Data Imputation with DAG and VAE

Zhaoyue Wang[1] , **Yue Wu**[1]
[1]McGill University
{zhaoyue.wang, yue.wu6}@mail.mcgill.ca

Abstract

As machine learning models getting more advanced and can produce more accurate predictions, it is increasingly crucial for the underlying dataset to be of high quality. In real life situations, datasets may have missing values, and this can cause problems for many machine learning algorithms. Commonly, these observations are either discarded, or have the missing values filled in with the mean of mode of the feature. We proposed a VAE based data imputation model that can recover the data by find the relationship between this feature and other features. More specifically, we perform iterative missing data imputation and causal structure discovery. Our model is light-weighted and numerically practical. Experiments on synthetic data shows that our model perform exceedingly well when the percentage of missing data is less than 10%.

1 Introduction

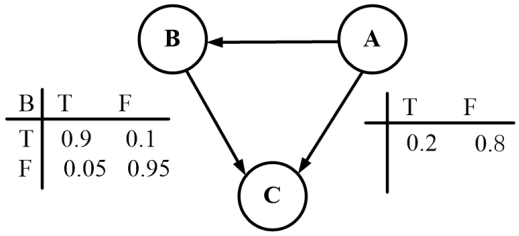
Machine learning is the most popular research direction of Artificial Intelligence in recent years, and there are already very detailed and in-depth research results. However, one limitation is that they rely heavily on large amounts of data, the common challenge faced by many machine learning models. The quantity and quality of training data directly influence the efficiency of machine learning models, but in practice, missing data is an common issue, so how to deal with those missing values in the training set is of great importance.

Moreover, in the medical field, there exits a high possibility of incomplete medical records in cancer treatment datasets, which impedes the research of cancer. Having a more integral data set and knowing the causal relation between feature would greatly aid cancer detection and diagnosis[Carter *et al.*, 2020]. In order to address this concern, we hope to investigate methods that enhance cancer detection.

In this project, we proposed to complete the missing data using some causal inference knowledge and machine learning models, compare the accuracy of different methods and investigate whether those completed data could improve the performance of various machine learning models.

The framework for causal inference is also known as the Rubin causal model (RCM) or Neyman–Rubin causal model, coined by Paul W. Holland [Holland, 1986]. This approach to statistical analysis of cause and effect is based on the concept of potential outcomes [Rubin, 2010]. It analyzes the response of an effect variable when a cause of the effect variable is changed. The change can be time-varying [Hernán and Robins, 2015]. Causal Inference stems from inquires in biology and social sciences [Pearl, 2009] that emphasize causality as opposed to correlation or distributional information. For example, the study of pharmacology is concerned with finding *why* a drug is efficient in one population over the other, rather than simply whether *it works*.

We intend to explore a less discussed application of causal inference. Real-world data we feed in to the machine learning model may involve noise or missing data. This decrease the stability and robustness of the performance of the model. Causal inference explicitly overcomes this problem by considering what might have happened when faced with a lack of information. Ultimately, this means we can utilize causal inference to make our ML models more robust and generalized.[Pandya, 2020]



B	T	F
T	0.9	0.1
F	0.05	0.95

T	F
0.2	0.8

C

B	A	T	F
T	T	0.95	0.05
F	T	0.2	0.8
T	F	0.25	0.75
F	F	0.01	0.99

Figure 1: Simple graph representation of causal relation between variables using a Bayesian network[1]

[1]A = radius (T: radius $\geqslant x$, F: radius $< x$)
B = area (T: area $\geqslant y$, F: area $< y$)
C= diagnosis (T: Malignant, F = benign).

