Learning MeDIL Causal Models using Generative Neural Networks

Adityakrishna Satya Chivukula

München 2020

Learning MeDIL Causal Models using Generative Neural Networks

Adityakrishna Satya Chivukula

Master's Thesis
Department of Statistics
Ludwig–Maximilians–Universität
München

Authored by,
Adityakrishna Satya Chivukula

Supervisors: Prof. Dr.-Ing. Moritz Grosse-Wentrup
Prof. Dr. Bernd Bischl

Declaration of Authorship

I hereby confirm that I have written the accompanying thesis by myself, without contributions from any sources other than those cited in the text and acknowledgements. This also applies to all graphics, images and tables included in the thesis.

....
(Place and date) (Signature)

Abstract

The aim of this master's thesis is to provide a framework for the learning of a category of Functional Causal Models called MeDIL models. These models allow reasoning of the latent variable structure of a dataset by exploiting dependencies among variables in the data. Specifically, minimal MeDIL models are considered. The approach makes use of generative modelling methods using neural networks. The proposed architecture and learning objective for the generative model is called the MMDnet. The design choices and assumptions are compared with the requirements of learning the latent causal models. The novelty of the approach in put in perspective of pre-existing approaches that solve similar tasks. Finally, the framework is tested on multiple artificial datasets. The specifics of training an MMDNet is reported along with performance of the models is reported by the ability to accurately recover the true distribution of the observed variables.

Contents

List of Figures

Chapter 1

Introduction

Several questions that we seek to answer through data analysis are essentially questions of causality. In such cases, standard statistical approaches do not suffice. This is illustrated by the straightforward example of Simpson's Paradox [Pearl, 2000]. Several fields also require causal reasoning, such as economics, epidemiology, neuroscience and psychology. It is therefore often advantageous, or even necessary, to understand the causal structure and mechanisms that generate the data that we collect and analyse. To this end, there are multiple frameworks for formalising notions of causality. Some of these include Granger causality for time-series data [Granger, 1969] and the Rubin causal model and potential outcomes framework for the design and analysis of randomised controlled trials [Holland, 1986]. A framework that is showing increased use is the Pearl causal framework [Pearl and Verma, 1991]. It provides a graphical representation of variables and their corresponding probability distributions using Directed Acyclic Graphs (DAGs).

In [Markham and Grosse-Wentrup, 2019], the authors argue that it is not unreasonable to assume that, in certain application domains, none of the variables are directly causally related, but are instead related exclusively through the presence of latent variables. They further introduce a novel algorithm to identify latent causal structures that are observationally consistent with the dataset. The main focus of this thesis is to be able learn the Functional Causal Model (FCM) for the minimal Measurement Dependence Inducing Latent Causal Model (minMCM).

1.1 Related Work

Understanding the latent variables involved in the generative process allows for more powerful data analysis. Non-Linear Independent Component Analysis is one of the primary approaches in this regard [Hoyer et al., 2009]. Variational Inference is a Bayesian approach to model latent variables of a data distribution. Specifically, Variational Autoencoders [Kingma and Welling, 2014] have been used to model complex high-dimensional distributions using lower-dimensional latent representations. Modifications to the training objective can enable more robust latent representations for the generative models by encouraging independence and representational properties of the latent variables[Burgess et al., 2018]. Causal Disentanglement is another class of methods that seek to model and represent a distribution using factors of variation in the generative process [Mathieu et al., 2016]. A combination of the two approaches is shown to be equivalent under certain conditions and capable of representing Functional Causal Models [Khemakhem et al., 2020].

These methods provide varying levels of flexibility with regard to nature of the possible causal mechanism that can be modelled. They rely on restrictive assumptions, such as allowing only linear relations or having the noise terms be exclusively non-Gaussian. But more crucially, all of the methods enforce that all of the latent variables be causes for all of the observed variables. This violates the *measurement faithfulness* property of the minMCM.

The approach presented in this thesis allows the learning of a latent variable generative model that does not require full connectedness between latent and observed variables and additionally allows the functional relations to be non-linear. Roughly speaking, a generative model is trained with a similar objective as outlined in [Li et al., 2015]. These models however do not strictly correspond to a Functional Causal Model. The setup of the generative model so as to constitute a valid FCM is similar to the setup in [Goudet et al., 2017].

The remainder of this thesis is structured as follows. In Chapter 2, some fundamentals of causality and graph terminology as introduced. Further the definition, properties and identification of minMCM are elaborated upon. Chapter 3 presents background on generative modelling approaches. Chapter 4 contains the framework proposed to learn functional relations in minMCMs and experiments conducted on synthetic datasets to validate the approach. Chapter 5 concludes the work with discussion points regarding the learnability and scalability of the proposed method.

Chapter 2

Causality Background

In this section, we introduce basic graph terminology in the context of Functional Causal Models and define the MeDIL model and minimal MCM. Furthermore, an overview of the algorithm to identify the minMCM is provided and the properties of the minMCM are described.

2.1 Graph Terminology

A Graph $\mathcal{G}$ is defined as a tuple $\mathcal{G} = \langle \mathbf{V}, \mathbf{E} \rangle$, where V is the set of vertices and E is the set of edges represented as an ordered tuple $(v_i, v_j), i \neq j$ connecting any two vertices in V, and is visually simply represented with a line. An edge is called *undirected* if for any given edge $(v_i, v_j) \in E$, the edge (v_j, v_i) is also in E. Similarly, an edge (v_i, v_j) from v_i to v_j in E, also represented as $v_i \to v_j$, is called *directed* if (v_j, v_i) is not in E. A graph $\mathcal{G}$ is called directed if all the edges in E are directed edges.

A path in a graph is a sequence of edges such that the start node of an edge in this sequence is the same as the end node of the previous edge. A path is called *directed* if all the edges in the path are directed edges. A *cycle* is a directed path in a graph that starts and ends on the same vertex. A graph $\mathcal{G}$ is called *acyclic* if there are no cycles in the graph. The parents of a vertex v in $\mathbf{V}$, written as, $\mathrm{pa}(v)$, are the set of vertices k, such that there exists a directed edge (k, v) in $\mathbf{E}$. A vertex k is called an *ancestor* of vertex v if there exists a directed path from k to v. Similarly, v is called a *descendant* of k.

