Deep IV: A Flexible Approach for Counterfactual Prediction

Jason Hartford [1] **Greg Lewis** [2] **Kevin Leyton-Brown** [1] **Matt Taddy** [2]

Abstract

Counterfactual prediction requires understanding causal relationships between so-called *treatment* and *outcome* variables. This paper provides a recipe for augmenting deep learning methods to accurately characterize such relationships in the presence of *instrument variables (IVs)*—sources of treatment randomization that are conditionally independent from the outcomes. Our IV specification resolves into two prediction tasks that can be solved with deep neural nets: a first-stage network for treatment prediction and a second-stage network whose loss function involves integration over the conditional treatment distribution. This *Deep IV* framework[1] allows us to take advantage of off-the-shelf supervised learning techniques to estimate causal effects by adapting the loss function. Experiments show that it outperforms existing machine learning approaches.

1. Introduction

Supervised machine learning (ML) provides many effective methods for tasks in which a model is learned based on samples from some data generating process (DGP) and then makes predictions about new samples from the same distribution. However, decision makers would often like to predict the effects of *interventions* into the DGP through *policy changes*. Such changes impact the relationship between inputs and outcomes, making straightforward prediction approaches inappropriate. In order to accurately answer such *counterfactual* questions it is necessary to model the structural (or causal) relationship between policy (or "treatment") and outcome variables.

For example, consider an airline that wants to use historical data to optimize the prices it charges its customers: in this case, price is the treatment variable and the customer's decision about whether to buy a ticket is the outcome. There are two ways that a naive analysis could lead to incorrect counterfactual predictions. First, imagine that price varies in the training data because the airline gradually increases prices as a plane fills. Around holidays, more people want to fly and hence planes become fuller leading to higher prices. So, in our training set we observe examples with high prices and high sales. A direct ML approach might incorrectly predict that if the airline were to increase prices at other times in the year they would also observe increased sales, whereas the true relationship between price and sales is surely negative. Typically we can observe holidays, and include them in the model, so that we can correct for their effects. This case where an observable feature (holidays) is correlated with both the outcome and treatment variable is called "selection on observables".

But say there is also sometimes high demand because of conferences, which the airline is not able to observe. This presents a more challenging problem: even if price distortions due to holidays were taken into account, a naive analysis could conclude that higher prices drive higher demand, whereas in fact high price is only correlated with high demand via the latent conference variable. This is the more challenging case of "selection on unobservables".

The gold standard for establishing causal relationships is to conduct "AB" experiments, with subjects randomly assigned to different values of the treatment variable. In the airline example, this would mean assigning passengers random prices that do not depend on the number of seats sold. Enough data collected in this way would make it straightforward to identify the true relationship between price and demand even if the latent variable were never observed. However, such a strategy would be exceedingly expensive (the airline would sometimes turn away interested passengers for nearly-empty flights and offer steep discounts for nearly-full flights). Depending on the timescale over which such randomization were conducted, it could also hurt the airline's long-term interests (because it could be perceived as unfair) and could fail if passengers were able to hide their identities (e.g., by logging in from a different computer if they didn't like the price).

[1]University of British Columbia, Canada [2]Microsoft Research, New England, USA. Correspondence to: Jason Hartford <jasonhar@cs.ubc.ca>, Matt Taddy <taddy@microsoft.com>.

Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

[1]Implementation with all simulation experiments is available at https://github.com/jhartford/DeepIV

The alternative is to work with observational data, but doing so requires explicit assumptions about the causal structure of the DGP (Bottou et al., 2013). Most recent approaches to using machine learning methods such as trees (Wager & Athey, 2015; Athey & Imbens, 2016) and deep networks (Johansson et al., 2016; Shalit et al., 2016) for causal inference in observational data leverage an "unconfoundedness" assumption that the treatment is conditionally independent of any latent variables given the observed features. This amounts to assuming away selection on unobservables, which may or may not be reasonable depending on the setting.

We can do without this assumption and get around both types of selection if we can identify one or more *instrumental variables (IVs)* that only affect treatment assignment and not the outcome variable. In our airline example, the cost of fuel could be such an instrument: its variation is independent of demand for airline tickets and it affects sales only via ticket prices. Changes in the cost of fuel thus create movement in ticket prices that is *independent* of our latent variable, and this movement is as good as randomization for the purposes of causal inference. See Figure 1 for an graphical illustration of this example and of the general class of causal graphs that we consider.