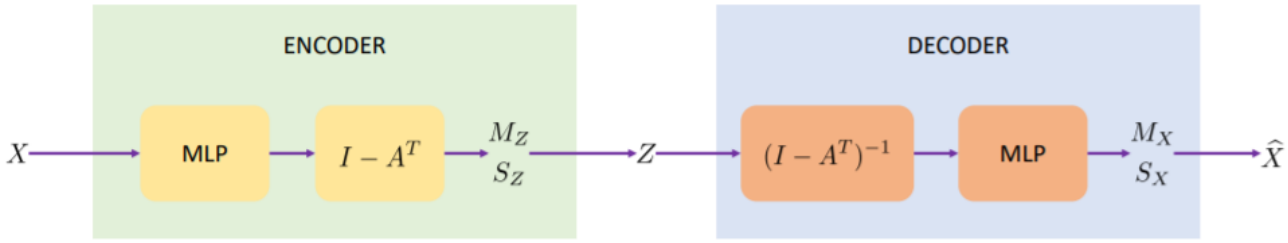


Figure 2: Architecture of neural network for causal discovery

2 Background

2.1 Causal Inference

Causality describes the relationship between cause and effect. Given cause C and effect E, if the conditional probability $P(E|C)$ is higher than the absolute probability $P(E)$, C increases the probability of E.

The presentation of causal relation can take the form of a graph, in particular, a directed acyclic graph (DAG) or a bayesian network. Figure 1 is a simplified demonstrated inspired from the Wiscosin Breast Cancer dataset [Dua and Graff, 2017].

The graph captures the causal relation (represented by conditional probabilities) between three variables: radius, area and diagnosis. An arrow from A to B in the graph indicates that A is the cause while B is the effect.

we want to treat causal discovery and data imputation as two independent procedures, so we focus on causal discovery on complete dataset. Existing model includes [Chickering, 2002] that uses Markov properties of DAGs to exploit the inner relation between features and determine the Markov equivalence class. These approaches are typically constraints-based, score-based or a hybrid of those two. The most famous constraint-based approach is the PC algorithm [Spirtes *et al.*, 2000], and there are several approaches based on it, such as Really Fast Causal Inference (RFCI) [Colombo *et al.*, 2011] and multi-core PCs [Duy Le *et al.*, 2015]. Examples of score-based approach include Greedy Equivalence Search (GES) [Nandy *et al.*, 2015] with high-dimensional consistency. More recent approaches are [Zheng *et al.*, 2018] and [Yu *et al.*, 2019a]. They will be discussed in more detail later on this paper.

2.2 Data imputation

Learning causal relationship between features of data allow us to recover the missing data.

There are 3 types of missing data.

Missing completely at random (MCAR) The missing data is independent of the observed and unobserved data. MCAR data can be considered a simple random selection of the complete data set. For example, data transmitted through the internet could be lost due to data corruption or packet lost due to internet interruption. It does not introduce bias. In general, it is more preferable to encounter compare to the other 2 types.

Missing at random (MAR) The missing data is systematically related to the observed but not the unobserved data. For example, a 50 questioned self-evaluative questionnaire about patient's health status for both male and female participants may receive more completed of questions from female then male, if male participants are less likely to answer questions regarding their personal health condition. In this case, the level of completion of the questionnaire is directly related to the sex of the participant. Moreover, the related feature should be one that can be fully observed (like their sex) instead of unobserved (like their self-identified gender). However, even if the complete case analysis is biased, proper accounting for the known factors can produce unbiased results in analysis.

Missing not at random (MNAR) The missing data is systematically related to the unobserved data. Extending from the previous example, the data is MNAR if participants who self identify as transgender is also less likely to complete the questionnaire. In this case, because the reason for missing data is not measured, it is more likely for the completed data to contain bias.

Previous data imputation methods includes matrix factorization [Dean and Varshney, 2021] and multi-objective algorithm [Khorshidi *et al.*, 2020]. More recently, [Wang *et al.*, 2020] uses GAN to perform data imputation and causal discovery. We propose a more light-weighted solution using the structure discovery method in[Yu *et al.*, 2019b].