2.2 Causal Models

Definition 1 (Causal Model)**.** A *causal model* of a set of variables V is a DAG, in which each node corresponds to a distinct element in V and the set of edges E correspond to causal influences of elements in V on other elements in V. [Pearl and Verma, 1991]

Definition 2 (Functional Causal Model)**.** A *functional causal model* is a triple $\mathcal{M} = \langle \mathbf{V}, \mathbf{F}, \boldsymbol{\epsilon} \rangle$, where

- $\mathbf{V}$ is the set of (endogenous) random variables,
- $\mathbf{F}$ is a set of functions defining each endogenous variable V_i as a function of its direct causes (i.e., parents or $\mathrm{pa}(V_i)$) and its corresponding exogenous random variable ϵ_i, so

that for each $V_i \in \mathbf{V}$, we have $V_i := f_i(\mathrm{pa}(V_i), \epsilon_i)$. Furthermore, $\mathbf{F}$ is constrained such that no V_i is a direct cause of itself or any of its causes, removing the possibility of causal cycles.

- $\boldsymbol{\epsilon}$ defines a joint probability distribution over the exogenous (or noise) variables, with a corresponding $\epsilon_i \in \boldsymbol{\epsilon}$ for each $V_i \in \mathbf{V}$, and with ϵ_i being independent from ϵ_j for each $\epsilon_i, \epsilon_j \in \boldsymbol{\epsilon}$

The endogenous random variables induce a probability distribution P over the variables $\mathbf{V}$. A probability distribution P and DAG $\mathcal{G}$ is called *Markov compatible* if the probability distribution P, admits the factorisation $P(\mathbf{V}) = \prod_{i=0}^{n} P(x_i|pa(x_i)), \forall x_i \in \mathbf{V}$ [Pearl et al., 2016]

The set of distributions, P, that are Markov compatible with a given DAG $\mathcal{G}$ can be characterised by the set of (conditional) independencies satisfied by the distribution. This set of (conditional) independencies can be extracted from the DAG $\mathcal{G}$ using the d-separation criterion. A path p is said to be *d-separated* by a set of nodes Z if and only if, p contains, (1) a chain $i \rightarrow m \rightarrow j$ or a fork $i \leftarrow m \rightarrow j$ with $i, j, m \in V$ such that $i, j \notin Z$ and $m \in Z$ or (2) p contains a collider: $i \rightarrow m \leftarrow j$ with $m \notin Z$ and all descendants of $m \notin Z$.

The *causal Markov condition* states that, conditioned on the parents of a variable $X \in \mathbf{V}$, i.e. $pa(X) \in V$ the variable X is independent of all other variables in V except for its effects. This implies that variables that are d-separated in $\mathcal{G}$ are (conditionally) independent. The *causal faithfulness condition* states that if any two variables in V are (conditionally) independent, they are d-separated in $\mathcal{G}$. Together the causal Markov condition and the causal faithfulness condition imply that X causes Y if and only if X and Y are dependent conditioned on the set of all direct causes of X, i.e. $\mathrm{pa}(X)$[Hausman and Woodward, 1999].

Finally, one of the conditions that is assumed for some of the traditional causal modelling approaches is sufficiency. The *causal sufficiency condition* states the all variables relevant to the model are observed and that there are no unobserved common causes between variables in V that introduce spurious dependencies in P. The MeDIL model deals in the regime where this assumption is violated.

2.2.1 MeDIL Causal Model

The Measurement Dependence Inducing Latent Causal Model is causal model where all the observed variables are treated as measurement variables that are caused exclusively by latent variables. This condition is referred to as *strong insufficiency.* All dependencies that exist in the data are through a common latent cause rather than a direct causal relation between the variables. A MeDIL model is defined as follows,

Definition 3 (Measurement Dependence Inducing Latent Causal Model (MCM))**.** A graphical MCM is a DAG, given by the triple $\mathcal{G} = \langle \mathbf{L}, \mathbf{M}, \mathbf{E} \rangle$. $\mathbf{L}$ and $\mathbf{M}$ are disjoint sets of vertices, while $\mathbf{E}$ is a set of directed edges between these sets of vertices, subject to the following constraints:

1. all vertices in $\mathbf{M}$ have in-degree of at least 1 and out-degree of 0

2. all vertices in $\mathbf{L}$ have out-degree of at least 1

3. $\mathbf{E}$ contains no cycles

Further, it is shown that several MeDIL causal models may be *observationally consistent* with a given dataset. To this end, the notion of minimality is introduced. A MCM is *minimal* if it is the least expressive MCM that induces identical dependence relations in the measurement variables, and is called a minMCM (minimal MCM). The minMCM can be identified by finding the Edge Clique Covering (ECC) of the undirected dependency graph of the measurement variables. An *undirected dependency graph* (UDG) is an undirected graph of a set of variables $\mathbf{V}$ such that there exists an undirected edge (v_i, v_j) in E, if an only if v_i and v_j are probabilistically dependent. The test for independence between each pair of measurement variables is done using the distance correlation test.

Distance Correlation

Given two random variables X and Y, the distance correlation (dCor) $\mathcal{R}(X,Y)$ evaluates to zero, if and only if X and Y are independent. [Székely et al., 2007]. The distance correlation is defined as,

$$\mathcal{R}^2(\mathbf{X},\mathbf{Y}) = \frac{\mathcal{V}^2(\mathbf{X},\mathbf{Y})}{\mathcal{V}^2(\mathbf{X},\mathbf{X})\,\mathcal{V}^2(\mathbf{Y},\mathbf{Y})} \tag{2.1}$$

The estimator for distance correlation test requires the computation of distance covariance(dCov) and distance variance (dVar). The sample distance covariance is computed using centered Euclidean distance matrices. Given the samples $\mathbf{X}$ and $\mathbf{Y}$, each of size n, we compute the pairwise euclidean matrices,

$$a_{i,j} = \|X_i - X_j\| \qquad\qquad b_{i,j} = \|Y_i - Y_j\|$$

We then calculate the following centered distances,

$$A_{i,j} = a_{i,j} - \frac{1}{n}\sum_{l=1}^{n} a_{i,l} - \frac{1}{n}\sum_{k=1}^{n} a_{k,j} + \frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n} a_{k,l}$$
$$B_{i,j} = b_{i,j} - \frac{1}{n}\sum_{l=1}^{n} b_{i,l} - \frac{1}{n}\sum_{k=1}^{n} b_{k,j} + \frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n} b_{k,l}$$

