Adjusting for Confounders with Invariant Representation Learning

Carina Schnuck
cschnuck@student.ethz.ch

February 2021

1 Introduction

The problem of estimating an unbiased causal effect is inevitable for experiments in a wide range of fields. Thus, an equally wide range of methods has been proposed to tackle this problem including but certainly not limited to methods using machine learning. Being more specific, the goal often is to predict a causal effect from a treatment variable D on some outcome variable Y. In these settings one usually has to adjust for confounders (which are assumed to be observed in the following) in order to attain unbiased estimates of the causal effect. Classical approaches like propensity score matching [13] rely on correctly specifying the underlying model, thus, the validity of such methods is scrutinized if the specification is far from the true model. Similarly, there might not always be a proper instrument available to use for a two-stage regression and estimating the causal effect of treatment D on outcome Y remains a non-trivial task. Moreover, these classical methods are not directly capable of capturing nonlinear relationships between confounders and treatment or outcome variable. To solve this issue several approaches using machine learning have been introduced (see section 2).

Here, I will investigate the possibility of using invariant feature representations obtained from a variant of a Variational Autoencoder (VAE) [8] to adjust for confounders in a regression setting. The approach followed here is based on [11] and will be referrd to as Invariant *VAE* (*IVAE*) in the following. The idea is to use the latent representations found by *IVAE* to predict treatment and outcome variable and then perform a second stage regression with the attained predictions. Using a *VAE* generally allows to assume a prior on the latent variables. *IVAE* leverages this by using a prior that enforces independence between the latent representation of treatment variable D or outcome variable Y and the confounders while at the same time compressing information about D or Y.

Closest to the method proposed here is Double Machine Learning (*DML*) [1]. Both *DML* and *IVAE* aim to find that variation in treatment and outcome variable independent of confounders and then use these for a regression in a second step. The two differ in the way these confounder independent variations are found: *DML* uses the residual from the predictions of a first step machine learning model while *IVAE* uses the latent variable from a *VAE* as proposed in [11] to attain confounder invariant treatment and outcome variable. When comparing the advantages and disadvantages of both methods, *DML* has the advantage that it allows almost any kind of machine learning tool for the first stage prediction while *IVAE* is tied to a *VAE* in the first stage. Further, for *DML* there already exist some statistical guarantees on the consistency of the estimator [1], thus, theoretical ground work for convergence bounds has already been made. This is in contrast to *VAE*'s where guarantees for the consistency of inference are not yet established or only for global variables but not for the latent variables themselves [12, 15] (which is what would be needed here). The use of a *VAE* could on the other hand be promising in terms of interpretation and data generation. If *IVAE* is successful it can find latent representations that compress information contained in treatment and outcome variable which could be useful for further analysis e.g. for data visualization. The use of a *VAE* to find latent representations also gives rise to the possibility to draw samples from the latent space. In *IVAE* this could be interesting since one could sample from the latent space and then vary the confounder to see how this influences the reconstruction (for details on the model architecture of *IVAE* see section 3.1). However, whether there is a real use case for data generation still remains in question. Further, the results in section 5 suggest that *IVAE* is not yet able to produce unbiased estimates of the causal effect in a regression setting.

2 Literature Review

Past literature relevant for this project can be separated into two parts. On one hand, there is literature on machine learning methods used to adjust for confounders, on the other hand, there is literature on invariant feature learning. While the first can be located in the intersection of machine learning and social sciences, the second has been discussed in the machine learning universe solely. Thus, to my knowledge invariant feature learning has not been applied to estimate causal effects in a social science setting yet and constitutes a novel method.

A straight forward approach to use machine learning to adjust for confounders is direct estimation of the outcome model where a machine learning model is used to estimate the Average Treatment Effect (ATE) given the treatment and possible confounders (e.g. [3] use Bayesian regression tree models to estimate causal effects). However, asymptotic properties and construction of valid confidence intervals are not well understood for these methods. Further, a direct estimation of the outcome model using machine learning gives rise to regularization bias due to the necessity to use regularization with these methods. On the contrary, doubly robust methods, which combine characteristics of propensity score matching and direct outcome modeling, aim to mitigate bias from both confounders and regularization. A very prominent example of a doubly robust method is the above mentioned *DML* estimator as proposed in [1] which uses machine learning models to estimate the dependency between treatment and confounders as well as outcome and confounders. Inspired by *DML*, in this project a different method from machine learning, invariant feature representation, is used to predict those variations in treatment and outcome that are invariant of (observed) confounders in order to eventually estimate an unbiased causal effect.

