Causal Inference and Deep Latent Variable Models

Abhishek Tiwari [1] ABHI.TIW1@GMAIL.COM
Isabela Maria Carneiro de Albuquerque[1] ISABELAMCALBUQUERQUE@GMAIL.COM
João Batista Monteiro Filho[1] JOAOMONTEIROF@GMAIL.COM
Yassine Yaakoubi[2] YASSINEYAAKOUBI@OUTLOOK.COM
[1]Institut National de la Recherche Scientifique - Université du Québec
[2]École Polytechnique de Montréal

Abstract

In this work, we are interested in to perform causal inference, *i.e.* to quantify the effect on a measurable outcome of changing one of the conditions under which the outcome is measured. The challenge here lies in understanding the effect of the outcome explicitly caused by a control variable, referred to as treatment. In order to do so, we frame the causal inference problem within the Variational Autoencoder (VAE) framework in such a way that makes us able to infer hidden causes and predict the outcome given an observed treatment. We build upon previous works and verify the performance gain obtained by utilizing a richer prior, namely variational mixture of posteriors (VampPrior), on the latent layer of the VAE. Implementation was performed using a stochastic variational inference library and experiments were executed to verify the efficacy of this approach.

1. Introduction

Causal analysis is defined as a tool used to infer beliefs or probabilities under dynamic conditions (Pearl, 1995). Beliefs variations are induced by changes on treatments or external interventions. Real life experiments often involve variables called confounders which have an effect on both intervention and outcome. These confounders may not be observable and instead proxy variables (J. D. Angrist, 2008), which are their noisy observations, are available. These confounders and proxies share causal relationships with one another, as well as with the selected treatment and observed effect. Usually, those relationships are represented in terms of graphical models.

One of the goals of causal inference is to model the expected effect on the outcome of different treatments applied to an individual described by a set of proxy variables. However, the observed proxies are not directly related to treatment and outcome. Hence, knowledge about the true confounders is necessary. Recent works have focused on developing methods and conditions for getting true causal effects given the proxy variables (Selén., 1986) and also creating better methods to measure and apply them to causal inference (Wooldridge, 2009). However, current models rely on strong assumptions about the nature of confounders (Louizos et al., 2017), such as assuming a binary treatment. In this context, (Louizos et al., 2017) proposed to use a latent variable model, namely Variational Autoencoder (VAE), to model causality with hidden confounders and infer individual causal effect of the treatment on the outcome.

In this report we describe the methodology and steps taken to reproduce the work of (Louizos et al., 2017), which will be referred from now on as reference paper. We reimplemented the complete model from scratch using different neural networks and stochastic programming frameworks. In order to validate the implementation, we repeated experiments present in the original paper for a hybrid (part real and part synthetic) dataset and compared with results reported in the paper as well as the ones generated by executing the code released by the authors. Furthermore, we also tried to improve the original model by implementing a recently proposed more flexible prior for VAEs (Tomczak & Welling, 2017).

This report is organized as follows: in Section 2 a brief review about causality and VAEs is provided; in Sections 3 and 4 we present details of the reproduced work and extension; in Section 5 we depict the experiments performed and results; conclusions and future works are provided in Section 6.

IFT6269 - Probabilistic Graphical Models, Fall 2017. Project final report. Professor: Simon Lacoste-Julien.

2. Background

2.1. Causality

Rubin's Causal Model (Sekhon, 2008) is a framework developed for the statistical analysis of cause of effect based on the idea of potential outcomes. Consider:

- t_i, a binary treatment t for individual i with 1 referring to assigning the treatment and 0 to no treatment;
- y_i is the outcome on individual i given a treatment value.

Each individual can have two potential outcomes or (counterfactuals) available:

- $y_i(0)$, corresponding to not receiving the treatment;
- $y_i(1)$, corresponding to receiving the treatment.

In the hypothetical case where both potential outcomes are available, the effect for an individual, namely the Individual Treatment Effect (ITE), can be defined as the difference between the two potential outcomes. However, in practice only one of the outcomes is realizable for the individual i and it is given by:

$$y_i = y_i(0)(1 - t_i) + y_i(1)t_i. \tag{1}$$

This problem of the non-availability of one of the counterfactuals is called the fundamental problem of causal inference. Therefore, in order to get the treatment effect, we predict the Average Treatment Effect (ATE) as the expected value of the potential outcomes over the subjects. For a binary outcome, it is defined as:

$$ATE = \mathbb{E}[y_i(1)] - \mathbb{E}[y_i(0)]. \tag{2}$$

The above mentioned metric cannot be properly estimated if there are confounding variables in the system, which will introduce bias (Greenland et al., 1999). There are two ways to deal with such confounding variables. One is by choosing the proper study design (randomized trail), by nullifying the effect of confounding variables. Another possibility is to measure the confounders and "adjust" for them during treatment effect calculation.