The (biased) sample distance covariance is now defined as,

$$\mathcal{V}_n^2(\mathbf{X},\mathbf{Y}) = \frac{1}{n^2}\sum_{i=0}^{n}\sum_{j=0}^{n} A_{i,j}B_{i,j}$$

This can now be used to estimate the sample distance correlation value,

$$\mathcal{R}_n^2(\mathbf{X},\mathbf{Y}) = \frac{\mathcal{V}_n^2(\mathbf{X},\mathbf{Y})}{\mathcal{V}_n^2(\mathbf{X},\mathbf{X})\,\mathcal{V}_n^2(\mathbf{Y},\mathbf{Y})} \tag{2.2}$$

Chapter 3

Generative Modelling Background

In this section, some of the definitions and approaches to generative modelling are introduced. Primarily, the use of Deep Generative methods is motivated as a flexible framework for learning complex generative models. As FCMs represent the generative process of data, generative modelling approaches naturally yield themselves for the learning of such models.

3.1 Definitions and Notation

We define a dataset with $\mathbf{X} = \{\mathbf{x}^{(i)}\}, i \in \{1, \ldots, n\}$, where $\mathbf{x}^{(i)} \in \mathbb{R}^d$, n is the number of samples and d the number of dimensions. Let $p(\mathbf{x}; \boldsymbol{\theta})$ be a data generating distribution with parameters $\boldsymbol{\theta}$. We assume that $\mathbf{X}$ is an i.i.d sample drawn from $p(\mathbf{x}; \boldsymbol{\theta^*})$, which is called the true data generating distribution, with $\boldsymbol{\theta^*}$ referred to as the true parameter set.

In Bayesian and Generative modelling settings, inference is framed as estimating the joint distribution $p(\mathbf{z}, \mathbf{x})$ over latent variables z and observed variables x, the latent variables govern the data distribution. The joint distribution can be written as, $p(\mathbf{z}, \mathbf{x}) = p(\mathbf{z})\ p(\mathbf{x}|\mathbf{z})$, where $p(\mathbf{z})$ is the density over latent variables and is called the prior. In a generative model, we draw a sample from the prior and relate it to observations using the conditional distribution $p(\mathbf{x}|\mathbf{z})$, called the likelihood. For Bayesian models, performing inference is essentially estimating the posterior distribution, i.e. $p(\mathbf{z}|\mathbf{x})$. In the case of complex models, this is done via approximate inference. Examples of approximate inference methods include Markov Chain Monte Carlo [Hastings, 1970], which is a sampling based approach, and Variational Inference [Blei et al., 2018], which is an optimisation based approach. When this is cast this as an optimisation problem, it is possible to utilise Deep Learning methods to find near-optimal solutions.

3.2 Deep Learning

Neural Networks are a class of parameterised functions. A simple feed-forward neural network is a series of linear transformations interspersed with non-linear functions. The introduction of the non-linearity in the function computation allows neural networks, in principle, to represent any arbitrary function [Cybenkot, 1989]. These parameterised functions can then be trained using a multitude of optimisation algorithms to learn the function appropriate for the task at hand. Deep Learning is the all encompassing term for training neural networks of arbitrary structure (in terms of connectivity) and size (in terms of number of parameters) to perform well at a specified task. The flexibility offered by this approach allows for deep learning to be used as the tool of choice for a variety of Machine Learning tasks. We restrict ourselves to the sub-domain of Deep Generative Models.

In the case of simple feed-forward networks, model can be written as follows,

$$
\begin{aligned}
h^0 &= x \\
h^l &= \sigma(W^l h^{l-1} + b^l), \qquad\qquad l \in \{1, \ldots, L\} \\
\hat{y} &= h^L
\end{aligned}
$$

The linear transformations W^l and b^l are the trainable parameters and σ is the non-linear function, also called the *activation* function. L is the number of hidden layers in the network. The neural network is trained using Stochastic Gradient Descent (SGD) by computing the gradients with respect to a loss function. The choice of loss function used on the output of the neural network reflects the properties of the function that are desired.

3.3 Generative Adversarial Networks (GANs)

A Generative Adversarial Network is an approach to learning a generative model for complex high dimensional densities[Goodfellow et al., 2014]. In this framework, we initialise a neural network G, called the *generator*, that takes as input a latent vector $\mathbf{z}$ sampled from a specified prior i.e. $\mathbf{z} \sim p(\mathbf{z})$, and produces a sample from the data distribution. Simultaneously, we initialise a neural network D that takes as input a sample from the data distribution and classifies if it is from the true data generating distribution or not. The generator and discriminator are trained adversarially by alternatively optimising either the generator or the discriminator. The training objective that is optimised is,

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}(\mathbf{x})}[\log(D(\mathbf{x}))] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(\mathbf{z}))] \tag{3.1}$$

As the training progresses, the generator gradually becomes better at generating samples that are similar to the true data generating distribution.

Of specific interest to us are Generative Moment Matching Networks [Li et al., 2015]. Here the authors solve the same task as a GAN, which is to train a generative model of a complex data distribution. But they achieve this without having to train a separate discriminator network. They remove the need for a discriminator by instead using the Maximum Mean Discrepancy test statistic as a loss function. In doing so, the generator network is trained to produce samples, so as to minimise the MMD. Note that when the MMD is computed with a generated sample and the true dataset sample, it evaluates to zero.