The IV framework has a long history, especially in economics (e.g., Wright, 1928; Reiersøl., 1945). It provides methods for learning the regression function that relates the treatment and response variables under the "interventional distribution" for DGPs that conform to the graphical model shown in Figure 1 (Pearl, 2009). Most IV applications make use of a two-stage least squares procedure (2SLS; e.g., Angrist & Pischke, 2008) that applies a model of linear and homogeneous treatment effects (e.g., all airline customers must have the same price sensitivity). Nonparametric IV methods from the econometrics literature relax these assumptions (e.g., Newey & Powell, 2003; Darolles et al., 2011). However, these methods typically work by modeling the outcome as an unknown linear combination of a prespecified set of basis functions of the treatment and other covariates (e.g. Hermite polynomials, wavelets, or splines) *and* then modeling the conditional expectation of each of these basis functions in terms of the instruments (i.e., the number of parameters is quadratic in the number of basis functions). This requires a strong prior understanding of the DGP by the researcher; also, the complexity of both specification and estimation explodes when there are more than a handful of inputs.

Advances in deep learning have demonstrated the power of learning latent representations of complex features spaces (for recent surveys, see eg. LeCun et al., 2015; Schmidhuber, 2015). This paper's goal is to use these powerful learning algorithms for IV analysis. We do this by breaking IV analysis into two supervised stages that can each be targeted with deep networks and that, when solved, allow us to make counterfactual claims and perform causal inference. Specifically, we first model the conditional distribution of the treatment variable given the instruments and covariates and then target a loss function involving integration over the conditional treatment distribution from the first stage. Both stages use deep neural nets trained via stochastic gradient descent (Robbins & Monro, 1951; Bottou, 2010). We also present an out-of-sample causal validation procedure for selecting hyper-parameters of the models on a validation set. We refer to this setup as the Deep IV framework.

Section 2 describes our general IV specification and its decomposition into two learning tasks. Section 3 outlines neural network estimation for these tasks with particular attention paid to the SGD routine used in model training and our causal validation procedure. Section 4 presents experimental results that illustrate the benefits of our methods.

2. Counterfactual prediction

We aim to predict the value of some outcome variable y (e.g., sales in our airline example) under an intervention in a policy or treatment variable p (e.g., price). There exists a set of observable covariate features x (e.g., holidays) that we know affect both p and y. There also exist unobservable latent variables e that may affect x, p and y (e.g., conferences). Counterfactual prediction aims to recover $\mathbb{E}[y|\text{do}(p), x]$ in the context of the graphical model given by Figure 1, where the do$(\cdot)$ operator indicates that we have intervened to set the value of the policy variable p (as per Pearl, 2009). We assume the y is structurally determined by p, x and e as

$$y = g(p, x) + e. \tag{1}$$

That is, $g(\cdot)$ is some unknown and potentially non-linear continuous function of both x and p, and we assume that the latent variables (or "error") e enter additively with unconditional mean $\mathbb{E}\,e = 0$. We allow for errors that are potentially correlated with the inputs: $\mathbb{E}[e|x, p] \neq 0$ and, in particular, $\mathbb{E}[pe|x] \neq 0$.

Define the counterfactual prediction function

$$h(p, x) \equiv g(p, x) + \mathbb{E}[e|x], \tag{2}$$

which is the conditional expectation of y given the observables p and x, *holding the distribution of e constant as p is changed*. Note that we condition only on x and not p in the term $E[e|x]$; this term is typically nonzero, but it will remain constant under arbitrary changes to our policy variable p.[2] Thus $h(p, x)$ is the structural equation that we estimate. It is

[2] It may be easier to think about a setting where $e \perp\!\!\!\perp x$, so that the latent error is simply *defined* as being due to factors orthogonal to the observable controls. In that case, $h(p, x) = g(p, x)$. All of our results apply in either setup.

Figure 1. (Left) Our air-travel demand example, with arrows representing causal effects. Price is the policy variable, sales is the outcome, and holidays are observable covariates. There is a big 'conference', unobserved to the policy-maker, that drives demand and (due to the airline's pricing algorithms) price. The instrument is the cost of fuel, which influences sales only via price. (Right) The general structure of the DGP under our IV specification; x represents observable features, p is our treatment variable of interest, z represents the instruments, and latent effects e influence the outcome y additively.

useful because to evaluate policy options (e.g. changing the ticket price from p_0 to p_1) we can look at the difference in outcomes $h(p_1, x) - h(p_0, x) = g(p_1, x) - g(p_0, x)$.