In order to develop causal analysis in the framework of graphical models, (Pearl, 1995) has added some relevant notation to understand the causality concepts:

1. The directed arrows in a directed graph represent the causal relationships between the variables involved, as opposed to the common use to represent conditional independence between variables;
2. The graphs hence created represent the investigators understanding of major causal influence among measurable quantities in the domain;
3. The causal effect by a treatment variable t on an outcome y is represented by $\mathbb{E}[y|do(t = 1)]$, where *do* represents the fact that the treatment has been kept at a specific value by external interventions on the system which do not affect other variables and their causal relationships in the system.

As per the above definition, *do* symbol removes the t from the given mechanism and sets it to specific value by some external intervention. Using this understanding, $P(y|do(t = 1))$ will represent the probability of y induced by deleting all equations corresponding to t and setting t equals 1 in the remaining factors.

Pearl defines the causal effect for a given treatment t and an outcome y and other confounding variables Z as:

$$P(y|do(t)) = \sum_Z P(Z, y|do(t)), \tag{3}$$

where the new distribution $P(Z, y|do(t))$ is called the post-intervention distribution. The summation done over Z is referred to as "adjusting" for or controlling the confounding variables. The different treatment effects can then be calculated in a similar way as the difference of the expected value of this causal effect. One of the central themes of causal analysis is to verify whether the post intervention distribution is deducible by the pre-intervention joint distribution $P(t, y, Z)$. This is known as the identifiability of the causal effect. The problem of identifiability can be simplified using graphical analysis as all of assumed correlation and underlying assumptions are directly represented by the graphical model.

The use of Eq. 3 assumes that the causal effect is observable from experimental data in the case when we know all the involved confounding variables. However, in practical experiments this might not be the case and some of the confounding variables are unobserved due to study constraints and other reasons. In such a case, experimenters are left with two choices:

1. Try to find the minimum set of variables to be measured and adjusted for it, hence providing the true value of causal effect. Such set of factors is known as the admissible set or the sufficient set. This information can be deduced from the graphical model and once captured can save time and complexity for collecting the data.
2. In cases where observations of the admissible set are also not possible, experimenters can collect proxy variables.

The first choice often makes use of the back door criterion (Pearl, 2009) to get the admissible set of factors. In this case, a set S with treatment t and outcome y is said to be admissible if the following conditions hold:

1. No element in S is a descendant of the treatment variable;
2. The elements of S d-separates all the "back-door" paths from the treatment variable to the outcome variable, namely all paths which end with an arrow pointing towards the treatment variable.

The criterion can be understood based on the idea that while the direct relation between t and y carries useful causal information, the "back-door" paths carry undesired causal effects. Hence, measuring and controlling these admissible sets is equivalent to d-separating t and y interaction except for the direct one.

The graph presented in Fig. 1 represents the causal relationship to be studied in this work. Its admissible set is given by $\{Z\}$ and we can directly measure the correct causal effect if we know and adjust for the confounder.

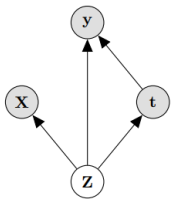


Figure 1. Graph representing the assumed causal model (Louizos et al., 2017).

If the back-door criteria is met, it can be shown that the post intervention factorization in Eq. 3 can only be described using the variables of the admissible set Z, given as follows:

$$P(y|do(t)) = \sum_z P(y|t, Z)P(Z). \tag{4}$$

In case confounders are not measurable, the experimenters are left with the second choice. *i.e.* using proxy variables to approximate it. Using proxies in place of unmeasured confounders has been shown to reduce bias (J. D. Angrist, 2008). It is also known that the proxies cannot be directly used as confounders as this may lead to even more bias (Fuller, 1987), (Rothman et al., 2008), (Griliches & Hausman, 1986). Due to the above restrictions, the problem of approximating the hidden confounders through available data has received a lot of attention (Cai & Kuroki, 2012), (Rothman et al., 2008), (Miao et al., 2016) (Kuroki & Pearl, 2014).