Invariant feature learning originally aims to learn a model that ignores certain nuisance factors in order to achieve better generalization. A typical example would be to enforce a model to ignore the rotation of handwritten numbers in a classification task where the labels are the ground truth numbers. Two approaches have mainly been used to achieve this goal; namely modifications of *VAE*'s and adversarial models. The use of *VAE*'s to find invariant latent representations is closely tied to the literature on disentanglement of *VAE*'s [4, 10] since invariance is often achieved by assuming a prior on the latent variables that explicitly penalizes dependency among latent variables or groups of those. Examples for the use of *VAE*'s to find latent representations with respect to some "outside" variable include [11] who propose to include a penalty on the mutual information between a latent encoding of input features and some outside variable (the confounders in our case) in the optimization problem of an autoencoder and [2] who introduce a novel restriction on the factorization of the prior in a variational autoencoder and add appropriate penalties to the training objective. On the other hand, using adversarial training, a series of papers [6, 7, 5], seeks to learn a latent representation of the input variable that can be split into a part that is predictive of the outcome and another part that is predictive of the noise variables (or again confounders in our case). Regardless of the approach used to attain invariance the idea would be to find representations of the treatment variable D and outcome variable Y invariant of confounders X. These invariant representations can then be used in a second stage regression to attain an unbiased estimate of the underlying causal effect. In the following the focus will be on *VAE*'s since they are generally more convenient to train than adversarial models and easier to interpret. Since estimating causal effects is largely done by non-machine learning experts this can be a crucial advantage of *VAE*'s.

3 Methods

In this section, first, the assumptions of the underlying causal econometric model will be stated. Subsequently *IVAE*, the proposed method to adjust for confounders in this setting, will be explained in more detail.

Assume there exists a partially linear regression model (similar to the assumptions in *DML*):

$$Y = \theta D + g(X) + U \tag{1}$$

$$D = m(X) + V \tag{2}$$

Simply regressing Y on D would lead to confounder bias. Using a machine learning technique to estimate $\hat{g}(X)$ for direct outcome modeling would lead to regularization bias. With *DML* one would try to learn both $\hat{g}(X)$ and $\hat{m}(X)$ to mitigate confounder and regularization bias. The methods proposed here differ in that they will not try to learn $\hat{g}(X)$ and $\hat{m}(X)$ directly but rather representations of D and Y that are

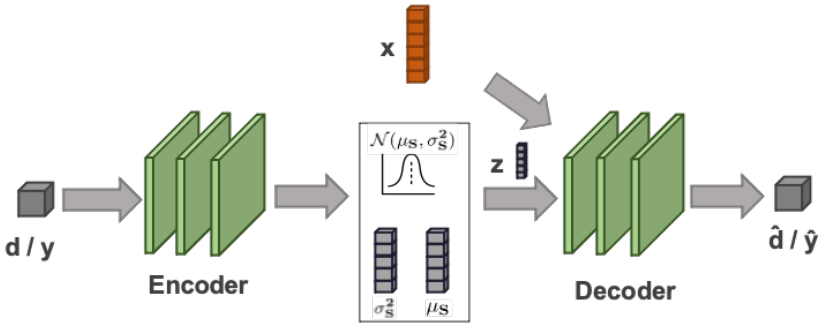


Figure 1: Proposed network architecture using a variational autoencoder. For treatment variable D: Input d is given to the encoder neural network which outputs the posterior mean and covariance of $q(z|d)$ (restricted to be a normal distribution). A sample z is then drawn from $q(z|d)$ and given together with confounder x to the decoder nerual network. The decoder neural network finally outputs the reconstruction $\hat{d}$.

invariant of X (or intuitively speaking do not contain any information from X). Assume D can be fully described by a set of hidden features, such that we can find some (possibly nonlinear) function h that takes as input these hiddn features and outputs D. In the partially linear regression setting (2) assume that we can find two groups of hidden features such that we can assign each hidden feature to either X or V. Hidden features associated with X will be called $\{f_j^D\}$, hidden features not associated with X, thus associated with V in the case of D, will be denoted by $\{g_j^D\}$ respectively. Similarly it is assumed that Y can be fully described by hidden features $\{f_j^Y\}$ and $\{g_j^Y\}$, where $\{f_j^Y\}$ contains variation from X and $\{g_j^Y\}$ from both D and U. One can then represent D and Y as

$$D = h_D^f(\{f_j^D\}) + h_D^g(\{g_j^D\}) \tag{3}$$

$$Y = h_Y^f(\{f_j^Y\}) + h_Y^g(\{g_j^Y\}) \tag{4}$$

Using only $\{g_j^D\}$ to predict D will yield predictions $\tilde{D}$ which contain the variation in D independent from X much like in an instrumental variable estimation. The same procedure is used for Y to get $\tilde{Y}$. Similar to the double machine learning approach in a last step $\tilde{D}$ and $\tilde{Y}$ will be used to estimate θ via regression.

These considerations now pose the question of how to estimate $\tilde{D}$ and $\tilde{Y}$, those parts of variation of the treatment and outcome variable invariant of the confounders. In the following, *IVAE*, a variation of a *VAE*, is described which could be used for estimating $\tilde{D}$ and $\tilde{Y}$ and subsequently adjust for confounders.