In standard supervised learning settings, the prediction model is trained to fit $\mathbb{E}[y|p, x]$. This will typically be *biased* against our structural objective because

$$\mathbb{E}[y|p, x] = g(p, x) + \mathbb{E}[e|p, x] \neq h(p, x) \tag{3}$$

since our treatment is not independent of the latent errors by assumption and hence $\mathbb{E}[e|p, x] \neq \mathbb{E}[e|x]$. This object is inappropriate for policy analysis as it will lead to biased counterfactuals:

$$\begin{aligned}&\mathbb{E}[y|p_1, x] - \mathbb{E}[y|p_0, x] \\ &\quad = g(p_1, x) - g(p_0, x) + \Big(\mathbb{E}[e|p_1, x] - \mathbb{E}[e|p_0, x]\Big).\end{aligned} \tag{4}$$

In our airline example, high prices during conferences imply that $\mathbb{E}[e|p_1, x] - \mathbb{E}[e|p_0, x]$ will be positive, resulting in the incorrect prediction that higher prices are associated with higher sales if this bias is sufficiently large.

Fortunately, the presence of *instruments* allows us to estimate an unbiased $\hat{h}(p, x)$ that captures the structural relationship between p and y. These are sets of variables z that satisfy the following three conditions.

Relevance $F(p|x, z)$, the distribution of p given x and z, is not constant in z.

Exclusion z does not enter Eq. (1)—i.e., $z \perp\!\!\!\perp y \mid (x, p, e)$.

Unconfounded Instrument z is conditionally independent of the error—i.e., $z \perp\!\!\!\perp e \mid x$.[3]

[3]Under the additive error assumption made in Eq. (1), unconfoundedness of the instrument is not necessary: we could replace this assumption with the weaker mean independence assumption $E[e|x, z] = 0$ without changing anything that follows. We use the stronger assumption to facilitate extensions, e.g. to estimating counterfactual quantiles. Our assumption is similar to the 'unconfoundedness' assumption in the Neyman–Rubin potential outcomes framework (Rosenbaum & Rubin, 1983) (i.e. $p \perp\!\!\!\perp e \mid x$). But our assumption is weaker—in particular, we allow for $p \not\perp\!\!\!\perp e|x$—and so the matching and propensity-score re-weighting approaches often used in that literature will not work here.

Taking the expectation of both sides of Equation (1) conditional on $[x, z]$ and applying these assumptions establishes the relationship (cf. Newey & Powell, 2003):

$$\begin{aligned}\mathbb{E}[y|x, z] &= \mathbb{E}[\, g(p, x)|x, z] + \mathbb{E}[e|x] \\ &= \int h(p, x) dF(p|x, z),\end{aligned} \tag{5}$$

where, again, $dF(p|x, z)$ is the conditional treatment distribution. The relationship in Equation (5) defines an *inverse problem* for h in terms of two directly observable functions: $\mathbb{E}[y|x, z]$ and $F(p|x, z)$. IV analysis typically splits this into two stages: first estimating $\hat{F}(p|x_t, z_t) \approx F(p|x_t, z_t)$, and then estimating $\hat{h}$ after replacing F with $\hat{F}$.

Most existing approaches to IV analysis assume linear models for the treatment density function $\hat{F}$ and the counterfactual prediction function $\hat{h}$ to solve Equation (5) in closed form. For example, the two-stage least-squares (2SLS) procedure (e.g., Angrist et al., 1996) posits $y = \gamma p + x\beta_y + e$ and $p = \tau z + x\beta_p + v$, with the assumptions that $\mathbb{E}[e|x, z] = 0$, $\mathbb{E}[v|x, z] = 0$, and $\mathbb{E}[ev] \neq 0$ (which implies $\mathbb{E}[ep] \neq 0$). This procedure is straightforward: fit a linear model for p given x and z and use the predicted values $\hat{p}$ in a second linear model of y. This is a statistically efficient way to estimate the effect of the policy variable (i.e. γ) as long as two strong assumptions hold: linearity (i.e., both first- and second-stage regressions are correctly specified) and homegeneity (i.e., the policy affects all individuals in the same way).[4]

Flexible nonparametric extensions of 2SLS either replace the linear regressions with a linear projection onto a series of known basis functions (Newey & Powell, 2003; Blundell et al., 2007; Chen & Pouzo, 2012) or use kernel-based methods as in Hall & Horowitz (2005) and Darolles et al. (2011). This system of series estimators is an effective strategy for introducing flexibility and heterogeneity with low dimensional inputs, but the approach faces the same limitations as kernel methods in general: their performance depends on the choice of kernel function; and they often become com-

[4]The estimated $\hat{\gamma}$ remains interpretable as a 'local average treatment effect' (LATE) under less stringent assumptions (see Angrist et al., 1996, for an overview).

putationally intractable in high-dimensional feature spaces $[x, z]$ or with large numbers of training examples.