Maximum Mean Discrepancy

Given two samples $X \sim p$ and $Y \sim q$, the Maximum Mean Discrepancy test statistic can be used to ascertain if $p = q$ [Gretton et al., 2008].

$$\mathrm{MMD}^2[\mathcal{F}, p, q] = \mathbb{E}_{x,x'}\left[k\left(x, x'\right)\right] - 2\mathbb{E}_{x,y}[k(x,y)] + \mathbb{E}_{y,y'}\left[k\left(y, y'\right)\right] \tag{3.2}$$

Here, k is the Gaussian kernel. The unbiased estimator for MMD is given by,

$$\begin{aligned}\mathrm{MMD}_u^2[\mathcal{F}, X, Y] =& \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} k(x_i, x_j) + \frac{-1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} k(y_i, y_j) \\ & - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, y_j)\end{aligned} \tag{3.3}$$

The MMD statistic is equal to zero, if and only if, $p = q$. All the terms involved in the computation of the MMD are differentiable and can therefore be used as a loss function. A generative network network that produces samples from the data distribution can be trained by using zero as the target value, as is the case with most other loss functions.

Chapter 4

MMDNet

In our problem setting we have a dataset of observed variables $\mathbf{X} \in \mathbb{R}^{n \times d}$, where n is the number of samples and d is the number of observed variables. We get the structure of the minMCM, $\mathcal{G}$ corresponding to the dataset, using the algorithm refer in section 2. This constitutes the structure learning component of identifying the FCM. Let l be the number of latent variables in $\mathcal{G}$. We now set up the generative model, from here on in referred to as MMDNet, to learn the functional relations given the minMCM model $\mathcal{G}$ and the data sample $\mathbf{X}$. Every observed variable is a non-linear stochastic function of a subset of the latent variables. This function is represented as a neural network that takes input of the form $x_i = f_i(\mathrm{pa}(x_i), \epsilon_i),\ f_i : \mathbb{R}^{k_i+1} \rightarrow \mathbb{R}$, where $k_i = |\mathrm{pa}(x_i)|$ is the number of the number of latent variables that cause x_i and the one additional variable is the exogenous noise term. It can be shown that using this setup, there exists a set of neural network that can arbitrarily closely approximate the true distribution [Goudet et al., 2017]. An example UDG and its corresponding minMCM model is shown in 4.1. By construction, the parents of the observed variables are exclusively latent variables that are naturally unobserved. We choose a distribution over the latent variables such that all of the latent variables are mutually independent, i.e. $p(\mathbf{z}) = \prod_{i=0}^{l} p(z_i)$. The default choice is a uniform distribution $z_i \sim U(-1, 1)$. The choice of latent variable distribution is further discussed in 5. Further, we sample the exogenous noise terms from a standard normal distribution. The FCM model now can be written as

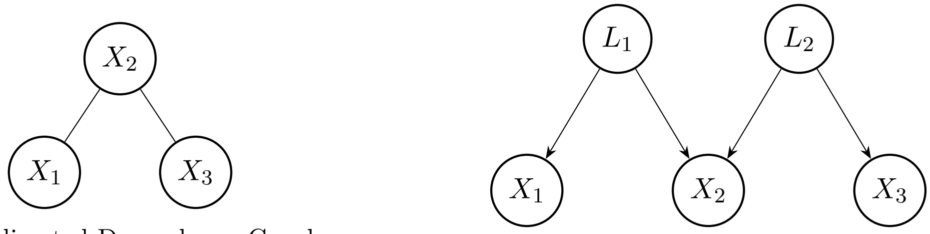
$$Z_i = \epsilon_i, \qquad \epsilon_i \sim U(-1, 1), \qquad i \in \{1, \cdots, l\} \tag{4.1}$$

$$X_i = f_i(pa(X_i), \epsilon_i), \qquad \epsilon_i \sim \mathcal{N}(0, 1), \qquad i \in \{1, \cdots, d\} \tag{4.2}$$

A sample is drawn from the assumed latent distribution and the corresponding subsets are fed into each of the neural networks to generate a sample from $\hat{\mathbf{X}} \sim \hat{p}(\mathbf{X})$. We want to reduce the difference between the generated sample $\hat{\mathbf{X}}$ and true sample $\mathbf{X}$. We do this by computing the Maximum Mean Discrepancy (MMD) as a loss function.

4.1 Dataset Generation

We generate artificial datasets by sampling from a chosen minMCM with a specified mechanism to observed variables. In the experiments conducted, two types of mechanism are used, a polynomial mechanism and a neural network mechanism.



(a) Undirected Dependency Graph

(b) Corresponding Minimal Latent Model

Figure 4.1: The Minimal Latent Model is recovered from the Undirected Dependency Graph

Polynomial Mechanism

The polynomial mechanism uses polynomial terms of specific degree, of the parents of an observed variables including the exogenous noise term in the MCM, e.g. that of 4.1. In this example, a polynomial mechanism of degree 2 for the variable x_2 would be computed as $x_2 = a + b\,\epsilon_2^2 + c\,l_1^2 + d\,l_2^2 + e\,\epsilon_2 l_1 + f\,l_1 l_2 + g\,\epsilon_2 l_2$, where $\{a, \cdots, g\}$ are random coefficients. In the general case, the equation is as follows,

$$x_i = \sum_{p_1+p_2+\cdots+p_{k+1}=d} \alpha_m \prod_{t=1}^{k+1} a_t^{p_t}, \qquad\qquad a_t \in \{\mathrm{pa}(x_i), \epsilon_i\}$$

Here, α_m are random coefficients sampled from $U(-1, 1)$, d is the degree of the polynomial, and a_t is the set of the parents of x_i and its corresponding exogenous noise term. Sampling from this model is done by keeping the randomly chosen coefficients constant and then feeding the values of the latent variables and the exogenous noise term, both randomly sampled from the uniform distribution $U(-1, 1)$.

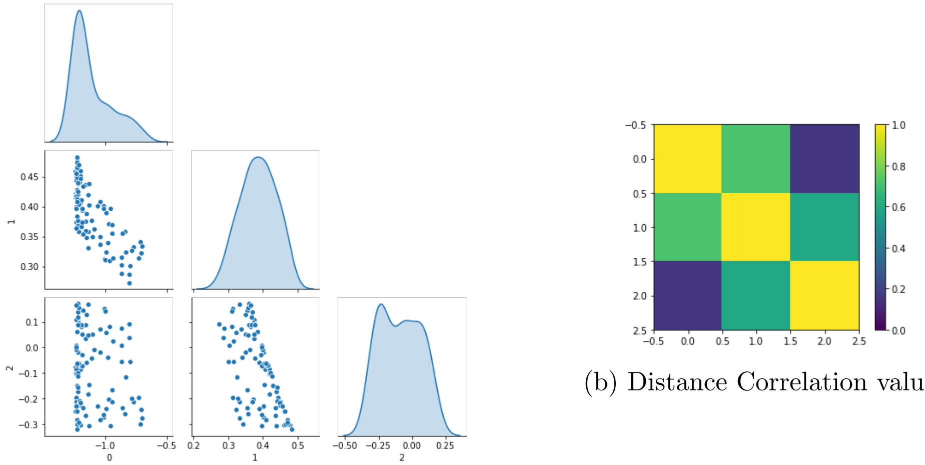
Neural Network Mechanism

Similar to the polynomial mechanism, A neural network identical to the one used in MMDNet is randomly initialised. Keeping the weights of the network constant, the neural network is fed with values of latent and exogenous noise terms sampled from the uniform distribution $U(-1, 1)$.