Our reference paper has used the rules for intervention manipulation defined by (Pearl, 1995) for the simple graph with confounders and proxies shown in Fig. 1 to express the causal effect in terms of probability distributions and then find the joint probability with the confounder in a latent variable and generative modeling framework. Given a proxy X, outcome y, binary treatment t and confounder Z, we use the back-door criteria to get:

$$P(y|X, do(t = 1)) = \int_Z P(y|X, do(t = 1), Z)P(Z|X, do(t = 1))dZ. \tag{5}$$

Further, using the intervention manipulation rules, we obtain:

$$P(y|X, do(t = 1) = \int_Z P(y|X, t = 1, Z)P(Z|X)dZ. \tag{6}$$

2.2. Variational Autoencoders

As pointed out in the work of (Lamb et al., 2016), variational autoencoders (VAE) (Kingma & Welling, 2014) (Rezende et al., 2014) have emerged as a popular framework within the context of generative models that support tractable approximate inference leveraging neural networks both for generative modeling and for approximate inference in a latent variables model.

Assume $p(X, Z)$, where X is the observed data and Z is the latent representation. $p(X, Z)$ can be decomposed into the likelihood and the prior as: $p(X, Z) = p(X|Z)p(Z)$. Using Baye's Rule to calculate the posterior gives:

$$p(Z|X) = \frac{p(X|Z)p(Z)}{\int_z p(X|z)p(z)}. \tag{7}$$

The integral in the denominator makes the analytical solution for posterior intractable. Thus, VAE approximates it with the family of distributions $q_\lambda(Z|X)$, where λ is the variational of parameters for the given family. We minimize the KL divergence to ensure that the approximate distribution used is close to the true posterior:

$$\begin{aligned} KL(q_\lambda(Z|X)||p(X|Z)) = &\mathbb{E}_q[log(q_\lambda(Z|X))] \\ &- \mathbb{E}_q[\log p(X, Z)] \\ &+ \log p(X). \end{aligned} \tag{8}$$

The optimal posterior will therefore be :

$$q^*_\lambda(Z|X) = \arg\min_\lambda KL(q_\lambda(Z|X)||p(X|Z)). \tag{9}$$

However, due to the occurrence of $p(X)$, the KL is still intractable. We can manipulate the above equation by defining the Evidence Lower Bound (ELBO):

$$ELBO(\lambda) = \log(p(X)) - KL(q_\lambda(Z|X)||p(X|Z)). \tag{10}$$

The ELBO can now be be expressed as:

$$ELBO(\lambda) = \mathbb{E}_q[\log p(X, Z)] - \mathbb{E}_q[\log q_\lambda(Z|X)], \quad (11)$$

which can be simplified using chain rule to:

$$ELBO(\lambda) = \mathbb{E}_q[\log p(X|Z)] - KL(\log q_\lambda(Z|X)||p(Z)). \quad (12)$$

The above equation consists of two main distributions. $q_\lambda(Z|X)$ is the latent variable model while $p(X|Z)$ is the generative model.

The VAE learns the parameters of these distributions using neural networks in an encoder/decoder setup. The encoder (inference model) takes input data (X) and outputs the parameters of the latent variable model $q_\theta(Z|X)$. The decoder (generative model) takes sampled latent variables (Z) as input and returns data samples from $p_\phi(X|Z)$. Parameters θ and ϕ, are typically the weights and biases of the neural networks which are selected to maximize the ELBO using stochastic gradient descent. The negative of the ELBO is the loss function used for the neural networks:

$$\begin{aligned} l(\theta, \phi) = &- \mathbb{E}_{q_\theta(z|x)}[\log p_\phi(X|Z)] \\ &+ KL(\log q_\theta(Z|X, \lambda)||p(Z)). \end{aligned} \quad (13)$$

The first term in above equation is the reconstruction loss. This term encourages the decoder to learn to reconstruct the data. The second term can be seen as a regularization term which tries to ensure that the approximation follows the prior distribution as much as possible.

3. Causal Effect Variational Autoencoder

In this project, we aim to study the Causal Effect Variational Autoencoder (CEVAE). In the work of (Louizos et al., 2017), the tackled problem was causal inference of effects of an unobserved confounder $\mathbf{Z}$ on an observed outcome $\mathbf{y}$, based on its noisy observations $\mathbf{X}$ and the treatment $\mathbf{t}$, *i.e.* learning the latent variable causal model. The graphical model presented in Fig. 1 represents the assumed causal model relating the aforementioned variables.

In order to solve this problem, two main assumptions were made: (i) $\mathbf{t}$ is binary; (ii) observations $(\mathbf{X}, \mathbf{t}, \mathbf{y})$ are enough to approximately recover the joint distribution $p(\mathbf{Z}, \mathbf{X}, \mathbf{t}, \mathbf{y})$.

The approach taken in the paper to solve the problem is to employ VAEs to infer complex non-linear relationships between the variables. Neural networks were used to parameterize the conditional distributions defined by the edges of the model represented by the causal graph.