3 Methodology

On a high level, our model first perform causal discovery on incomplete observed data $\overline{X}$. After a certain threshold,

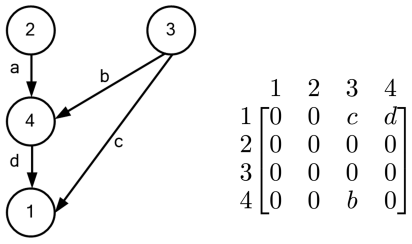


Figure 3: Causal graph (DAG) is represented as an adjacency matrix A

Algorithm 1 Iterative algorithm

Input: $\overline{X}$: Incomplete data matrix, K: Number of pre-trained iterations
Output: $\overline{X'}$: Imputed data matrix

Let Training$_D ata$ =$\overline{X}$ with all entries with missing value deleted
for i $\leqslant K$ **do**
  Perform causal discovery using Training_Data
end for
Let G = causal graph extracted from VAE
Training_Data=$Imputation(\overline{X}$,$G)$
while True **do**
  if some stopping criteria satisfied **then**
    break
  else
    Perform causal discovery using Training_Data
    Let G = causal graph extracted from VAE
    Training_Data=$Imputation(\overline{X}$,$G)$
  end if
end while
return $\overline{X'}$: Imputed Data Matrix

when we believe the causal graph should embody the relationship between the variables to a certain level, we will use this graph to perform data imputation to find imputed data $\overline{X'}$. Then we perform causal discovery on this new data $\overline{X'}$ and repeat the procedure. In the scope of this paper, we will only focus on cases where all variables have linear relation. Although non-linear cases are possible, linear cases are mathematically simpler and easier to implement. The pseudo-code for the whole iterative algorithm could be found in Algorithm 2.

3.1 Causal Discovery

The causal discovery part is based on the DAG-GNN algorithm [Yu *et al.*, 2019a], which uses the knowledge of the structural equation model (SEM) and Variational autoencoder (VAE). Figure 2 shows the general architecture of the causal discovery procedure.

Structural Equation Model (SEM)
Let $X \in \mathbb{R}^{m\times d}$, be a sample of *m* variables. Each row of X represents one variable, which is a d-dimensional vector. Then in linear SEM, we can write the equation as

$$X = A^T X + Z, \tag{1}$$

Where $Z \in \mathbb{R}^{m\times d}$, the noise matrix, indicates external effects including measurement error and $A \in \mathbb{R}^{m\times m}$, is the weighted adjacency matrix of its corresponding DAG (see figure 3). Every entry $a_{ij} \in A$, corresponds the causal weight of node *i* to node *j*, that is, the causal influence of variable j to variable i. Entries on the diagonal must be zero, since a variable cannot cause itself; if $a_{ij} = 0$ for some $i \neq j$, then there does not exist a direct causal effect between *i* and *j*. If all nodes are sorted by topological order, A should be strictly upper triangular; but here, this constraint is unnecessary in our neural network.

Then this equation can be transformed into two equations:

$$X = (I - A^T)^{-1} Z, \tag{2}$$

and

$$Z = (I - A^T)X. \tag{3}$$

The generalized form of equation is $X = f_A(Z)$, which almost all GNNs can be rewritten to [Bruna *et al.*, 2014; Chen *et al.*, 2018]. This can be considered as a function taking Z as input and returning X, through some unknown processes. Similarly, Z can also be considered as a function of X, and we can transform this procedure into a VAE.

Note that this procedure could be extended to nonlinear case, with some function f applied. Then equation (2) and (3) become

$$X = f_1((I - A^T)^{-1} Z) \tag{4}$$

and

$$Z = (I - A^T) f_2(X) \tag{5}$$

respectively. The two functions f_1 and f_2 will be learned by the Multilayer perceptron (MLP) in figure 2

Variational Autoencoder (VAE)
A variational autoencoder (VAE) consists of three parts: an encoder (or inference model), a decoder (or generative model) and a loss function kingma2014autoencoding. The encoder compresses the data X to the latent space Z (using $p_\theta(Z|X)$, which is intractable, so we use $q_\phi(Z|X)$ to approximate it), and the decoder reconstructs that same data $\hat{X}$ from the latent space (using $p_\theta(Z|X)$).