3. Estimating and validating DeepIV

We now describe how to use deep networks to perform flexible, scalable IV analysis in a framework we call DeepIV. We make two contributions that are each necessary components of the approach. First, we propose a loss function and optimization procedure that allows us to optimize deep networks for counterfactual prediction. Second, we describe a general procedure for out-of-sample validation of two-stage instrument variable methods. This allows us to perform causally valid hyper-parameter optimization, which in general is necessary for achieving good predictive performance using deep networks.

Our approach is conceptually simple given the counterfactual prediction framework described in Section 2. Rather than constraining ourselves to analytic solutions to the integral in Equation (5), we instead directly optimize our estimate of the structural equation, $\hat{h}$. Specifically, to minimize ℓ_2 loss given n data points and given a function space $\mathcal{H}$ (which may not include the true h), we solve

$$\min_{\hat{h}\in\mathcal{H}} \sum_{t=1}^{n} \left(y_t - \int \hat{h}(p, x_t) dF(p|x_t, z_t) \right)^2. \tag{6}$$

Since the treatment distribution is unknown, we estimate $\hat{F}(p|x, z)$ in a separate first stage.

So the DeepIV procedure has two stages: a first stage density estimation procedure to estimate $\hat{F}(p|x, z)$ and a second that optimizes the loss function described in Equation (6). In both stages hyper-parameters can be chosen to minimize the respective loss functions on a held out validation set, and improvements in performance against this metric will correlate with improvements on the true structural loss which cannot be evaluated directly. We briefly discuss these two stages before describing our methods for optimizing the loss given in Equation (6) and our causal validation procedure.

First stage: Treatment network In the first stage we learn $F(p|x, z)$ using an appropriately chosen distribution parameterized by a deep neural network (DNN), say $\hat{F} = F_\phi(p|x, z)$ where ϕ is the set of network parameters. Since we will be integrating over F_ϕ in the second stage, we must fully specify this distribution.

In the case of discrete p, we model $F_\phi(p|x, z)$ as a categorical $\text{Cat}(p \mid \pi(x, z; \phi))$ with $\text{p}(p = p^k) = \pi_k(x, z; \phi)$ for each treatment category p^k and where $\pi_k(x, z; \phi)$ is given by a DNN with softmax output. For continuous treatment, we model F as a mixture of Gaussian distributions where component weights $\pi_k(x, z; \theta)$ and parameters $[\mu_k(x, z; \phi), \sigma_k(x, z; \phi)]$ form the final layer of a neural network parametrized by ϕ. This model is known as a mixture density network, as detailed in §5.6 of Bishop (2006). With enough mixture components it can approximate arbitrary smooth densities. To obtain mixed continuous–discrete distributions we replace some mixture components with point masses. In each case, fitting F_ϕ is a standard supervised learning problem.

Second stage: Outcome network In the second stage, our counterfactual prediction function h is approximated by a DNN with real-valued output, say h_θ. We optimize network parameters θ to minimize the integral loss function in Equation (6) over training data D of size $T = |D|$ from the joint DGP $\mathcal{D}$,

$$\mathcal{L}(D; \theta) = |D|^{-1} \sum_t \left(y_t - \int h_\theta(p, x_t) d\hat{F}_\phi(p|x_t, z_t) \right)^2. \tag{7}$$

Note that this loss involves the *estimated* treatment distribution function, $\hat{F}_\phi$, from our first stage.[5]

3.1. Optimization for DeepIV networks

We use stochastic gradient descent (SGD; see algorithms in, e.g., Duchi et al., 2011; Kingma & Ba, 2014) to train the network weights. For F_ϕ, standard off-the-shelf methods apply, but second stage optimization (for h_θ) needs to account for the integral in Equation (7). SGD convergence only requires that each sampled gradient $\nabla_\theta \mathcal{L}_t$ is *unbiased* for the population gradient, $\nabla_\theta \mathcal{L}(\mathcal{D}; \theta)$. Lower variance for $\nabla_\theta \mathcal{L}_t$ will tend to yield faster convergence (Zinkevich, 2003) while the computational efficiency of SGD on large datasets requires limiting the number of operations going into each gradient calculation (Bousquet & Bottou, 2008).