The authors use VAE with the assumption that the the observations factorize conditioned on the latent variables, and use an inference network (Kingma & Welling, 2014) (Rezende et al., 2014) which follows a factorization of the true posterior. For the generative network (decoder), the architecture used has been inspired by TARNet (Shalit et al., 2016). Instead of conditioning on observations, the authors condition on the latent variables Z.

For describing the notation, we assume X_i corresponds to an input data point proxy (*e.g.* the feature vector of a given subject), t_i corresponds to the treatment assignment, y_i to the outcome of the of the particular treatment and Z_i corresponds to the latent hidden confounder. Each of the corresponding factors is described as:

$$p(\mathbf{z}_i) = \prod_{j=1}^{Dz} \mathcal{N}(z_{ij}|0, 1), \quad (14)$$

$$p(\mathbf{x}_i|\mathbf{z}_i) = \prod_{j=1}^{Dx} p(x_{ij}|\mathbf{z}_i), \quad (15)$$

$$p(t_i|\mathbf{z}_i) = \text{Bern}(\sigma(f_1(\mathbf{z}_i))), \quad (16)$$

with $p(x_{ij}|\mathbf{z}_i)$ being an appropriate probability distribution for the covariate j and $\sigma(.)$ being the logistic function. D_x is the dimension of X and D_z is the dimension of $\mathbf{z}$. For a discrete outcome we parametrize the probability distribution as a Bernoulli distribution parametrized by a TARnet:

$$p(y_i|t_i, \mathbf{z}_i) = \text{Bern}(\pi = \hat{\pi}_i), \quad (17)$$

with $\hat{\pi}_i = \sigma(t_i f_2(\mathbf{z}_i) + (1 - t_i) f_3(\mathbf{z}_i))$. Note that each of the $f_k(.)$ is a neural network parametrized by θ_k for $k = 1, 2, 3$.

The authors employ the following posterior approximation:

$$q(\mathbf{z}_i|\mathbf{x}_i, t_i, y_i) = \mathcal{N}(\mu_j = \bar{\mu}_{ij}, \sigma_j^2 = \bar{\sigma}_{ij}^2), \quad (18)$$

where:

$$\begin{aligned} \bar{\boldsymbol{\mu}}_i &= t_i \boldsymbol{\mu}_{t=0,i} + (1 - t_i)\boldsymbol{\mu}_{t=1,i}, \\ \bar{\boldsymbol{\sigma}}_i^2 &= t_i \boldsymbol{\sigma}_{t=0,i}^2 + (1 - t_i)\boldsymbol{\sigma}_{t=1,i}^2, \\ \boldsymbol{\mu}_{t=0,i}, \boldsymbol{\sigma}_{t=0,i}^2 &= g_2 \circ g_1(\mathbf{x}_i, y_i), \\ \boldsymbol{\mu}_{t=1,i}, \boldsymbol{\sigma}_{t=1,i}^2 &= g_3 \circ g_1(\mathbf{x}_i, y_i). \end{aligned}$$

This architecture is similar to TARnet's inference network, *i.e.* split them for each treatment group in t after a shared representation $g_1(\mathbf{x}_i, y_i)$ and each $g_k(.)$ is a neural network with variational parameters parameters ϕ_k. The objective function has been given by:

$$\begin{aligned} \mathcal{L} = \Sigma_{i=1}^N \mathbb{E}_{q(\mathbf{z}_i|\mathbf{x}_i, t_i, y_i)}[&\log p(\mathbf{x}_i, t_i|\mathbf{z}_i) + \log p(y_i|t_i, \mathbf{z}_i) \\ &+ \log p(\mathbf{x}_i) - \log q(\mathbf{z}_i|\mathbf{x}_i, t_i, y_i)]. \end{aligned} \quad (19)$$

For predicting new subject predictions, the treatment assignment t along with outcome y are required. The authors have introduced two auxiliary distributions which help predict y and t for new samples. The distributions are:

$$q(t_i|\mathbf{x}_i) = \text{Bern}(\pi = \sigma(g_4(\mathbf{x}_i))). \tag{20}$$

$$q(y_i|\mathbf{x}_i, t_i) = \text{Bern}(\pi = \bar{\pi}_i), \tag{21}$$

where $\bar{\pi}_i = t_i(g_6 \circ g_5(\mathbf{x}_i)) + (1 - t_i)(g_7 \circ g_5(\mathbf{x}_i))$. To estimate the parameters for these distributions, the lower bound has been modified to:

$$\mathcal{F}_{\text{CEVAE}} = \mathcal{L} + \sum_{i=1}^{N} (\log q(t_i = t_i^*|\mathbf{x}_i^*) + \log q(y_i = y_i^*|\mathbf{x}_i^*, t_i^*)), \tag{22}$$

where $\mathbf{x}_i$, t_i^* and y_i^* are the input (proxy) treatment and outcome values in the training set. The aforementioned autoencoder has been called Causal Effect Variational Autoencoder (CEVAE).

