely a probabilistic model from observations, and do not assert it as a causal model.



(a) Liver disorders Bayesian network[10].



(b) Quality of life Bayesian network[2].

Figure 4: Bayesian networks from publications used for evaluation.

Node marginalization was tested on the liver disorders network. The PBC node was marginalized. Noting that the node has 6 children, inference based on cliques was tested. Indeed, the algorithm identified a node where the PBC node was. Finally, the re-inferred node was marginalized to confirm that inference and marginalization are idempotent, and the network after the original marginalization was re-obtained. The exact graphs input to the program and obtained back are included in Appendix A. Specifically, Figure 6 shows the graph after marginalizing PBC, and Figure 7 shows the inferred node — note that the structure of the graph around `x0` is very similar to that around `pbc` in the original graph.

While the executions were not timed, it is worth noting that all operations except for inference did not take a significant amount of time. Inference (looking for cliques of order 6) on the liver disorders graph took on the order of several minutes to execute.

4.3 Availability

The code for the tool, all example inputs and outputs, and the source for this paper are available on my GitHub at https:

//github.com/Crazycolorz5/BayesianNetworkRewriting.

5 Discussion

It is known that Bayesian networks are, on their own, insufficient models of causality. Nevertheless, graphical models are useful for causal inference, and graphical models can capture many intuitions about causality (multiple causes, multiple effects, etc.). Thus when performing such inference, it is important to know how merely looking at dependence relations between data can lead us astray.

More concretely, causal inference asserts several assumptions on top of pure statistical inference[8]:

- The causal Markov condition: the set of variables has a causal structure that forms a DAG that satisfies the Bayesian conditional dependence rule.
- (Causal) faithfulness, decomposable as:

  Adjacency faithfulness: adjacent nodes are dependent given any combination of other nodes.

  Orientation faithfulness: when we have a v-structure (x, y, z), x and z are dependent given any subset of other nodes that contains y.

However, there are several problems. I have already shown that several distinct DAGs can be compatible with the same joint probability distribution. By assuming the causal Markov condition, we assert that there is some causal difference between these DAGs. It is unclear that we can, a priori, know which DAG represents the proper causal structure. Indeed, Verma & Pearl note that the recovery of a causal network from data is only modulo up to DAG equivalence[9]. It is also clear that the causal Markov condition (as presented) entails a form of completeness, e.g. common causes, which may appear as marginalized variables in a probabilistic network, are explicit. Once again, there appears to be no a priori way to know if all common causes are accounted for. I address these problem, and propose several suggestions, based on the findings in previous sections.

5.1 Measure Common Effects

The special importance of v-structures to the directness of a Bayesian network suggests that when designing a study, we should prioritize measuring variables that we believe will take this structure. Thus, the structure around these variables will be determined by the probability and so will not have an ambiguous direction. Namely, we would like to identify cases where there is a common effect (collider) of two independent causes (unshielded).

As an example, suppose we wanted to argue that smoking (S) causes lung cancer (C). We know from the example in Figure 2 that as drawn, we cannot (in the absence of outside information) determine whether it is smoking that plays a role in lung cancer, or vice versa. However, suppose we have another suspected cause of lung cancer, say asbestos exposure (A), and we believe that $S \perp A$. Then, in observation, we may find that $P(S, C, A) = P(S) * P(C \mid S, A) * P(A)$, and thus we know that the direction of those edges are forced. Then if we assume causal faithfulness, which one must to perform reasonable causal inference anyway, then we find that it must be the case that smoking causes lung cancer, not vice-versa.

5.2 The Importance of Ground Truth

An independent source of directionality can drastically narrow down the number of possible equivalent Bayesian networks. For example, in the motivating example, if we know externally that Smoking $\rightarrow$ Lung Cancer, we can eliminate options (3) and (4), as they do not obey this direction. Considering that the theoretical ceiling on direction assignments is 2^n, where n is the number of undirected edges, it is possible that each ground truth (that is not already implied by the v-structures) can reduce the possibility space exponentially.

It may also sometimes be the case that knowing one edge's direction can, through the introduction of a v-structure, allow us to deduce other edges' directions. For example, if we know that Smoking causes Lung Cancer, then it cannot be the case that Mass on X-Ray causes Lung Cancer; if it did then (Smoking, Lung Cancer, Mass on X-Ray) would be a v-structure not present in the original network. And so, it must be the case that Lung Cancer is what causes Mass on X-Ray, not vice-versa. Thus if we have a reliable knowledge base, then there may be a snowball effect where we can become more certain of other pieces of knowledge.