4.2 Experiments

For testing of the approach, data from 4 different minMCMs was generated. The configuration and hyperparameters for testing the models were kept the same for all models. The training was implemented using PyTorch [Paszke et al., 2019] and PyTorch Lightning [Falcon, 2019] libraries. The dataset size used for each of the models tested was $n = 1000$. The effects used for the observed variables were modelled with two hidden layer networks with 15 hidden units each. The non-linear function used was the tanh function. Another hyperparameter. The networks were trained using the Adam optimiser [Kingma and Ba, 2015] with a default learning rate of $lr = 1\mathrm{e}{-3}$. The models were each trained for $n_{epochs} = 2000$.

The models are evaluated by generating $n_{eval} = 100$ dataset samples, each of them the same size as the true dataset $n = 1000$ from the trained model and averaging the MMD

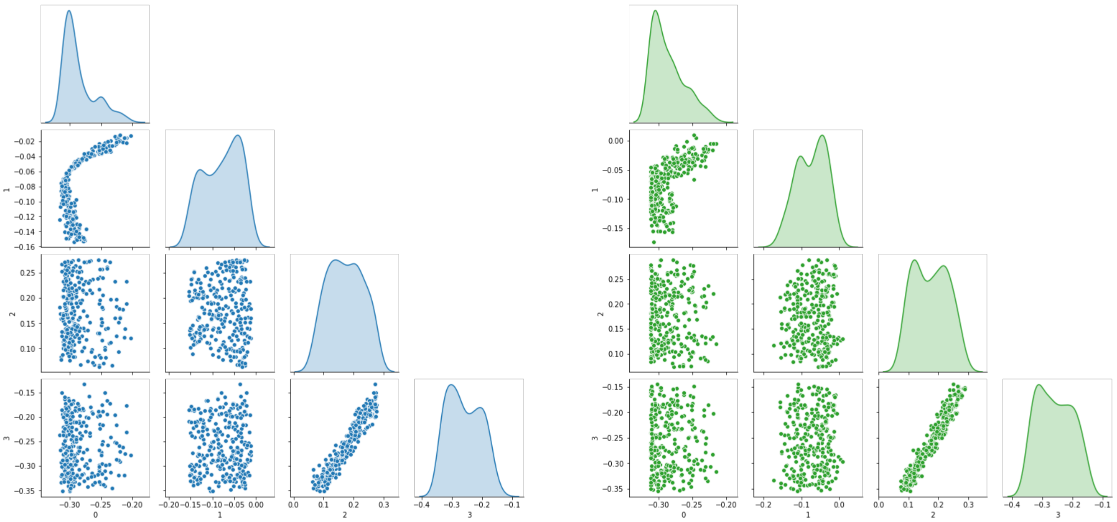


(a) Fitted generator

(b) Distance Correlation values

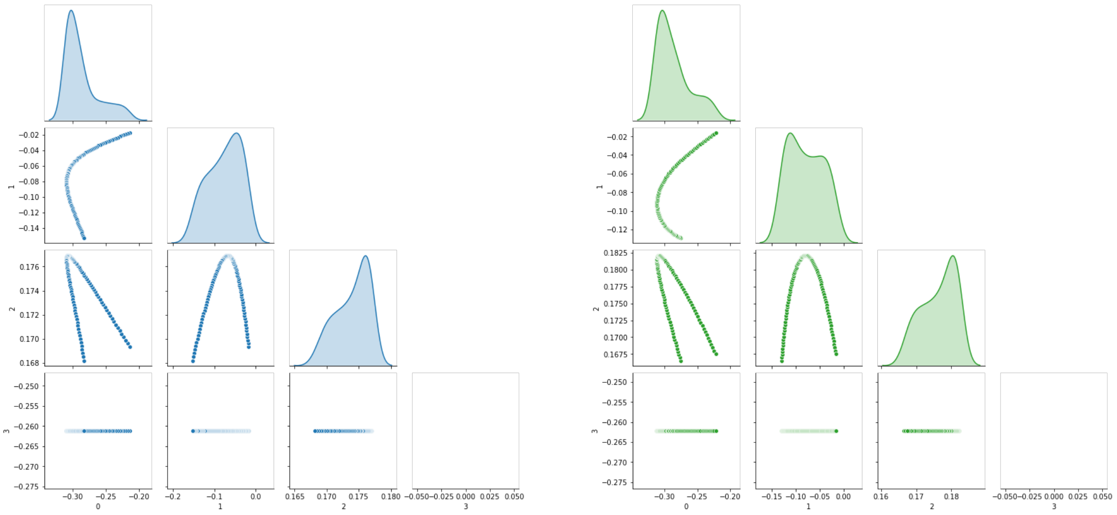
Figure 4.2: Pairplot for example UDG and corresponding Distance Correlation values

value for each of the samples. Figure 4.2 is a pairwise scatter plot of the observed variables. This plot is useful to visually ascertain the dependence between some the observed variables. In Figure 4.3, the pairwise scatter plot for the true dataset and the data sampled from the trained network are shown. The plots along the diagonal in these figures are kernel density estimates for the marginal of each observed variable. This is useful for a preliminary visual verification that the marginals are identical as well as the pairwise (in)depedence of each variable. This is better visualised in Figure 4.4. This plot is generated by interpolating along the first latent variable z_1 in $[-1, 1]$ and setting the remaining latent variables zero. The exogenous noise terms are also set to zero.