3.1 Learning Invariant Feature Representation Using a Variational Autoencoder

In a standard *VAE* setting [8] it is assumed that we can observe a sample of some (usually high-dimensional) variable, in our case for example the treatment variable D (the case for Y is analogous). A realization of D will be denoted with d and has ground-truth probability measure $\tilde{p}(d)$ which the *VAE* aims to approximate by $p(d)$. It is further assumed every d is generated by some latent (possible lower dimensional) variable z, i.e. that $p(d)$ has the decomposition $p(d) = \int p(d|z)p(z)dz$. Conceptually, one would like to maximize the log-likelihood of d, $\log p(d)$, in expectation with respect to the ground-truth probability measure $\tilde{p}(d)$. This would, however, require the marginalization over z which is, in general, infeasible. Thus, the *VAE* approximates the posterior distribution of z, with $q(z|d)$, parametrized by an encoder network, and parametrizes the conditional likelihood of x, $p(d|z)$, by a decoder network. One then aims to maximize a surrogate objective the so-called Evidence Lower Bound (ELBO):

$$\log p(d) \geq -KL[\ q(z|d)\ \|\ p(z)\ ] + \mathbb{E}_{z\sim q(z|d)}[\log p(d|z)] \tag{5}$$

where the first term corresponds to the KL-divergence between the posterior of z and its prior $p(z)$ and the second to a reconstruction loss. In practice one often introduces a trade-off parameter β that weighs the importance of the KL-divergence term. Higher β corresponds to higher penalty of divergence of the posterior $q(z|x)$ from the prior $p(z)$. If one assumes a prior that models the individual components

independent of each other, e.g. through a standard normal, higher β enforces independence between the latent variables [4]. One then arrives at the almost identical ELBO objective of the β-VAE[4]:

$$\log p(d) \geq -\beta KL[\, q(z|d) \parallel p(z) \,] + \mathbb{E}_{z\sim q(z|d)}[\log p(d|z)] \tag{6}$$

The assumption that an observed variable d is generated by some latent factors z largely coincides with the hidden features in the partially regression setting introduced in section 3. Thus, it seems natural to use a *VAE* to find those hidden features. The only thing missing now is the invariance of the latent variables z to the confounders X. It turns out that an additional assumption on the prior $p(z)$ together with a mutual information penalty can be used to approximate this.

[11] propose a variational autoencoder that aims to find latent embeddings of input features that are uninformative of some other variable but still useful for the reconstruction task at hand. In our setting this would e.g. mean that we learn some latent representation z from D that is uninformative of X but still useful to reconstruct D itself (and similar for Y). [11] impose this requirement that the latent encoding z be uninformative of X by enforcing that mutual information $I(z, X)$ be instead low. They aim to achieve this by introducing a corresponding penalty on the mutual information in the variational autoencoder setting and further pose a prior on z that assumes independence from X. They then show that this can be formulated as a variational autoencoder that encodes D and additionally gives X as an input to the decoder (see Figure 1) with the following variational bound to be maximized:

$$\begin{aligned}&\mathbb{E}_{(d,x)}[\log p(d|x)] - \lambda I(z,x) \\ &\qquad \geq \mathbb{E}_{(d,x)}\left[-\beta KL[\, q(z|d) \parallel p(z) \,] - \lambda KL[\, q(z|d) \parallel q(z) \,] + (1+\lambda)\, \mathbb{E}_{z\sim q}[\log p(d|z,x)]\right]\end{aligned} \tag{7}$$

where $q(z|d)$ is the posterior of z parametrized by the encoder, $p(z)$ the prior on z, $q(z)$ the marginal posterior of z, and $p(d|z, x)$ is the likelihood of x parametrized by the decoder. A crucial point is that the prior on z, $p(z)$, is assumed to be independent from X. Importantly, this is not a mere assumption but an active enforcement to make the posterior of z really independent of x. Thus, the penalty on the mutual information and the independence assumption on the prior are crucial for obtaining a representation of D invariant of X. The underlying idea is, if the decoder receives the information contained in X anyway and the latent encodings z learned from D are independent of X then these latent encodings should only contain information needed to reconstruct D not included in X i.e. are invariant of X. Thus, in a second step we use a (possibly simple) machine learning model to predict D from z[1], resulting in predictions $\tilde{D}$. If the *VAE* was successful in finding latent representations z that do not contain information from X then also all predictions obtained only from z should not contain any information from X. Applying the same procedure on Y, invariant predictions $\tilde{Y}$ are constructed. In a last step, $\tilde{Y}$ is regressed on $\tilde{D}$ to get an unbiased estimate of θ[2].

4 Datasets

To evaluate the proposed method two datasets are used. The first is a simulated dataset the second the NLSY dataset on wages of union and non-union workers.