4. Extension - Variational Mixture of Posteriors prior

In order to extend the work done by (Louizos et al., 2017) and as an attempt to improve CEVAE's performance, we implemented and included the Variational mixture of posteriors prior (VampPrior) proposed by (Tomczak & Welling, 2017). In this section, we describe what motivated this choice, details of this prior and aspects of its implementation.

As described previously, when training a VAE we aim to maximize the ELBO. Observing Eq. 12, we notice that by doing so, the term $KL(\log(q_\lambda(\mathbf{z}|\mathbf{x}))||p(\mathbf{z}))$ is minimized. In the standard VAE set-up, the prior is fixed to $\mathcal{N}(0, 1)$ and this minimization encourages the posterior to match this distribution. Thus, this KL-divergence can be understood as a regularization factor that tries to "push" the posterior to be closer to the (usually) more simple prior. However, if this KL-divergence is too close to 0, prior and posterior distributions will be too close. When this overregularization happens, $p(\mathbf{z}|x)$ does not encode relevant information from the training data as expected. As a result, the model learns a poor latent representation of the data and many latent dimension are inactive, which means that they do not change as the input changes (Hoffman & Johnson, 2016).

As a solution for the aforementioned issue, (Tomczak & Welling, 2017) proposed to choose a prior that maximizes the ELBO given the posterior, instead of choosing the prior in advance, which is the usual approach. For a given training set with $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ samples, the $p(\mathbf{z})$ that maximizes the ELBO is given by:

$$p(\mathbf{z}) = \frac{1}{N} \sum_{n=1}^{N} q(\mathbf{z}|\mathbf{x}_n), \tag{23}$$

which is the average of the aggregated posteriors calculated for all training samples. However, choosing this distribution as a prior for the VAE would bring two major drawbacks: (i) as it is necessary to compute $q(\mathbf{z}|\mathbf{x}_n)$ for all N training samples, this prior is highly computationally expensive; (ii) the use of the training data to calculate the prior makes the model more prone to overfitting. To mitigate those drawbacks, (Tomczak & Welling, 2017) proposed to use as prior a variational mixture of posteriors, which consists in calculating an average of posteriors over a set of K learnable pseudo-inputs $\mathbf{u}_k$. Hence, the VampPrior is defined as:

$$p(\mathbf{z}) = \frac{1}{K} \sum_{k=1}^{K} q(\mathbf{z}|\mathbf{u}_k). \tag{24}$$

The pseudo-inputs are a set of parameters jointly learned with the model. More specifically, a single-layer neural network is trained to map "dummy" inputs, such as a $K \times K$ identity matrix, to a set of K outputs with similar structure as the training data.

For the CEVAE, we replaced the prior for sample i defined in Eq.14 by:

$$p(\mathbf{z}_i) = \frac{1}{K} \sum_{k=1}^{K} q(\mathbf{z}_i|\mathbf{u}_k, t_i, y_i). \tag{25}$$

5. Experiments and Discussion

In order to verify the effectiveness of the proposed framework, experiments were performed with the code released by the authors [1] (Tensorflow+Edward). Moreover, we implemented the same architecture from scratch[2],[3] using different tools, namely Pytorch [4] and Pyro [5].

Experiments were performed with data obtained from the Infant Health and Development Program (IHDP) (Hill, 2011). In this dataset, the proxy variables are collected measurements of children and their mothers used during a randomized experiment that studied the effect of home visits by specialists on future cognitive test scores. Treated and control outcomes are simulated, thus allowing us to know the "true" individual causal effects of the treatment.