We can approximate the integral with respect to a probability measure with the average of draws from the associated probability distribution: $\int h(p) dF(p) \approx B^{-1} \sum_b h(p_b)$ for $\{p_b\}_1^B \overset{iid}{\sim} F$. Hence we can get an unbiased estimate of Equation (7) by replacing the integral with a sum over samples from our fitted treatment distribution function, $\hat{F}_\phi$:

$$\mathcal{L}(D; \theta) \approx |D|^{-1} \sum_t \left(y_t - \frac{1}{B} \sum_{\dot{p}\sim\hat{F}_\phi(p|x_t,z_t)} h_\theta(\dot{p}, x_t) \right)^2 := \hat{\mathcal{L}}(D; \theta). \tag{8}$$

This equation can be used to estimate $\nabla_\theta \mathcal{L}$ with an important caveat: if we want to maintain unbiased gradient estimates, *independent* samples must be used for each instance of the integral in the gradient calculation. To see this, note that the

[5] We can replace Eq. (7) with other functions, e.g., a softmax for categorical outcomes, but use ℓ_2 loss for most of our exposition.

gradient of Equation (8) has expectation

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\nabla_\theta\mathcal{L}_t &= -2\mathbb{E}_{\mathcal{D}}\Big(\mathbb{E}_{F_\phi(p|x_t,z_t)}\Big[y_t - h_\theta(p^k,x_t)\Big] \\ &\qquad \cdot \mathbb{E}_{F_\phi(p|x_t,z_t)}\Big[h'_\theta(p^k,x_t)\Big]\Big) \qquad (9)\\ &\neq -2\mathbb{E}_{\mathcal{D}}\mathbb{E}_{F_\phi(p|x_t,z_t)}\Big[\Big(y_t - h_\theta(p^k,x_t)\Big)h'_\theta(p^k,x_t)\Big],\end{aligned}$$

where the inequality holds so long as $\mathrm{cov}_{F_\phi(p|x_t,z_t)}\Big[\Big(y_t - h_\theta(p^k,x_t)\Big)h'_\theta(p^k,x_t)\Big] \neq 0$. We thus need a gradient estimate based on unbiased MC estimates for each $\mathbb{E}_{F_\phi(p|x_t,z_t)}$ term in Equation (9). We obtain such an estimate by taking two samples $\{\dot{p}_b\}_1^B, \{\ddot{p}_b\}_1^B \overset{iid}{\sim} F_\phi(p|x_t,z_t)$ and calculating the gradient as

$$\widehat{\nabla}^B_\theta\mathcal{L}_t \equiv -2\left(y_t - B^{-1}\sum_b h_\theta(\dot{p}_b,x_t)\right)B^{-1}\sum_b h'_\theta(\ddot{p}_b,x_t). \quad (10)$$

Independence of the two samples ensures that $\mathbb{E}\widehat{\nabla}^B_\theta\mathcal{L}_t = \mathbb{E}_{\mathcal{D}}\nabla_\theta\mathcal{L}_t = \nabla_\theta\mathcal{L}(\mathcal{D};\theta)$, as desired. The variance of our estimate depends on B, the number of samples that we draw. Each of these samples is relatively expensive to compute because they require a forward pass through the network $\hat{h}_\theta(\dot{p},x_t)$. If this varies significantly with $\dot{p}$ we might need a large number of samples to get a low-variance estimate of the gradient, which is computationally intensive.

An alternative is to optimize an upper bound on Equation (8). By using Jensen's inequality and the fact that the squared error function is convex we get that

$$\hat{\mathcal{L}}(D;\theta) \leq |D|^{-1}\sum_t \sum_{\dot{p}\sim\hat{F}_\phi(p|x_t,z_t)} \Big(y_t - h_\theta(\dot{p},x_t)\Big)^2. \quad (11)$$

Taking the RHS of Equation (11) as the objective and calculating the gradient leads to a version of Equation (10) in which a single draw can be used instead of two independent draws. This objective is easy to implement in practice as it just involves drawing samples during training. This is well-supported in deep network implementations because it is analogous to the data augmentation procedures that are commonly used to encourage invariance in deep networks. The analogy is more than just aesthetic—by optimizing this loss, we are essentially encouraging the network to be invariant to variations in the treatment that cannot be explained by our features and instrument. This encourages the network to ignore the effects of unobserved confounding variables.