The simulated dataset (see Figure 2a) with $n = 10000$ samples is generated with the following procedure for every D_i, X_i, Y_i with $i \in \{1, ..., n\}$:

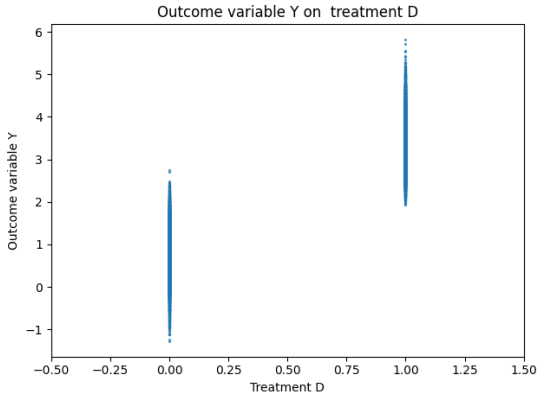
1. To generate D_i and X_i, the factors of variation $\{f_j^D\}_{j=1}^5$ and $\{g_j^D\}_{j=1}^3$ are drawn from a uniform distribution. D_i is then calculated as a sum of additively separable nonlinear functions of $\{f_j^D\}_{j=1}^3$ and $\{g_j^D\}_{j=1}^5$ plus an independent noise term $\epsilon^D \sim \mathcal{N}(0, 0.04)$:

$$D_i = \sum_{j=1}^{5} h_j^D(f_{ji}^D) + \sum_{j=1}^{3} \tilde{h}_j^D(g_{ji}^D) + \epsilon_i^D$$

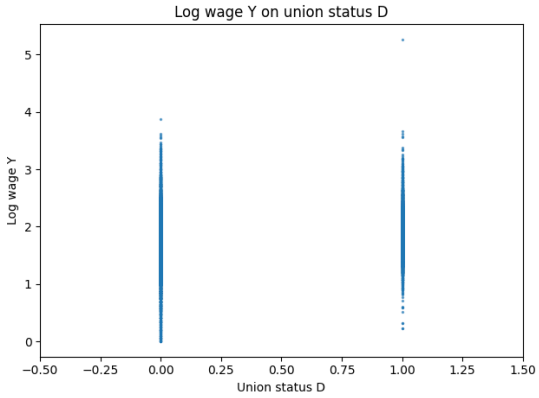
   where $h_j^D, \tilde{h}_j^D$ are either a polynomial of up to order 2 or the logarithm. X_i is assumed to be equal to the factors of variations $\{f_j^D\}_{j=1}^5$ themselves, i.e. $X_i \in \mathbb{R}^5$.

[1] Technically, one does not use the sampled latent encodings z to predict D in the second step since they contain unnecessary noise from the posterior distribution. For this reason the mean of the posterior distribution $q(z|d)$ is used to predict D and the mean of $q(z|y)$ is used to predict Y.

[2] Train test split of the dataset follows that of *DML*.



(a) Outcome variable Y on treatment variable D for simulated data set.



(b) Log wage Y on union member status D for NLSY data set.

Figure 2

2. D is then transformed into a binary variable. For that purpose it is first standardized by subtracting the sample mean and dividing by its sample standard deviation. D_i greater than or equal to zero will then be encoded as 1 and D_i smaller than zero will then be encoded as 0. This way the dataset will be roughly balanced.

3. Y_i is then a sum of a linear function of D_i and additively separable nonlinear functions of $\{f_j^Y\}_{j=1}^3$ plus an independent noise term $\epsilon^Y \sim \mathcal{N}(0, 0.04)$:

$$Y_i = \theta D_i + \sum_{j=1}^{3} h_j^Y(f_j^X) + \epsilon_i^Y$$

where again h_j^Y is either a polynomial of up to order 2 or the logarithm. The causal effect θ is set to 4.

The number of factors of variation and the nonlinear functions are chosen rather arbitrarily and could certainly be initialized in a different way. However, the results of section 5 do not significantly change when varying the number of factors but holds for a wider range. Thus, in the following only the specification from above will be analysed. The use of a simulated dataset allows to compare the estimate of the causal effect by the proposed method with the true effect θ and thus, allows to analyse the unbiasedness of the method. Further, since the number of factors not associated with X are known for D and Y this can be incorporated into the architecture of the *IVAE* by specifying three- and one-dimensional latent factors for D and Y such that an ideal environment for the *IVAE* is created. In practice, of course, one would not know the number of latent factors beforehand and thus this number should be treated as a tunable hyperparameter. The benefit of knowing the number of latent factors lies in the fact that if the *IVAE*, even with a correct specification of latent variables, cannot find good latent representations it can be ruled out that this is due to a false specification of the latent dimension.

For completeness the proposed method also is evaluated on a NLSY dataset to estimate a causal effect from workers' union status on wages[3] (see Figure 2b). The dataset contains information on the log wages of young working women aged 14 to 26. Beside age the dataset also contains information whether a worker is member of a union or not. The union status and log wage will be treated as the treatment variable D and outcome variable Y respectively. The dataset further contains information on the individual workers of which we will use the following as confounders to adjust for: `age, year, race, msp, collgrad, nev_mar, not_smsa`, and `south` [4].