Due to differences in the behavior of Pyro with respect to Edward, we had to change the activation function of the output layers of all the neural networks predicting the parameters of Bernoulli distributions. In the architecture pro-

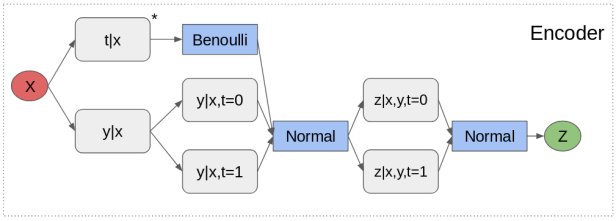
[1] https://github.com/AMLab-Amsterdam/CEVAE

[2] https://github.com/IsabelaAlbuquerque/CEVAE-VampPrior

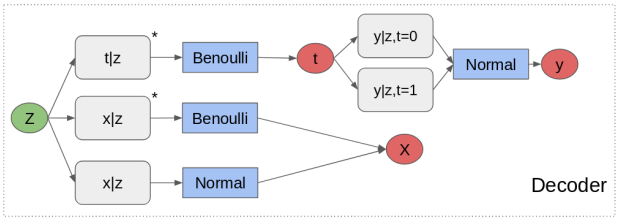
[3] https://github.com/joaomonteirof/ift6269/tree/master/project

[4] http://pytorch.org/

[5] http://pyro.ai/



(a)



(b)

Figure 2. Model architecture for both encoder and decoder. Each gray box corresponds to a fully connected neural network. The * symbol indicates where it was necessary to change the activation function in the output layer from identity to sigmoid.

posed in the reference paper, authors utilized identity activation in the output layers of those neural networks and elu activation in the intermediate layers. This can lead to negative outputs. In Edward, negative arguments are automatically truncated to 0 when one tries to sample from a Bernoulli random variable. Pyro, instead, throws an exception when such a case occurs. Thus, we substituted the output activation of those layers to the sigmoid function. We performed the same modification in the author's code to perform a fair comparison. Architecture details can be seen in figure 2.

5.1. Sampling from VampPrior

The implementation of VAEs using Pyro requires a method to sample from both prior and posterior distributions. Hence, in order to use a VampPrior in our model, we had to implement a sampling scheme for it.

From Eq. 24 and the fact that the $q(\mathbf{z}|\mathbf{u}_k)$ is Gaussian, one can notice that the VampPrior is a rescaled mixture of Gaussians with parameters sampled from the encoder model. Therefore, the VampPrior $p(\mathbf{z}_i)$ for sample i will be parametrized by

$$\mu_i = \frac{1}{K}\sum_{k=1}^{K}\mu_k, \tag{26}$$

$$\sigma_i^2 = \frac{1}{K}\sum_{k=1}^{K}\sigma_k^2, \tag{27}$$

where μ_k and σ_k^2 are the parameters of the posterior for each pseudo-input.

5.2. Results

Results are shown in Table 1. Values of Average Treatment Effect (ATE) and Precision in Estimation of Heterogeneous Effect (PEHE) are presented. ATE and PEHE are defined as follows:

$$\text{ATE} = \left|\frac{1}{N}\sum_{i=1}^{N}\left[(\hat{y}_{i1}-\hat{y}_{i0})-(y_{i1}-y_{i0})\right]\right|, \tag{28}$$

$$\text{PEHE} = \frac{1}{N}\sum_{i=1}^{N}\left[(\hat{y}_{i1}-\hat{y}_{i0})-(y_{i1}-y_{i0})\right]^2, \tag{29}$$

where $\hat{y}_{i1}$ ($\hat{y}_{i0}$) and y_{i1} ($\hat{y}_{i0}$) correspond to the predicted and actual outcomes when $t_i = 1$ ($t_i = 0$) , respectively.

The table contains the results claimed in the reference paper for 1000 replications (CEVAE-paper). For practical reasons, all the experiments we executed were replicated 100 of times. Our executions using the author's implementations are CEVAE-rep for the original version and CEVAE-mod for the version including the modifications described previously. Our implementation was able to match very closely the results obtained with the original code. One can also notice that the inclusion of VampPrior improved test ATE.

Table 1. PEHE and ATE obtained on training and test data. Results for comparison models reported in the original paper are above the horizontal line and results obtained in the experiments executed during the project are below.