5.3 Directionality from Observation

It is a common assertion that one cannot argue for causality from observational studies. This is a claim that has already been addressed previously; just because a study is an RCT does not mean it can prove causality, and just because a study is observational does not mean it does not offer evidence for a causal claim. Traditional ways to argue for direction from observational studies is to use time (if A happens before B, then there is the direction of time from A to B). The well-defined directions induced by colliders suggests that they may be used to asserting directionality from pure observation. This supports that well-designed observational studies can be valid bases for arguing causality.

5.4 Heuristics for Underlying Causes

As in the clique-based approach, or the approach used by Elidan et al.[4], certain patterns of complexity (more formally phrased in the language of cliques), give us reason to test for presence of an underlying variable. Of course, it may be the case that in reality, the relations really are just complex, but experimentally, they found this heuristic approach to yield good results.

This approach appears to be consonant with the principle of parsimony (informally known as Occam's razor). Fewer premises need to hold for a simpler explanation as opposed to a one resting on many premises. Here, our assumptions include that each edge in the graph corresponds to a real, causal connection. The principle of parsimony suggests that we ought to prefer an alternate, simpler graph that is still sufficient to explain our observations. Thus, the approach of looking to simplify the most complex parts of a graph seem to abide by this classical principle.

5.5 Computational Complexity

Clique (and semi-clique) detection is well-known to be NP-complete. This poses a problem if we are to use clique detection to find underlying causes. If we take a more direct interpretation that such cliques are an indicator of hidden causes, then the interpretation of the problem of NP-complete suggests that while the discovery of underlying causes is hard, verification is comparatively easy. It is interesting that this observation lines up well with current approaches to science, where the majority of work is exploratory in nature, and most work has a negative result.

At the very least, the computational difficulty justifies the use of heuristic approaches. Yet, the use of heuristics introduces another layer of uncertainty (either that we have found all underlying causes or in the validity of causes found). Analogies may be drawn to uncertainties in the philosophy of science — of knowing completeness of an explanation or a priori knowing that a step in our beliefs is invalid.

6 Further Work

Were I to expand on this project, I would like to prove the conjectures for how marginalization and conditioning affect a network. I found it very surprising that, while the behavior of these operations for simple cases were shown, I could not find a source for a full graph transformation rule. Out of the sources I could locate, Barber[2] went the most in-depth, but instead of fully describing the graph operations, they went on to describe ascertaining dependence via d-separation (which tells us about paths and thus only indirectly the structure). This appears to be a problem that is more applied in the area of undirected graphs and variable elimination in machine learning. Still, the mathematics should give a clear answer to how it affects the setting of Bayesian networks.

I would like to work with data, similar to how Bayes Server[1] allows for visualization of Bayesian networks. This would allow for some more practical experiments relating graph-based (as opposed to data-based) inference, as the validity of the inference techniques based on graph structure must be validated against the data. Even more so than validity, since such techniques often rely on heuristics, evaluation of their effectiveness in actual data is necessary to compare techniques' relative effectiveness.

References

[1] Bayes server. https://www.bayesserver.com/.

[2] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[3] David Maxwell Chickering. A transformational characterization of equivalent bayesian network structures, 2013.

[4] Gal Elidan, Noam Lotner, Nir Friedman, and Daphne Koller. Discovering hidden variables: A structure-based approach. 08 2001.

[5] Brendan Fong. Causal theories: A categorical perspective on bayesian networks, 2013.

[6] Samantha Kleinberg. Causal inference week 4. 2021.

[7] Daphne Koller. Graph-based perspective on variable elimination. Coursera lecture.

[8] Joseph Ramsey, Jiji Zhang, and Peter L. Spirtes. Adjacency-faithfulness and conservative causal inference, 2012.

[9] Tom S. Verma and Judea Pearl. On the equivalence of causal models, 2013.

[10] Hanna Wasyluk, Agnieszka Onisko, and Marek Druzdzel. Support of diagnosis of liver disorders based on a causal bayesian network model. *Medical science monitor : international medical journal of experimental and clinical research*, 7 Suppl 1:327–32, 06 2001.

[11] Selene Xu, Wesley Thompson, Jacqueline Kerr, Suneeta Godbole, Dorothy D. Sears, Ruth Patterson, and Loki Natarajan. Modeling interrelationships between health behaviors in overweight breast cancer survivors: Applying bayesian networks. *PloS one*, 13(9):e0202923–e0202923, Sep 2018. 30180192[pmid].