(a) True Sample (b) Generated Sample

Figure 4.3: Pairwise plot of all observed variables, in the true sample and the generated sample.



(a) True Sample (b) Generated Sample

Figure 4.4: Interpolating along the first latent variable keeping the second latent variable constant

Chapter 5

Conclusion

The MMDNet is capable of learning any FCM to arbitrary degree of precision, however this leaves out of the picture the learnability, in the sense of converging to a satisfactory solution and the scalability of the method. The computation of the MMD Loss is a potential bottleneck because it takes quadratic time in sample size to compute. There exists a linear time approximation to computation of this loss function that was not tested as the experiments performed were only in the low dimensional data space and were of a relatively small sample size. The choice of distribution for latent variables and the exogenous variables also affects learnability of the network. The MMDNet model does not necessitate the use of any specific latent distribution, only that it is possible to sample (efficiently) from this distribution. Once again, the choice of distribution over latent variables was kept constant across experiments, namely the uniform distribution. This was chosen for convenience and not a restriction of the model. It is interesting to note here what occurs when a latent model that allows all latent variables to cause all the observed variables was trained on a dataset that included independencies among observed variables. In other words, when a causal model that violates faithfulness is forcibly chosen. This strucutre implies that the none of the observed variables were (unconditionally) independent. In this case, the MMDNet learned functional relations that were (empirically) unfaithful, i.e. the model learned to generate data that had the same pairwise unconditional independencies despite sharing a latent cause.

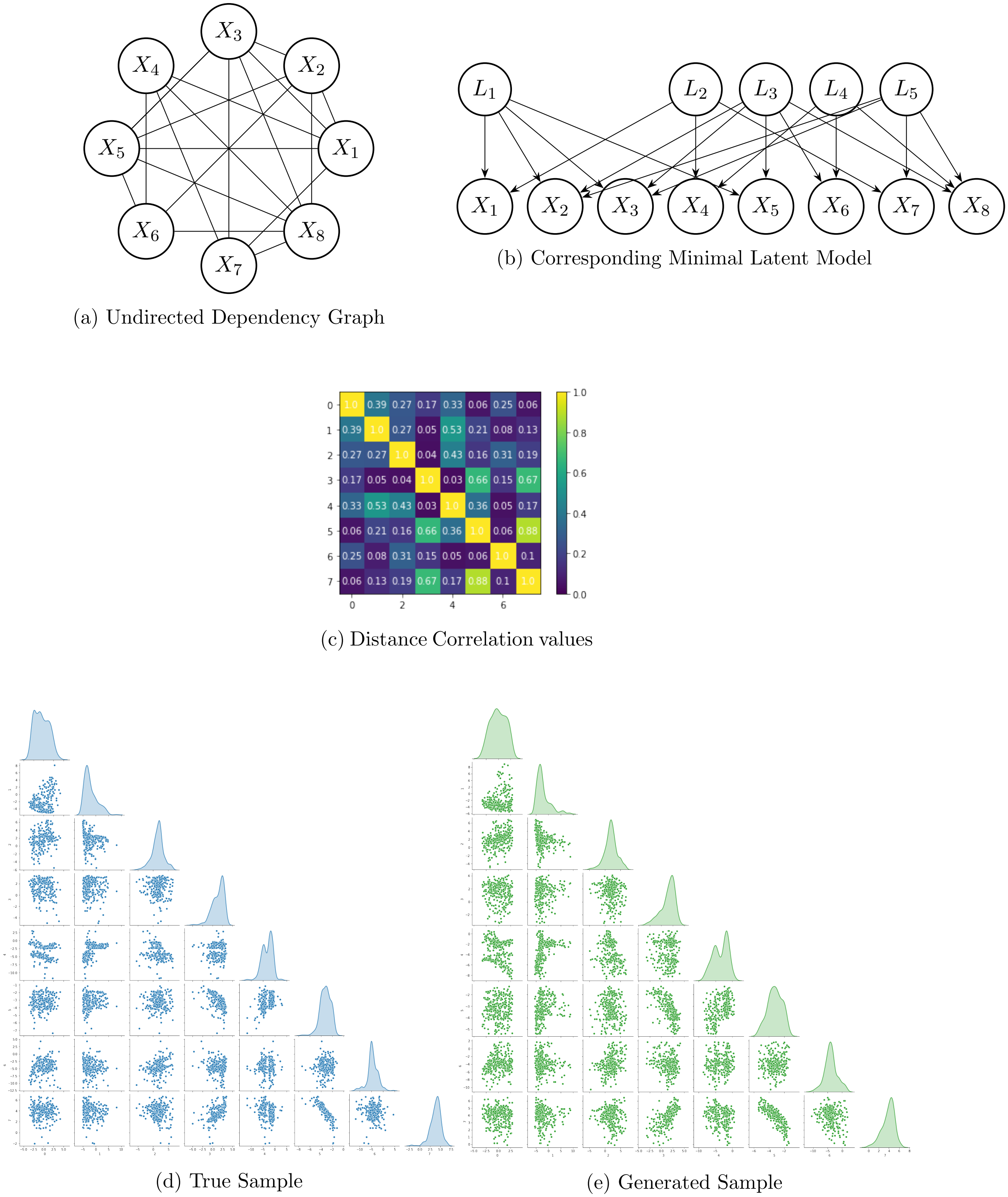
The learning of the MCM is agnostic to how the MCM was identified, with regard to non-linear independence tests and the notion of minimality, and vice versa. This makes the approach easy to apply in a range of situations.

While the experiments performed are exclusively on minMCM structures, learning the MMDNet is possible for any MeDIL model that is observationally consistent. Moreover, the minMCM retrieved from the approach also need not be unique. One could utilise domain expertise with regard to the nature of the data being analysed to justify selecting any one of the many options. Further, simultaneously analysing multiple MCMs could potentially lead to interesting conclusions.

Appendix A

Test Models and Plots

Graphs and plots for the datasets tested on are presented here. The figures contain the undirected dependency graph for the dataset and its corresponding minMCM. An example pairwise scatter plot of the variables from the true dataset and a sample genererated from the fitted MMDNet along with Distance Correlation values are plotted for each of the datasets used.