Given some distribution Z of and samples $X_1, ..., X_n$, one may train the VAE by maximizing the log-evidence

$$\frac{1}{n}\sum_{k=1}^{n} logp(X_K) = \frac{1}{n}\sum_{k=1}^{n} log \int p(X_k|Z)p(Z)dZ, \tag{6}$$

which is intractable, so we refer to evidence lower bound (ELBO) instead. We have:

$$L_{ELBO} = \frac{1}{n}\sum_{k=1}^{n} L_{k,ELBO}, \tag{7}$$

where

$$\begin{aligned} L_{k,ELBO} =& E_{q(Z|X_k)}[logp(X_k|Z)] \\ & - D_{KL}(q(Z|X_k)||p(Z)). \end{aligned} \tag{8}$$

The first term is the reconstruction error, which measures how far the reconstructed $\hat{X}$ is from the origin data X (i.e. how much information is lost during the approximation). The second term is the regularisation term using the Kullback-Leibler (KL) divergence, a measurement of the distance between two distributions. Here, we hope the distribution of latent encoding Z approach the Standard Normal distribution ($\mathcal{MN}(0, I, I)$), otherwise we give some penalty. Note that without this term, the autoencoder is solely trained with as few loss as possible, ignoring how the latent space is organised, which will lead to overfitting problem.

Acyclic Constraint

Later we will need to topologically sort the adjacent matrix A, which requires A to be acyclic. Let's use some properties of adjacency matrix outlined and proved in [Duncan, 2004].

Theorem 1 () *The (i, j)th entry B^k_{ij} of B^k, where B is the binary adjacency matrix of graph G, counts the number of walks of length k starting from node i to node j.*

In other words, if any diagonal element of B^k is positive, a cycle exists. By this theorem, $B \in \{0, 1\}^d$ is DAG if and only if:

$$tr(I - B)^{-1} = tr \sum_{k=0}^{\infty} B^k = d \tag{9}$$

But even if we take the finite series $tr \sum_{k=1}^{d} B^k$ of the infinite series $tr \sum_{k=0}^{\infty} B^k$, it is still impractical for numerical reasons. Even for small value of d, computing B^k may exceed machine precision.

Moreover, our adjacency matrix contains negative and positive weights. So we want to find a acyclic constraint that is both practical and can work with matrix of negative and positive integers. Let $\circ$ be the Hadamard product, the element-wise multiplication, and so $A \circ A$ satisfies non-negativity. [Zheng *et al.*, 2018] purposed the constraint $h(A) = tr[e^{(A \circ A)}] - m = 0$, where e^A is the matrix exponential of A. Notice that since our adjacency matrix is not binary, $h(W') > h(W)$ either a) W' have more cycles or b) W''s cycle are more heavily weighted. However, [Yu *et al.*, 2019a] points out the matrix exponential involved in this formulation may not be available in all deep learning platforms. They purpose an alternative constraint that is more practical: $trace((I + \alpha A \circ A)^m) - m = 0$, for any $\alpha > 0$. Altough $(I+\alpha B)^m$ can also be numerically difficult to compute when the eigenvalues of B is large, choosing the parameter α carefully can make the computation easier. So finally, we use above equation as the equality constraint when maximizing ELBO. We add the following to the learning problem:

$$h(A) = tr[(I + \alpha A \circ A)^m] - m, \alpha > 0 \tag{10}$$

However, the down side of mot using exponential of matrix is that it does not completely guarantee acyclicity. In our experiments with data imputation, we found out that even when the hyperparameters remain the same, occasionally the algorithm halts because one of the causal graph turned out to be cyclic. But when we only want to find a causal graph, we never encounter this issue. This lead us to suspect that when we iteratively perform causal discovery and data imputation, because the data imputation changes the dataset, it is more likely that a causal graph containing a cycle but still satisfy our constraint.

Training

Based on the above sections, our goal is to

$$\begin{aligned} \max_{A,\theta} \quad & f(A, \theta) \equiv L_{ELBO} \\ s.t. \quad & h(A) \equiv trace((I + \alpha A \circ A)^m) - m = 0, \end{aligned} \tag{11}$$

where the adjacency matrix A and all the parameters θ in the network are unknown. Then applied the augmented Lagrange trick mentioned in [Yu *et al.*, 2019a], we will be able to transform this learning problem to a VAE.