However, we do not have theoretical guarantees that optimizing this upper bound on $\mathcal{L}(D;\theta)$ leads to good counterfactual performance. While it may converge more quickly because it exhibits lower variance, it will have worse asymptotic performance as it only approximates the desired loss function. We evaluated this tradeoff experimentally by comparing optimizing the upper bound to the more computationally expensive unbiased procedure. In our simulations, we found that upper bound loss tended to give better performance under practical computational limitations.

Discrete treatment and outcome spaces The discussion thus far has focused on continuous treatment and outcome spaces because they are more challenging mathematically. When the treatment space is discrete and low dimensional, F_ϕ is modeled as a categorical response so the gradient of Equation (7) can be expressed exactly as

$$\begin{aligned}\nabla_\theta\mathcal{L}_t = -2\Bigg(y_t - \sum_k \pi_k(x_t,z_t;\phi)h_\theta(p^k,x_t)\Bigg) \\ \cdot \sum_k \pi_k(x_t,z_t;\phi)h'_\theta(p^k,x_t). \quad (12)\end{aligned}$$

Thus when the outcome space is discrete, we also have less to worry about with respect to bias in gradient updates. For discrete outcomes, we use the softmax loss and can use single-sample MC gradient estimates without introducing bias or large amounts of variance because the gradient with respect to the softmax loss does not involve a product of random variables.

3.2. Causal validation

There is a widespread belief that "[s]tandard methods for hyperparameter selection, such as cross-validation, are not applicable when there are no samples of the counterfactual outcome" (Shalit et al., 2016). This would be a significant problem in our setting because out-of-sample (OOS) validation procedures are crucial for tuning deep network hyper-parameters and optimization rules. Fortunately, both steps in our Deep IV procedure can be validated by simply evaluating the respective losses on held out data. Consider a held-out dataset $D_{h\text{-}o}$. Our first stage, fitting F_ϕ, is a standard density estimation problem and can be tuned to minimize the OOS deviance criterion

$$\min_\phi \sum_{d_l\in D_{h\text{-}o}} -\log f_\phi(p_l|x_l,z_l), \quad (13)$$

where f_ϕ is either the probability mass function or density function associated with F_ϕ, as appropriate. Second stage validation proceeds *conditional* upon a fitted F_ϕ, and we seek to minimize the held-out loss criterion

$$\min_\theta \sum_{d_l\in D_{h\text{-}o}} \left(y_l - \int h_\theta(p,x_l)dF_\phi(p|x_l,z_l)\right)^2. \quad (14)$$

The integral here can either be exact or MC approximate via sampling from F_ϕ.

Each stage is evaluated in turn, with second stage validation using the best-possible network as selected in the first stage.

This procedure guards against the 'weak instruments' bias (Bound et al., 1995) that can occur when the instruments are only weakly correlated with the policy variable. To see why, consider the worst case where the instruments are independent of the policy variable, i.e., $F(p|z) = F(p)$. Without validation, a sufficiently powerful model will perfectly overfit by approximating the conditional distribution $F(p|z_i)$ with a point mass at p_i. As a result, the causal loss function will approximate the standard prediction loss, leading to spurious conclusions. By contrast, the expected first-stage loss minimizer on the validation set is the model that best approximates the unconditional distribution $F(p)$, and given that, the second stage minimizer best approximates $g(p) = \bar{y}$ which predicts no relationship when there is no evidence in favor of it. That said, it should be noted that these criteria provide *relative* performance guidance: improving on each criterion will improve performance on counterfactual prediction problems, but without giving any information about how far $\hat{h}_\theta(p, x)$ is from true $h(p, x)$.

Our causal validation procedure is sequential: stage two's validation depends on the model chosen in the first stage. This greedy procedure is computationally efficient and causally valid, but potentially suboptimal because the first-stage loss is not of independent interest. Peysakhovich & Eckles (2017) concurrently developed a casual validation framework that considers both stages jointly in domains with categorical instruments and no observable features. These assumptions are too restrictive for the problems we consider, but it would be interesting to investigate whether there exists a joint optimization approach that offers practical benefits under more permissive assumptions.

4. Experiments

We evaluated our approach on both simulated and real data. We used simulations to assess DeepIV's ability to recover an underlying counterfactual function both in a low-dimensional domain with informative features and in a high-dimensional domain with features consisting of pixels of a handwritten image. We compared our approach to 2SLS and to a standard feed-forward network, evaluated the effectiveness of hyper-parameter optimization, and contrasted our biased and unbiased loss functions with various numbers of samples underlying the SGD step. We also considered a real-world dataset where ground truth was not available, showing that we could replicate the findings of a previous study in a dramatically more automatic fashion.