(a) Undirected Dependency Graph

(b) Corresponding Minimal Latent Model

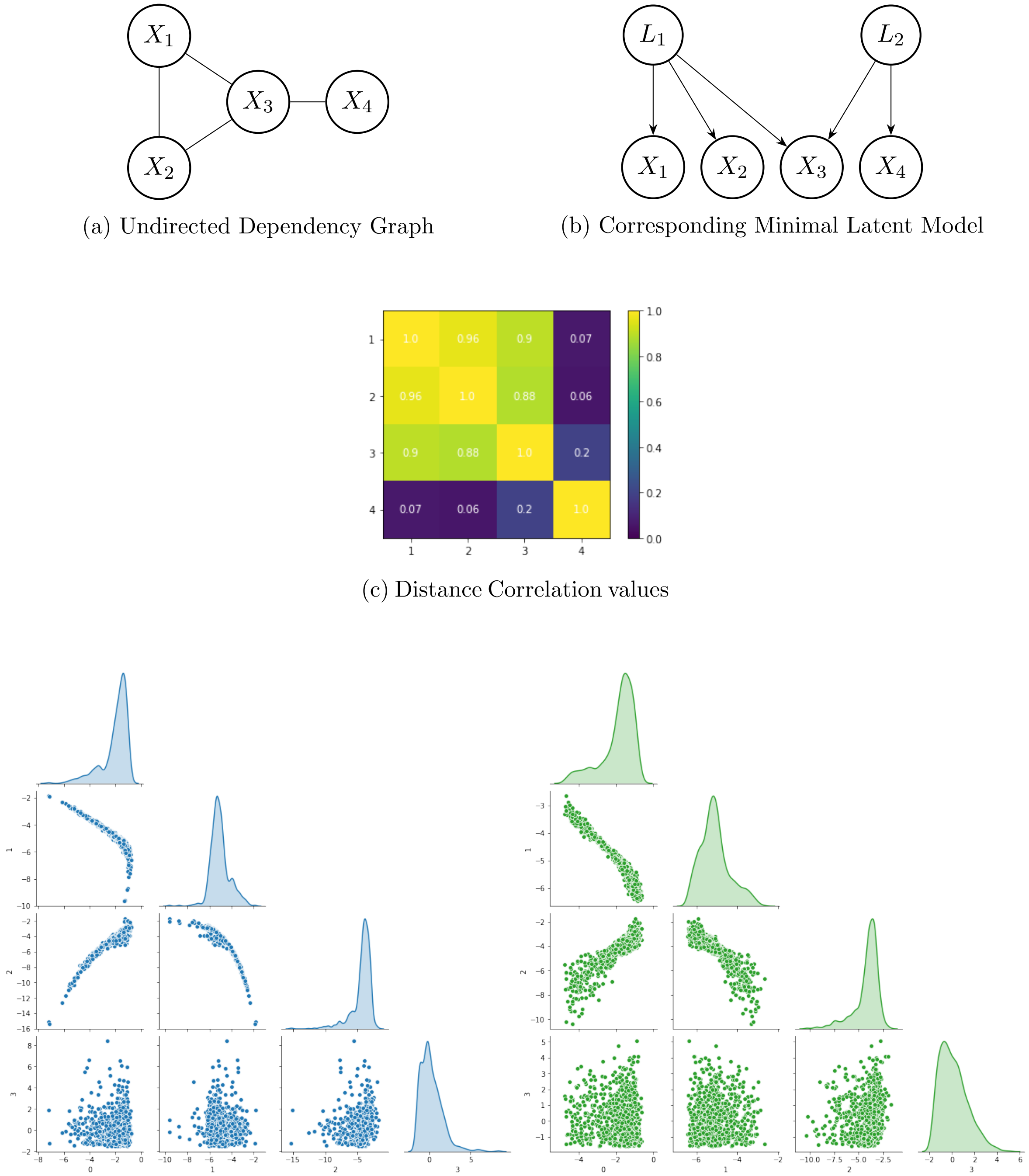
	0		2		4		6	
0	1.0	0.39	0.27	0.17	0.33	0.06	0.25	0.06
1	0.39	1.0	0.27	0.05	0.53	0.21	0.08	0.13
2	0.27	0.27	1.0	0.04	0.43	0.16	0.31	0.19
3	0.17	0.05	0.04	1.0	0.03	0.66	0.15	0.67
4	0.33	0.53	0.43	0.03	1.0	0.36	0.05	0.17
5	0.06	0.21	0.16	0.66	0.36	1.0	0.06	0.88
6	0.25	0.08	0.31	0.15	0.05	0.06	1.0	0.1
7	0.06	0.13	0.19	0.67	0.17	0.88	0.1	1.0