3.2 Imputation

We recover the missing data using our causal relation graph as adjacency matrix. We first use topological sort to find the order of appearance of the variables. In figure 3, variable 2($V2$) and $V3$ appears before $V1$ and $V4$, while $V4$ appears before $V1$. Since $V2$ and $V3$ do not have any relationship with each other, they are seen as equivalent. Moreover, because we do not make a distinction, in the strict definition, between the cause and effect, we can recover the cause from the effect. For example, let a dataset follow exactly the relationship in figure3. Let there be some row in $\overline{X}$ missing the value for V4 but not the rest (we assume that there will be only one missing value in each observation.) We can recover V4 by finding the values of $V2$ and $V3$ and their corresponding weights a and b. Let v_j denotes the value of variable V_j: $v_4 = a \cdot v_2 + b \cdot v_3$. We can find a and b in the row 4, corresponding to $V4$ in the adjacency matrix A. In general, for variable V_child with parents, let $\mathcal{S}_child$ be the set of its parent variable(s) V_j.

$$v_{child} = \sum_{V_j \in \mathcal{S}} A_{child,j} \cdot v_j \tag{12}$$

If $V2$ from figure 3 is missing in one of the observation, we can find its child in G. Any of the child is good. In this case, we find $V4$. Then we can find the parents of $V4$ and the corresponding weights a and b. We can recover V2: $v2 = \frac{v_4 - b \cdot v_3}{a}$. In general, for variable V_parent with no parents but has at least one child, let its variable(s) V_child be any one of its child. Let $\mathcal{S}_c$ be the set of parents of V_child.

$$v_{parent} = \frac{\sum_{V_j \in \mathcal{S} \setminus V_{parent}} A_{child,j} \cdot v_j}{A_{child,parent}} \tag{13}$$

We can apply (12) and (13) to all of the missing variables where they either have at least one parent variable or one child variable. If a variable is a stand-alone variable, that means its value will not affect or be affected by any other variables. We can assign it a random number. Our data imputation can be formulated to the following algorithm:

4 Experiment

4.1 Performance Comparison

Baseline Algorithms

Based on two causal discovery algorithms DAG-GNN [Yu *et al.*, 2019a] (score-based approach) and max-min parents-children-addictive noise model (MMPC) [Cai *et al.*, 2018] (hybrid approach), six algorithms are built to be our comparison baselines:

LD-DAG Delete all entries with missing values and then perform causal discovery with GNN.

GAN-DAG Perform imputation first using generative adversarial network (GAN) and then use the imputation results for causal discovery with GNN.

Algorithm 2 Imputation

Input: $\overline{X}$: Incomplete data matrix , G: Acyclic causal graph
Output: $\overline{X'}$: Imputed data matrix

Let Ordered_Vertices= $topological\_sort(G)$.
for vertex in Ordered_Vertices **do**
  Parents= $find\_parents\_of$(vertex)
  if Parents==NULL **then**
    Children= $find\_children\_of$(vertex)
    if Children==NULL **then**
      assign some random noise to it
    else
      Assign value using formula (13)
    end if
  else
    Assign value using formula (12)
  end if
end for
return $\overline{X'}$: Imputed Data Matrix

ICL Iteratively perform imputation using GAN and causal discovery with GNN.

LD-MMPC Delete all entries with missing values and then perform causal discovery using MMPC.

GAN-MMPC Perform imputation first using generative adversarial network (GAN) and then use the imputation results for causal discovery using MMPC.

MC-MMPC Iteratively perform imputation using GAN and causal discovery using MMPC.

Parameters Setting

We use synthetic data to ensure the access to the ground truth graph G. G is random sampled from the Erdős–Rényi (ER) model with variable size and graph degree equal to 30 and 2 respectively. After the generation of G, we random generate the observation data set $\overline{X}$ with sample size equal to 500, by setting the SEM type to be lineargauss. (i.e. the causal relation between features are linear). In out synthetic dataset, we randomly (following the MCAR mechanism) delete some data entries according to proportion p ($p \in \{10\%, 30\%, 50\%\}$). We also set K = 3 for all the experiments displayed in the graphs below.