A Analyzed Graphs

```
obesity,
alcohol,
sex,
age,
steatosis,
pbc,
hyperbilirubinemia,
triglycerides,
ast,
esr,
ggtp,
le_cells,
antimytochondrial_antibodies,
alt,
cholesterol,
bilirubin,
musculo_skeletal_pain,
itching,
jaundice;
obesity->steatosis,
alcohol->steatosis,
sex->pbc,
sex->hyperbilirubinemia,
age->pbc,
age->hyperbilirubinemia,
steatosis->triglycerides,
steatosis->ast,
steatosis->alt,
steatosis->esr,
steatosis->cholesterol,
steatosis->ggtp,
pbc->esr,
pbc->cholesterol,
pbc->ggtp,
pbc->bilirubin,
pbc->le_cells,
pbc->antimytochondrial_antibodies,
hyperbilirubinemia->esr,
hyperbilirubinemia->ggtp,
hyperbilirubinemia->bilirubin,
le_cells->musculo_skeletal_pain,
antimytochondrial_antibodies->
  musculo_skeletal_pain,
bilirubin->itching,
bilirubin->jaundice
```

Figure 5: The text representation of the liver disorders network.

```
age,
alcohol,
alt,
antimytochondrial_antibodies,
ast,
bilirubin,
cholesterol,
esr,
ggtp,
hyperbilirubinemia,
itching,
jaundice,
le_cells,
musculo_skeletal_pain,
obesity,
sex,
steatosis,
triglycerides;
age -> antimytochondrial_antibodies,
age -> bilirubin,
age -> cholesterol,
age -> esr,
age -> ggtp,
age -> hyperbilirubinemia,
age -> le_cells,
alcohol -> steatosis,
antimytochondrial_antibodies -> cholesterol,
antimytochondrial_antibodies -> le_cells,
antimytochondrial_antibodies ->
  musculo_skeletal_pain,
bilirubin -> antimytochondrial_antibodies,
bilirubin -> cholesterol,
bilirubin -> esr,
bilirubin -> ggtp,
bilirubin -> itching,
bilirubin -> jaundice,
bilirubin -> le_cells,
cholesterol -> le_cells,
esr -> antimytochondrial_antibodies,
esr -> cholesterol,
esr -> ggtp,
esr -> le_cells,
ggtp -> antimytochondrial_antibodies,
ggtp -> cholesterol,
ggtp -> le_cells,
hyperbilirubinemia -> bilirubin,
hyperbilirubinemia -> esr,
hyperbilirubinemia -> ggtp,
le_cells -> musculo_skeletal_pain,
obesity -> steatosis,
sex -> antimytochondrial_antibodies,
sex -> bilirubin,
sex -> cholesterol,
sex -> esr,
sex -> ggtp,
sex -> hyperbilirubinemia,
sex -> le_cells,
steatosis -> alt,
steatosis -> ast,
steatosis -> cholesterol,
steatosis -> esr,
steatosis -> ggtp,
steatosis -> triglycerides
```

Figure 6: The liver disorders network after marginalizing pbc.

```
age,
alcohol,
alt,
antimytochondrial_antibodies,
ast,
bilirubin,
cholesterol,
esr,
ggtp,
hyperbilirubinemia,
itching,
jaundice,
le_cells,
musculo_skeletal_pain,
obesity,
sex,
steatosis,
triglycerides,
x0;
age -> hyperbilirubinemia,
age -> x0,
alcohol -> steatosis,
antimytochondrial_antibodies -> cholesterol,
antimytochondrial_antibodies -> le_cells,
antimytochondrial_antibodies ->
  musculo_skeletal_pain,
bilirubin -> antimytochondrial_antibodies,
bilirubin -> cholesterol,
bilirubin -> esr,
bilirubin -> ggtp,
bilirubin -> itching,
bilirubin -> jaundice,
bilirubin -> le_cells,
cholesterol -> le_cells,
esr -> antimytochondrial_antibodies,
esr -> cholesterol,
esr -> ggtp,
esr -> le_cells,
ggtp -> antimytochondrial_antibodies,
ggtp -> cholesterol,
ggtp -> le_cells,
hyperbilirubinemia -> bilirubin,
hyperbilirubinemia -> esr,
hyperbilirubinemia -> ggtp,
le_cells -> musculo_skeletal_pain,
obesity -> steatosis,
sex -> hyperbilirubinemia,
sex -> x0,
steatosis -> alt,
steatosis -> ast,
steatosis -> cholesterol,
steatosis -> esr,
steatosis -> ggtp,
steatosis -> triglycerides,
x0 -> antimytochondrial_antibodies,
x0 -> bilirubin,
x0 -> cholesterol,
x0 -> esr,
x0 -> ggtp,
x0 -> le_cells
```

Figure 7: The marginalized liver disorders network after inferring an underlying cause of order 6.