(c) Distance Correlation values

(d) True Sample

(e) Generated Sample

Figure A.1: minimal MCM 1



(a) Undirected Dependency Graph

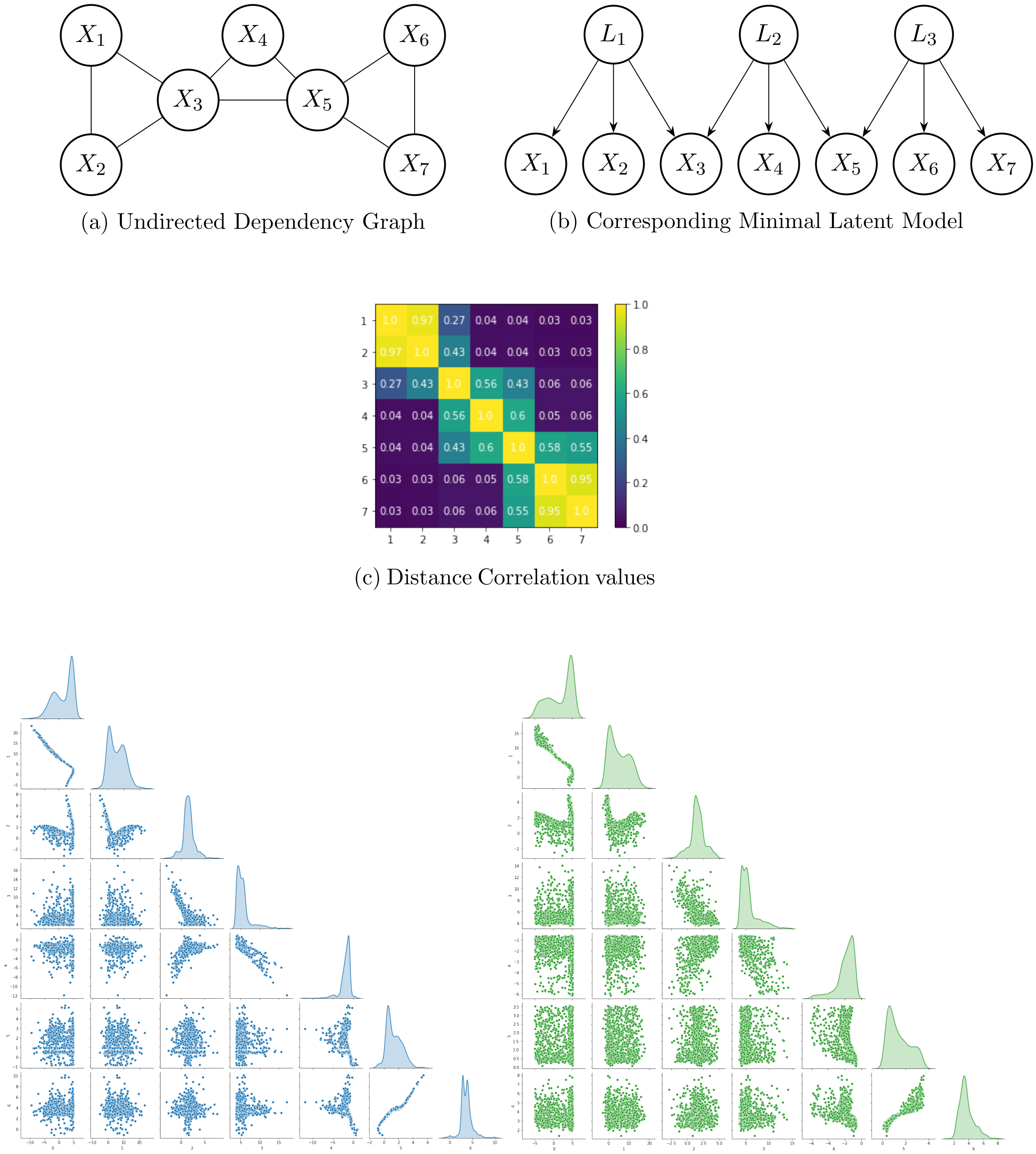
(b) Corresponding Minimal Latent Model

(c) Distance Correlation values

(d) True Sample

(e) Generated Sample

Figure A.2: minimal MCM 2



(a) Undirected Dependency Graph

(b) Corresponding Minimal Latent Model

(c) Distance Correlation values

(d) True Sample

(e) Generated Sample

Figure A.3: minimal MCM 3

Bibliography

[Blei et al., 2018] Blei, D. M., Kucukelbir, A., and Mcauliffe, J. D. (2018). Variational Inference: A Review for Statisticians. Technical report.

[Burgess et al., 2018] Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. (2018). Understanding disentangling in β-VAE.

[Cybenkot, 1989] Cybenkot, G. (1989). Approximation by Superpositions of a Sigmoidal Function*. Technical report.

[Falcon, 2019] Falcon, W. A. (2019). PyTorch Lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning Cited by*, 3.

[Goodfellow et al., 2014] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Nets. Technical report.

[Goudet et al., 2017] Goudet, O., Kalainathan, D., Caillou, P., Guyon, I., Lopez-Paz, D., and Sebag, M. (2017). Causal Generative Neural Networks.

[Granger, 1969] Granger, C. W. J. (1969). Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica*, 37(3):424–438.

[Gretton et al., 2008] Gretton, A., Borgwardt, K., Rasch, M. J., Scholkopf, B., and Smola, A. J. (2008). A Kernel Method for the Two-Sample Problem. *Journal of Machine Learning Research*, 1:1–10.

[Hastings, 1970] Hastings, W. K. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97.

[Hausman and Woodward, 1999] Hausman, D. M. and Woodward, J. (1999). Independence, Invariance and the Causal Markov Condition. Technical report.

[Holland, 1986] Holland, P. W. (1986). Statistics and Causal Inference. *Journal of the American Statistical Association*, 81(396):945.

[Hoyer et al., 2009] Hoyer, P. O., Janzing, D., Mooij, J. M., Peters, J., and Schölkopf, B. (2009). Nonlinear causal discovery with additive noise models. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 21*, pages 689–696. Curran Associates, Inc.

[Khemakhem et al., 2020] Khemakhem, I., Kingma, D. P., Monti, R. P., and Hyvärinen, A. (2020). Variational Autoencoders and Nonlinear ICA: A Unifying Framework. Technical report.

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR.

[Kingma and Welling, 2014] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR.

[Li et al., 2015] Li, Y., Swersky, K., and Zemel, R. (2015). Generative Moment Matching Networks. *32nd International Conference on Machine Learning, ICML 2015*, 3:1718–1727.

[Markham and Grosse-Wentrup, 2019] Markham, A. and Grosse-Wentrup, M. (2019). Measurement Dependence Inducing Latent Causal Models.

[Mathieu et al., 2016] Mathieu, M., Zhao, J., Sprechmann, P., Ramesh, A., and LeCun, Y. (2016). Disentangling factors of variation in deep representations using adversarial training. *Advances in Neural Information Processing Systems*, pages 5047–5055.

[Paszke et al., 2019] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H., Larochelle, H., Beygelzimer, A., d Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

[Pearl, 2000] Pearl, J. (2000). *Causality: Models, Reasoning, and Inference.* Cambridge University Press, USA.

[Pearl et al., 2016] Pearl, J., Glymour, M., and Jewell, N. P. (2016). *Causal Inference in Statistics: A Primer.* Wiley.

[Pearl and Verma, 1991] Pearl, J. and Verma, T. (1991). A Theory of Inferred Causation. Technical report.

[Székely et al., 2007] Székely, G. J., Rizzo, M. L., and Bakirov, N. K. (2007). Measuring and Testing Dependence by Correlation of Distance. 35(6):2769–2794.