The NLSY dataset offers a convenient way to test the proposed method since it has already been employed in the context of *DML*, hence results can be readily compared.

5 Results

While the previous sections have given a motivation as to why the use of a *VAE* might work in the context of finding representations of outcome and treatment variable invariant of confounders that can

[3]The NLSY dataset is available here: http://www.stata-press.com/data/r10/nlswork.dta
[4]For further details on the dataset and chosen variables see: https://rdrr.io/rforge/sampleSelection/man/nlswork.html

then be used to estimate an unbiased causal effect, this section will show that a *VAE* will not work in this simple regression setting, at least not if implemented as proposed in [11]. This section, thus aims to give an understanding as to why this is the case. It will be shown that although *IVAE* is able to improve on its training objective (7) it is still limited by the low dimensionality of the data it is trained on. Although of course a *VAE* parametrized by a neural net, can model highly complex functions, the complexity of a given trained neural network still is tied to the complexity of the input feature space of the training data, which is determined by the one-dimensional treatment variable D and one-dimensional output variable Y.

The specification of the *VAE* architecture used for all plots in this section is indicated in Table 1. Further a batch size of 132, learning rate of 10^{-4}, λ penalty of 1, and β penalty of 10 was used. The dimension of the latent variables is chosen to be 2, mainly for visualization reasons. A higher number of latent variables or even the correct number of latent variables as outlined in section 4 do not alter the results. To predict $\tilde{Y}$ from the latent embedding learned with the *IVAE* a multi-layer perceptron (MLP) with two hidden layers of size 100 and 50, and a L2-penalty of 10^{-4} was used. To predict $\tilde{D}$ simple OLS regression was used. Further, an MLP with size 20 and 10 and L2-penalty of 10^{-4} was used as the first stage machine learning model in the *DML* method that is used for comparison. Although only shown for one specific set of parameters and architectures, the results laid out here hold for a very wide range of network specifications and loss penalty parameters. Indeed, I was not able to find any combination that would give results differing from the ones shown here. Specifically, I tested a grid search for loss parameters β and λ over $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\} \times \{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$ with results very much the same. This is not surprising given the comments and thoughts considered below and in section 6.

As seen in Figure 3, 4, 5 and 6 *IVAE* is able to improve on its training objective for both the simulated and NLSY dataset. The reconstruction loss ($\mathbb{E}_{z\sim q}[\log p(d|z,x)]$ in 7) and both the KL divergence to the prior ($KL[\, q(z|d) \parallel p(z_D)\,]$ in 7) and to the marginal posterior ($KL[\, q(z|d) \parallel q(z)\,]$ in 7) decrease during training. Figure 3, 4, 5 and 6 *IVAE* also suggest they converge after a maximum of 250 epochs for D and Y on both the simulated and NLSY dataset. Figure 7, however, shows that for the simulated dataset the estimated coefficient by *IVAE* is almost identical to the OLS estimate of simply regressing Y on D, meaning it is biased when confounders exist. For the NLSY dataset an assessment of the estimated causal effect is less clear. Figure 7 shows that the estimated coefficient associated with *IVAE* lies somewhere between the OLS and *DML* estimate albeit closer to *DML*.

Overall, *IVAE* seems to be unable to consistently estimate an unbiased causal effect. The reason for this might be seen in Figure 8 and 9. Although the neural network that parametrizes the encoder of *IVAE* may be very rich in terms of modeling capacity which gave rise to the hope that *IVAE* can actually find good latent representations of D and Y, (in the best case even latent representations close to the true factors of variation from which the data was generated) it can be seen that the latent encodings collapse to two points for D and that the two dimensions of the latent representations of Y are a nonlinear function of each other. The question to answer now is whether this collapse is due to some misspecification within the training framework or whether this is a pathological characteristic of the proposed method. Unfortunately, it seems that the latter is the true reason. Since the encoder of *IVAE* only receives the information from a single observation of D or Y the output is limited to this information. In the case of D this means that the encoder, regardless of its size, will always only be able to find two different means for the posterior distribution $q(z|d)$ over the whole dataset, one mean for $d = 1$ and one mean for $d = 0$. Thus, the means found by *IVAE* are nothing else than an embedding of 0 and 1 in a higher dimension (the dimension of the latent representation), consequently all samples with $d = 1$ will have the same mean and all samples with $d = 0$ will have a different but also constant mean. Thus, *IVAE* actually cannot learn latent representations of D invariant to X since the latent representation of D contains exactly the same information as D (just embedded in a possibly higher dimension). A similar argument holds for Y which explains why the regression using $\tilde{D}$ and $\tilde{Y}$ yields almost exactly the same results as a simple OLS regression of Y on D.