Experiment Result

We evaluate the accuracy of our model by comparing the final causal graph and the ground truth causal graph using Structural Hamming Distance (SHD). SHD represents the minimum number of edit operations (edge insertions, deletions or flips) required to transform one graph to another graph. Hence, the smaller the better. We perform experiments to investigate how the quantity of missing data affect the result of our algorithm. Figure 5 demonstrates the performance (represented by SHD) of our algorithm and other six baselines. The results are the average of twelve random trails using the parameters mentioned in the above section. Note that the results for other parameter settings are not included, but they are almost consistent with this one.

As shown in Figure 5, MMPC-based algorithms achieve the worst performance, since the quantity of causal discovery has a direct influence on the results. Comparing LD-DAG, GAN-DAG and ICL, it is not surprising that directly remove all the missing entries leads to the highest SHD value, since much information is lost during that process. Note that GAN-DAG performs imputation and causal discovery separately without any iterative steps, but still achieve a relatively high performance, which proves the GAN's excellent performance in data imputation (see discussion for more details); but ICL still achieves a smaller SHD, which implies the importance of iterative steps.

Our algorithm performs well when missing portion p is relatively small. When p= 10%, our model could even outperform ICL. However when p increase, SHD value increases rapidly. We suspect that one cause of this phenomenon is our choice of K. Recall that in our algorithm, we perform causal discovery for K iterations before we first do data imputation, and used the imputed data as the training data afterwords. So under certain condition when p is relatively large, the large portion of badly-imputed data may not be consistent with the original data and mislead the VAE to a wrong direction. For instance, at $p = 30\%$, the performance of our algorithm is even worse than LD-DAG, meaning that the data imputation after K iterations of causal discovery is even worse than no data imputation at all. However, when we test with $K = 4$ and $K = 5$, there is no significant increase in performance. This shows that our model probably cannot perform effectively when the percentage of missing data is high.

4.2 Consistency Analysis

We recorded the mean and standard deviation for a2 trails when missing proportion $p \in \{10\%, 30\%, 50\%\}$ respectively, and it can be inferred from Figure 5 that the standard deviation is very high, especially for the case when $p = 10\%$ or $p = 50\%$, which implies that our performance is not very consistent. However, we also test some complete synthetic data on the causal discovery and imputation separately and get high standard deviation only in the causal discovery part. It can be conclude that the VAE is not stable, due to the choice

Figure 4: Performance comparison (mean of 12 trails) using SHD, lower is better

Figure 5: Mean and Standard deviation for 12 trails

of stopping criteria and the probability of getting stuck in some local optimum. Some wise choice of parameters may address or assuage this problem.

5 Discussion

5.1 Conclusion

We present a VAE based model that interactively perform data imputation and causal discovery. Our model is light-weighted and numerically practical. We evaluate the result of experiments by comparing our final causal graph and the ground truth graph. We found out that our model perform exceedingly well when the percentage of missing data is less than 10%, although the performance deteriorates fast as the percentage of missing data increase.

5.2 Future Work

There are many dimensions to explore in the future.

1. We can explore other datasets. We can test on real life datasets, for example the Wisconsin Breast Cancer dataset [Dua and Graff, 2017]. This dataset have only 10 features, so it should be computationally easy. In addition, we can also test on other kinds of input such as text or image. We can use word embedding techniques to convert text to numbers. This would introduce significantly more dimensions. So it would be interesting to explore how will the constraints perform under such stress.
2. We can also explore whether our imputation have any practical benefits. We can compare our algorithm to other more naive methods of dealing with missing data, such as simply deleting the incomplete rows, using the mean/median value of the feature, using the most frequent value of the feature etc. We can use some machine learning model on the imputed dataset and see if our algorithm perform better. In theory, we expect the performance to be better, because we essentially "recovered" the missing values so it should be closer to the ground truth.
3. Finally, we can explore other methods of data imputation and causal discovery, Like those proposed in [Yu *et al.*, 2021] and [Wang *et al.*, 2020], and discovering causal structure from observational data in the presence of latent variables as investigated in [Jabbari *et al.*, 2017]. Considering the cases with latent variable allow data imputation to perform better when the missing datatype involves unobserved data (MNAR data). In general, the future direction is to explore ways to perform accurate causal discovery on datasets with more complex structures and relationships between variables.