Method	Train PEHE	Train ATE	Test PEHE	Test ATE
OLS-1	$5.8 \pm .3$	$.73 \pm .04$	$5.8 \pm .3$	$.94 \pm .06$
OLS-2	$2.4 \pm .1$	$.14 \pm .01$	$2.5 \pm .1$	$.31 \pm .02$
BLR	$5.8 \pm .3$	$.72 \pm .04$	$5.8 \pm .3$	$.93 \pm .05$
k-NN	$2.1 \pm .1$	$.14 \pm .01$	$4.1 \pm .2$	$.79 \pm .05$
TMLE	$5.0 \pm .2$	$.30 \pm .01$	–	–
BART	$2.1 \pm .1$	$.23 \pm .01$	$2.3 \pm .1$	$.34 \pm .02$
RF	$4.2 \pm .2$	$.73 \pm .05$	$6.6 \pm .3$	$.96 \pm .06$
CF	$3.8 \pm .2$	$.18 \pm .01$	$3.8 \pm .2$	$.40 \pm .03$
BNN	$2.2 \pm .1$	$.37 \pm .03$	$2.1 \pm .1$	$.42 \pm .03$
CFRW	$.71 \pm .0$	$.25 \pm .01$	$.76 \pm .0$	$.27 \pm .01$
CEVAE-paper	$2.7 \pm .1$	$.34 \pm .01$	$2.6 \pm .1$	$.46 \pm .02$
CEVAE-rep	$2.5 \pm .4$	$.28 \pm .04$	$2.2 \pm .3$	$.47 \pm .08$
CEVAE-modif	$3.0 \pm .5$	$.33 \pm .05$	$2.4 \pm .3$	$.49 \pm .07$
CEVAE-ours	$3.0 \pm .5$	$.33 \pm .04$	$2.5 \pm .3$	$.53 \pm .07$
CEVAE-ours + VP	$3.1 \pm .5$	$.39 \pm .07$	$2.6 \pm .3$	$.44 \pm .06$

The learning curves (negative ELBO) with the standard prior and with VampPrior are shown in Figures 3, 4, 5, and 6. Those curves obtained by calculating average and stan-

dard deviation of the negative ELBO over the 100 replications.

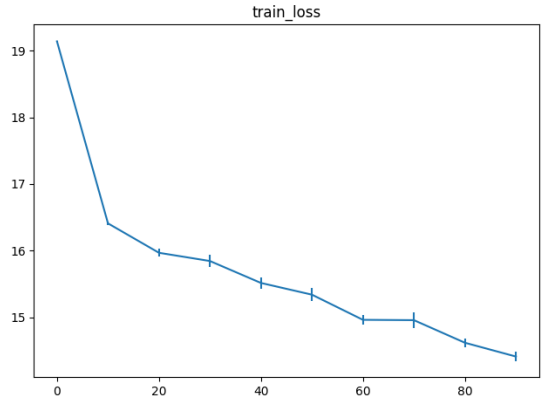


Figure 3. Learning curve for CEVAE with standard prior on the training set.

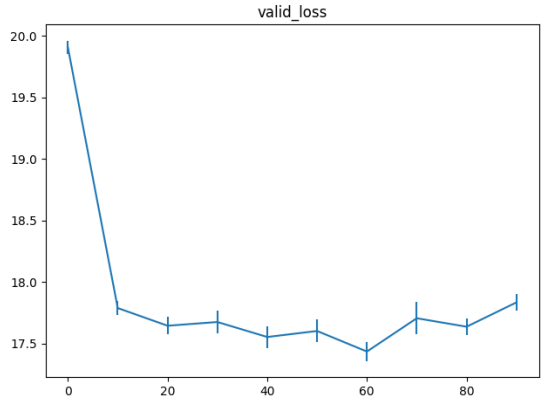


Figure 4. Learning curve for CEVAE with standard prior on the validation set.

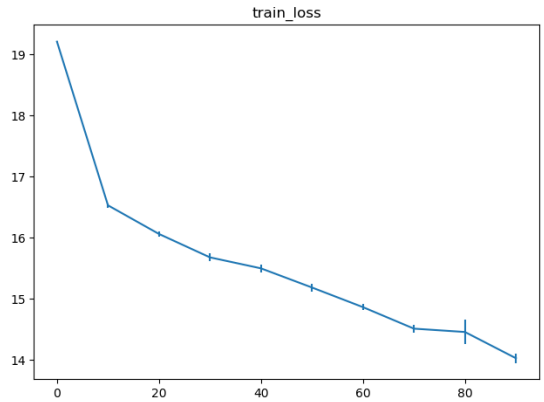


Figure 5. Learning curve for CEVAE with VampPrior prior on the training set.

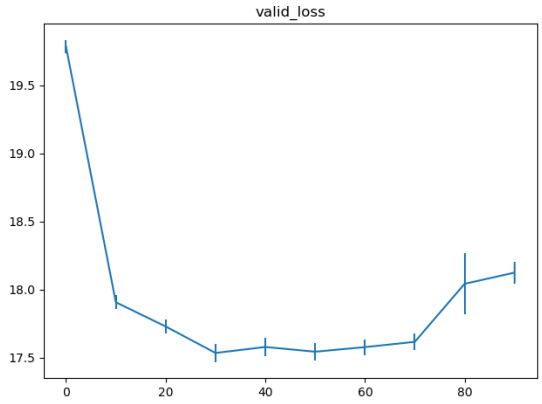


Figure 6. Learning curve for CEVAE with VampPrior prior on the validation set.