6 Conclusion

This project raises the questions whether *VAE*'s can be used to estimate an unbiased causal effect in a regression setting. In my opinion three problems have been encountered: low dimension of the treatment and outcome variable, enforcing independence within the latent representations, and absence of general guarantees with respect to *VAE*'s.

The core problem of using *IVAE* to decompose treatment and outcome variable into different factors

of variation is that the encoder of *IVAE* only receives as input a single scalar. Hence, the information contained in the learned latent representation can never contain more information for one single data point than that of a scalar. This problem is amplified by the fact that the treatment variable is a binary one and thus by construction the *VAE* can only find to different posterior means, i.e. two different latent embeddings. Thus, the alternative approach [2] mentioned in section 2 will suffer the same problem since the encoder in [2] also only receives D or Y as input. To fix this the encoder of the *VAE* would somehow also need to take into account the information contained in the confounders to actually find representations of D or Y invariant of X. How exactly the architecture and training objective of such a *VAE* would look like remains an open question and it still needs to be seen whether such an approach exists and whether it can yield unbiased estimates.

The second question that remains open is what kind of objective can effectively enforce independence among latent variables in the *VAE*. Beside the literature on disentanglement of *VAE*'s [4, 10] it could be worthwhile also considering other nonparametric independence measures [9, 14] to be used as an additional penalty term in the loss function. In this context one can also raise the question of what invariance actually means and if it is really equivalent to independence.

The last point is related to the question of how to exactly define independence. In causal regression it has been crucial that methods are provably unbiased and that one can draw conclusions on convergence at least in the infinite sample case. In a *VAE* this is hard for two reasons: first, as mentioned above the notion of independence among latent variables is not clearly defined and proposed methods often only intuitively motivate their choice of independence penalty but rigorous analysis of achieved independence is rare. Further, as already mention in section 1 guarantees for the consistency on inference on the latent variables in *VAE*'s are not yet established.

7 Remarks

Code for this project can be found on the github repository: https://github.com/cfschnuck/invariance-in-econ.

References

[1] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, and Whitney Newey. Double/debiased/neyman machine learning of treatment effects, 2017.

[2] Elliot Creager, David Madras, Jörn-Henrik Jacobsen, Marissa A. Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. Flexibly fair representation learning by disentanglement, 2019.

[3] P. Richard Hahn, Jared S. Murray, and Carlos Carvalho. Bayesian regression tree models for causal inference: regularization, confounding, and heterogeneous effects, 2019.

[4] I. Higgins, Loïc Matthey, A. Pal, C. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[5] Ayush Jaiswal, Daniel Moyer, Greg Ver Steeg, Wael AbdAlmageed, and Premkumar Natarajan. Invariant representations through adversarial forgetting, 2019.

[6] Ayush Jaiswal, Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Unsupervised adversarial invariance, 2018.

[7] Ayush Jaiswal, Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Unified adversarial invariance, 2019.

[8] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

[9] Romain Lopez, Jeffrey Regier, Michael I. Jordan, and Nir Yosef. Information constraints on auto-encoding variational bayes, 2018.

[10] Emile Mathieu, Tom Rainforth, N. Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders, 2019.

[11] Daniel Moyer, Shuyang Gao, Rob Brekelmans, Greg Ver Steeg, and Aram Galstyan. Invariant representations without adversarial training, 2019.

[12] Debdeep Pati, Anirban Bhattacharya, and Yun Yang. On statistical optimality of variational bayes, 2017.

[13] Paul Rosenbaum and Donald Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70:41–55, 04 1983.

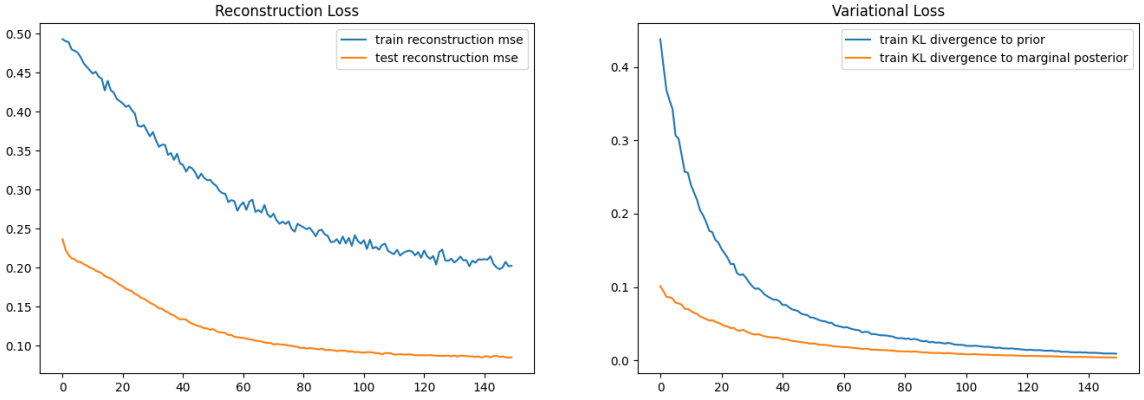
[14] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning, 2018.