4.1. Simulations

Our simulation models a richer version of the airline example described in Section 1. We assume that there are 7 customer types $s \in \{1, ..., 7\}$ that each exhibit different levels of price sensitivity. We model the holiday effect on sales by letting the customer's price sensitivity vary continuously throughout the year according to a complex non-linear function, $\psi_t = 2\left((t-5)^4/600 + \exp\left[-4(t-5)^2\right] + t/10 - 2\right)$. The time of year t is an observed variable, generated as $t \sim \text{unif}(0, 10)$. Prices are a function of ψ_t and of the fuel price z, with the motivation that they are chosen strategically by the airline in order to move with average price sensitivity. In our example, the high demand that results from conferences breaks the conditional independence between our treatment variable p and the latent effects e, thereby violating the "unconfoundedness" assumption. We model this abstractly by generating our latent errors e with a parameter ρ that allows us to smoothly vary the correlation between p and e. Sales y are then generated as

$$y = 100 + (10 + p)s\psi_t - 2p + e, \quad p = 25 + (z + 3)\psi_t + v$$
$$z,\ v \sim \text{N}(0, 1) \ \text{ and } \ e \sim \text{N}(\rho v, 1 - \rho^2).$$

Our target counterfactual function is $h(t, s, p) = (10 + p)s\psi_t - 2p$. To evaluate the model, we consider the counterfactual question, "What would sales have been if prices had been changed to p'?" Thus the price in our test set is set deterministically over a *fixed grid* of price values that spans the range of training set prices. The observed features $[t, s]$ are sampled as in the original data generating process and we compare estimated $\hat{h}$ against the ground truth h.

Low dimensional domain We evaluated structural mean square error (MSE) while varying both the number of training examples and ρ, the correlation between e and p. In addition to Deep IV, we considered a regular feed-forward network (FFNet) with the same architecture as our outcome network, a non-parametric IV polynomial kernel regression (NonPar, Darolles et al., 2011) using Hayfield et al. (2008)'s R implementation, and standard two-stage least squares (2SLS). Full details of model architectures and hyperparameter choices for all the models are given in the Appendix.

The results are summarized in Figure 2. The performance of NonPar, of 2SLS, and of our Deep IV model was mostly unaffected by changes in ρ, reflecting the fact that these models are designed to be resilient to unobserved confounders. For 1000 data points, NonPar's mean performance was better than 2SLS but failed to match DeepIV. Because of NonPar's excessive computational requirements we were not able to fit it to the larger datasets. 2SLS is constrained by its homogeneity and linearity assumptions, and so did not improve with increasing amounts of data. Adding regularized polynomial basis functions to 2SLS (2SLS(poly)) gives some empirical improvements in performance over 2SLS on larger datasets but the procedure is not causally valid because it violates 2SLS's linearity assumptions. Both forms of 2SLS performed far better than FFNet which did a

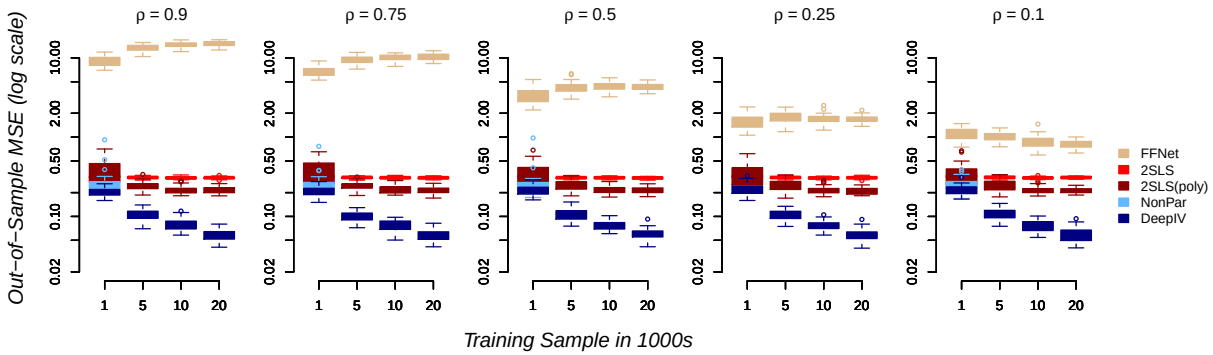

Figure 2. Out-of-sample predictive performance for different levels of endogeneity (ρ). Note that the test samples were generated with independent errors conditional upon a fixed grid of price values, breaking the endogeneity that existed in the training sample; this is why the feed-forward network did so poorly. Each model was fitted on 40 random samples from the DGP for each sample size and ρ-level.