By analyzing them, one can verify that the inclusion of VampPrior makes the model more prone to overfitting.

6. Conclusion

In this work, we analyzed the effectiveness of treating the causal inference problem under the variational autoencoders setup, via treating the latent variables in VAEs as hidden confounders. We were able to reproduce some of the results reported previously in the work of (Louizos et al., 2017). Moreover, we included in this approach another recent result reported in literature (Tomczak & Welling, 2017) and showed that some of the validation metrics improved after this extension.

It is important to notice that, despite the necessity of introducing some modifications in the architectures with respect to the model presented in the reference paper, *i.e.* the activation function of the output layers of the neural networks that predict the parameters of Bernoulli distributions, as described in Figure 2, we were able to reproduce fairly closely the results reported originally.

Besides that, even with relevant differences between Edward and Pyro, for instance Edward computes analytically the KL divergence term in the ELBO whenever the prior and posterior are Gaussians, results obtained with our implementation using Pyro, which employs stochastic variational inference, were consistent.

Another important result to highlight is that the inclusion of VampPrior improved test ATE values. However, other evaluated metrics were worse. In fact, by analyzing the learning curves provided in Figures 3, 4, 5, and 6, one clearly see an overfitting behavior after the inclusion of VampPrior.

In terms of future work, we intend to test our implementation using stochastic variational inference on other data, *e.g.* the Twins dataset analyzed in the reference paper.

Also, once we observed that different priors have impact in the performance, it is important to investigate and compare different pseudo-inputs generation schemes for VampPrior in order to avoid overfitting and to test other schemes proposed in literature to be able to provide richer posteriors such as inverse autoregressive flows (Kingma et al., 2016).

References

Cai, Zhihong and Kuroki, Manabu. On identifying total effects in the presence of latent variables and selection bias. *arXiv preprint arXiv:1206.3239*, 2012.

Fuller, WA. Measurement error models, ser. series in probability and mathematical statistics, 1987.

Greenland, Sander, Robins, James M, and Pearl, Judea. Confounding and collapsibility in causal inference. *Statistical science*, pp. 29–46, 1999.

Griliches, Zvi and Hausman, Jerry A. Errors in variables in panel data. *Journal of econometrics*, 31(1):93–118, 1986.

Hill, Jennifer L. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

Hoffman, Matthew D and Johnson, Matthew J. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, 2016.

J. D. Angrist, J.-S. Pischke. Mostly harmless econometrics: An empiricists companion. *Princeton University press*, 2008.

Kingma, Diederik P and Welling, Max. Stochastic gradient vb and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR*, 2014.

Kingma, Diederik P, Salimans, Tim, and Welling, Max. Improving variational inference with inverse autoregressive flow. *arXiv preprint arXiv:1606.04934*, 2016.

Kuroki, Manabu and Pearl, Judea. Measurement bias and effect restoration in causal inference. *Biometrika*, 101 (2):423–437, 2014.

Lamb, Alex, Dumoulin, Vincent, and Courville, Aaron. Discriminative regularization for generative models. *arXiv preprint arXiv:1602.03220*, 2016.

Louizos, Christos, Shalit, Uri, Mooij, Joris, Sontag, David, Zemel, Richard, and Welling, Max. Causal effect inference with deep latent-variable models. *arXiv preprint arXiv:1705.08821*, 2017.

Miao, Wang, Geng, Zhi, and Tchetgen, Eric Tchetgen. Identifying causal effects with proxy variables of an unmeasured confounder. *arXiv preprint arXiv:1609.08816*, 2016.

Pearl, Judea. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

Pearl, Judea. *Causality*. Cambridge university press, 2009.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Rothman, Kenneth J, Greenland, Sander, and Lash, Timothy L. *Modern epidemiology*. Lippincott Williams & Wilkins, 2008.

Sekhon, Jasjeet S. The neyman-rubin model of causal inference and estimation via matching methods. *The Oxford handbook of political methodology*, 2, 2008.

Selén., J. Adjusting for errors in classification and measurement in the analysis of partly and purely categorical data. *Journal of the American Statistical Association*, pp. 81(393):7581,, 1986.

Shalit, Uri, Johansson, Fredrik, and Sontag, David. Estimating individual treatment effect: generalization bounds and algorithms. *arXiv preprint arXiv:1606.03976*, 2016.

Tomczak, Jakub M and Welling, Max. Vae with a vampprior. *arXiv preprint arXiv:1705.07120*, 2017.

Wooldridge, Jeffrey M. On estimating firm-level production functions using proxy variables to control for unobservables. *Economics Letters*, 104(3):112–114, 2009.