[15] Yixin Wang and David M. Blei. Frequentist consistency of variational bayes. *Journal of the American Statistical Association*, 114(527):1147–1161, Aug 2018.

A Results: Tables and Figures

Encoder	**Decoder**
Layers: FC(1,16), DO, NL, BN, FC(16, 16), DO, NL mean layer: FC(16, z_dim) variance layer: FC(16, z_dim)	Layers: FC(l_dim, 16), DO, NL, BN, FC(16, 8), DO, NL, BN, FC(8, 1) - l_dim is the sum of the dimension of z and x combined - when input variable is D an additional sigmoid layer is added
Dictionary: FC(d_1,d_2) - fully connected layer mapping from $\mathbb{R}^{d_1}$ to $\mathbb{R}^{d_2}$ DO - dropout layer with dropout probability = 0.1 BN - batch normalization layer NL - nonlinear layer: Leaky ReLU	

Table 1: Specification of the encoder and decoder architecture of *IVAE*.

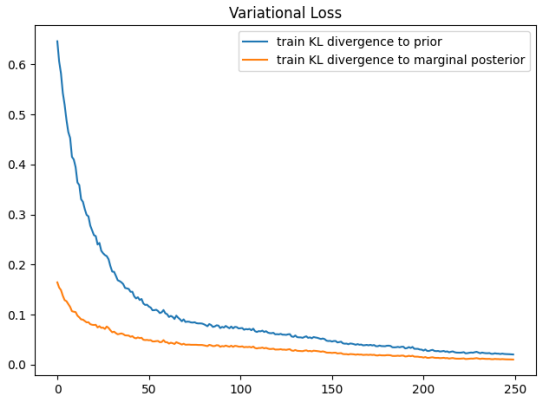


(a) Reconstruction loss on training and test set.

(b) KL-divergence to prior and marginal posterior on train set.

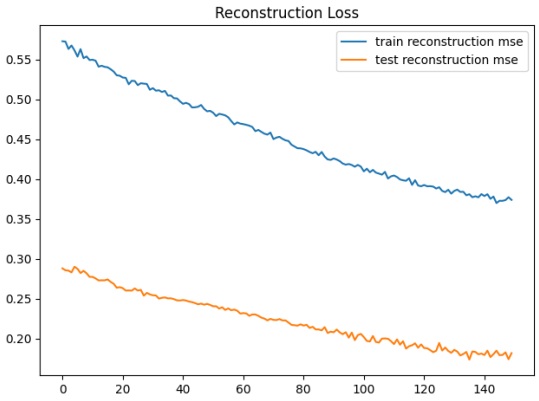
Figure 3: Loss for D on simulated data set.

(a) Reconstruction loss on training and test set.

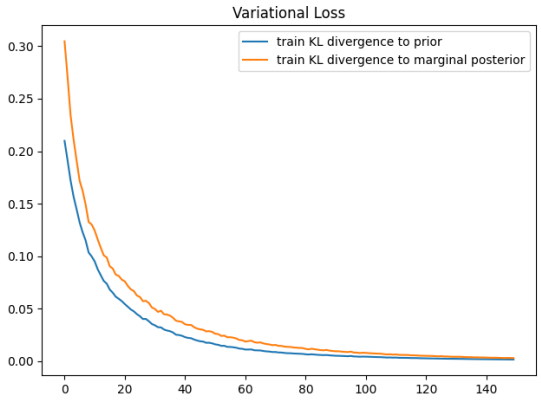


(b) KL-divergence to prior and marginal posterior on train set.

Figure 4: Loss for Y on simulated data set.



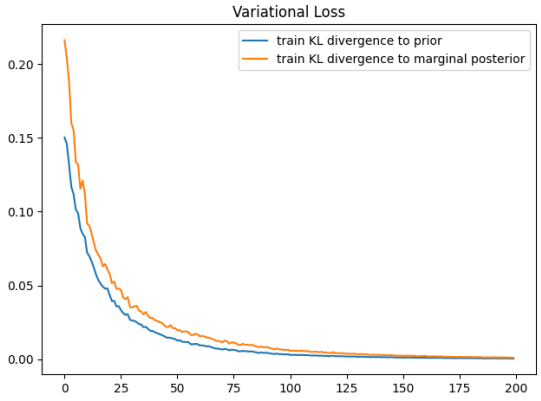
(a) Reconstruction loss on training and test set.



(b) KL-divergence to prior and marginal posterior on train set.