References

[Bruna *et al.*, 2014] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs, 2014.

[Cai *et al.*, 2018] Ruichu Cai, Jie Qiao, Zhenjie Zhang, and Zhifeng Hao. Self: Structural equational likelihood framework for causal discovery. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[Carter *et al.*, 2020] Stacy M. Carter, Wendy Rogers, Khin Than Win, Helen Frazer, Bernadette Richards, and Nehmat Houssami. The ethical, legal and social implications of using artificial intelligence systems in breast cancer care. *The Breast*, 49:25–32, 2020.

[Chen *et al.*, 2018] Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: Fast learning with graph convolutional networks via importance sampling, 2018.

[Chickering, 2002] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

[Colombo *et al.*, 2011] Diego Colombo, Marloes H Maathuis, Markus Kalisch, and Thomas S Richardson. Learning high-dimensional dags with latent and selection variables. In *UAI*, page 850, 2011.

[Dean and Varshney, 2021] Rebecca Chen Dean and Lav R. Varshney. Optimal recovery of missing values for non-negative matrix factorization. *IEEE Open Journal of Signal Processing*, 2:207–216, 2021.

[Dua and Graff, 2017] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[Duncan, 2004] A. J. Duncan. Powers of the adjacency matrix and the walk matrix. 2004.

[Duy Le *et al.*, 2015] Thuc Duy Le, Tao Hoang, Jiuyong Li, Lin Liu, and Huawen Liu. A fast PC algorithm for high dimensional causal discovery with multi-core PCs. *arXiv e-prints*, page arXiv:1502.02454, February 2015.

[Hernán and Robins, 2015] Miguel A. Hernán and James M. Robins. Longitudinal causal inference. In James D. Wright, editor, *International Encyclopedia of the Social Behavioral Sciences (Second Edition)*, pages 340–344. Elsevier, Oxford, second edition edition, 2015.

[Holland, 1986] Paul W. Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986.

[Jabbari *et al.*, 2017] Fattaneh Jabbari, Joseph Ramsey, Peter Spirtes, and Gregory Cooper. *Discovery of Causal Models that Contain Latent Variables Through Bayesian Scoring of Independence Constraints*, volume 2017, pages 142–157. 09 2017.

[Khorshidi *et al.*, 2020] Hadi A. Khorshidi, Michael Kirley, and Uwe Aickelin. Machine learning with incomplete datasets using multi-objective optimization models. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.

[Nandy *et al.*, 2015] Preetam Nandy, Alain Hauser, and Marloes H. Maathuis. High-dimensional consistency in score-based and hybrid structure learning. *arXiv e-prints*, page arXiv:1507.02608, July 2015.

[Pandya, 2020] Ravi Pandya. From how to why: An overview of causal inference in machine learning, Feb 2020.

[Pearl, 2009] Judea Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146, 01 2009.

[Rubin, 2010] D.B. Rubin. Causal inference. In Penelope Peterson, Eva Baker, and Barry McGaw, editors, *International Encyclopedia of Education (Third Edition)*, pages 66–71. Elsevier, Oxford, third edition edition, 2010.

[Spirtes *et al.*, 2000] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000.

[Wang *et al.*, 2020] Yuhao Wang, Vlado Menkovski, Hao Wang, Xin Du, and Mykola Pechenizkiy. Causal discovery from incomplete data: A deep learning approach, 2020.

[Yu *et al.*, 2019a] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks, 2019.

[Yu *et al.*, 2019b] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks, 2019.

[Yu *et al.*, 2021] Yue Yu, Tian Gao, Naiyu Yin, and Qiang Ji. Dags with no curl: An efficient dag structure learning approach, 2021.

[Zheng *et al.*, 2018] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with no tears: Continuous optimization for structure learning, 2018.