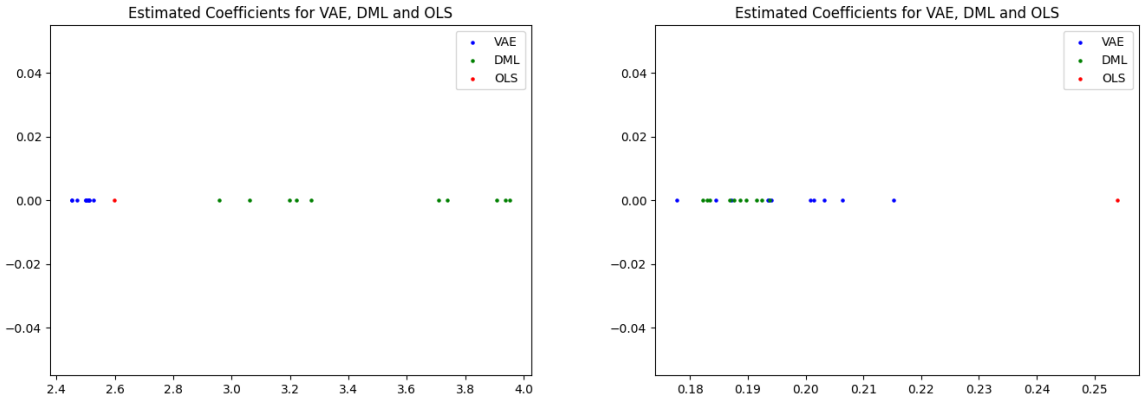
Figure 5: Loss for D on NLSY data set.

(a) Reconstruction loss on training and test set.



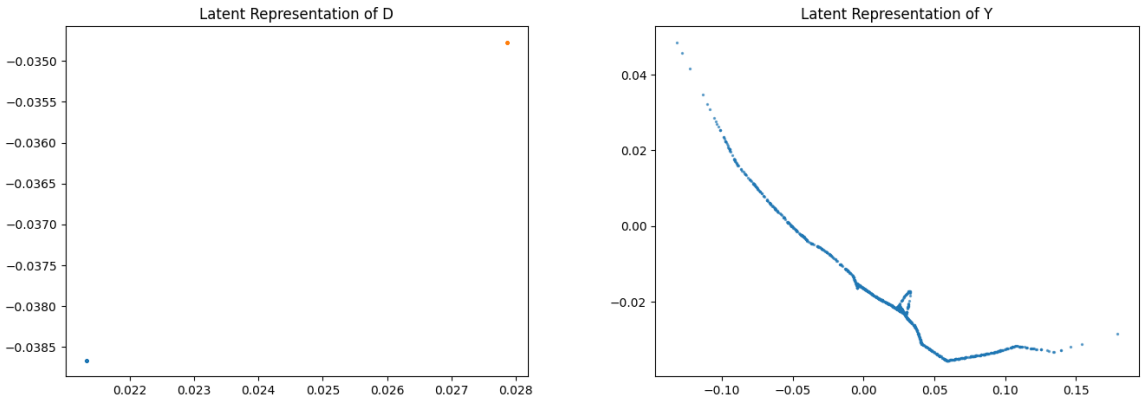
(b) KL-divergence to prior and marginal posterior on train set.

Figure 6: Loss for Y on NLSY data set.



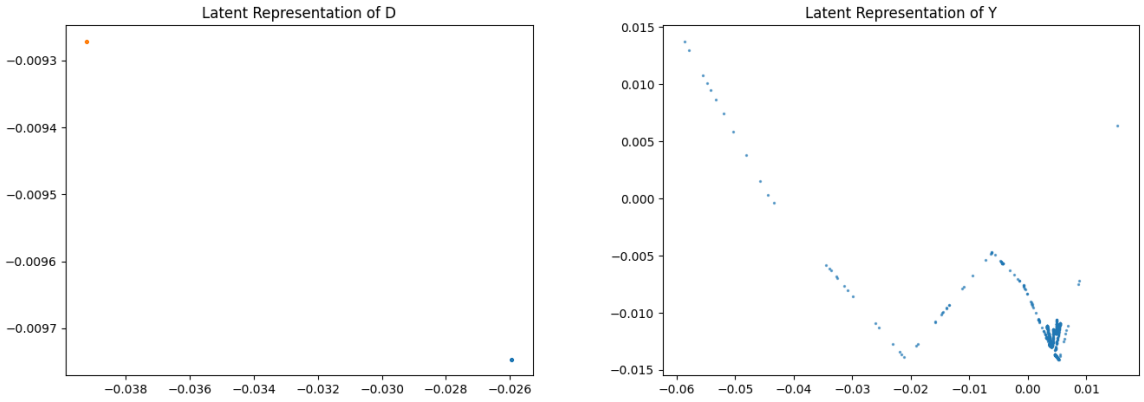
(a) Estimated coefficients for simulated dataset. (b) Estimated coefficients for NLSY dataset.

Figure 7: Estimated coefficients for causal effect.



(a) Latent representations of D. (b) Latent representations of Y.

Figure 8: Latent representations for simulated dataset.



(a) Latent representations of D. (b) Latent representations of Y.

Figure 9: Latent representations for NLSY dataset.