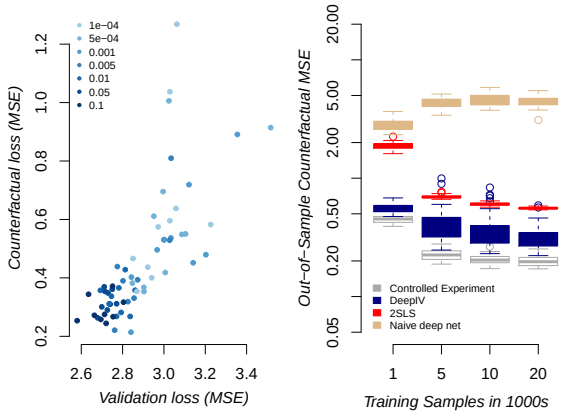

Figure 3. For the high-dimensional feature space problem we used a four-layer convolutional network to build an embedding of the image features which was concatenated with the observed time features and the instrument (first stage) and the treatment samples (second stage) and fed the resulting vector through another hidden layer before the output layer. *(Left)* Grid search over L2 and dropout parameters for the embedding used in the convolution network. *(Right)* Performance on an image experiment.

good job of estimating $\hat{h}(t,s,p)+\mathbb{E}[e|p]$ but a terrible job of recovering the true counterfactual. As ρ dropped, decreasing $\mathbb{E}[e|p]$, FFNet's performance improved but even with low levels of correlation between p and e it remained far worse than simple 2SLS. This occurred because we evaluated the models with respect to a fixed grid of treatment values which induced a covariate shift at test time. In contrast, Deep IV was the best performing model throughout and its performance improved as the amount of data grew.

High dimensional feature space In real applications, we do not typically get to observe variables like *customer type* that cleanly delineate our training examples into explicit classes, but may instead observe a large number of features that correlate with such types. To simulate this, we replaced the customer type label $s \in \{0,1,...,7\}$ with the pixels of the corresponding handwritten digit from the MNIST dataset (LeCun & Cortes, 2010). The task remained the same, but the model was no longer explicitly told that there were 7 customer types and instead had to infer the relationship between the image data and the outcome.

In this far more challenging domain, performance is sensitive to the choice of hyper-parameters, necessitating optimization on a validation set. Figure 3 shows an evaluation of the appropriateness of our loss function for hyper-parameter tuning, comparing our validation-set loss after grid search over Dropout and L2-regularization parameters to test set loss. We found a clear linear relationship between the losses; the best performing validation set model was among the best five performing models under the true causal loss.

We can get an upper bound on the performance of a particular model architecture by comparing its performance to the same architecture trained on data from a simulated randomized experiment on the same data-generating process. We simulated this by generating the outcome y with independent noise e and by generating p uniformly at random over its support. Thus the controlled model had to solve a standard supervised learning problem where errors were generated independently and there was no test-time covariate shift. As before, the naive deep network also shared the same architecture in addition to taking the instrument as input. This experiment showed that DeepIV was able to make up most of the loss in counterfactual prediction performance that the naive network suffered by not accounting for the causal prediction problem. However, there was still a gap in performance: with 20 000 data points the controlled experiment achieved an average mean squared error of 0.20 while DeepIV managed 0.32.

Loss Function	# Samples	Mean	Stdev
Upper bound	1	**0.32**	0.085
Unbiased	2	0.48	0.107
Unbiased	4	0.50	0.158
Unbiased	8	0.44	0.100
Unbiased	16	**0.39**	0.098

Table 1. Comparing loss functions on the high-dimensional image task. Bold indicates best performing models. Although the upper bound loss offered better mean performance, the two results are not statistically significantly different.

Performance of the unbiased loss relative to the upper bound loss We tested how our two approaches to optimization affected performance on the image task with 20 000 training examples. The results are summarized in Table 1. The upper bound loss loss gave significantly better performance than all but the unbiased loss version based on 16 samples, without requiring multiple passes through the network to evaluate the gradient. Thus, the bias introduced by the upper bound did not meaningfully degrade counterfactual performance in our experiments.

4.2. Application: Search-advertisement position effects

Our experiments so far have considered synthetic data. We now evaluate the utility of our approach on real data for which we do not have access to ground truth. This means that we cannot evaluate models in terms of their predictions; instead, we show that we can replicate the results of a previously published study in a dramatically more automated fashion. Specifically, we examine how advertiser's position on the Bing search page (their "slot") affects the probability of a user click, allowing for different treatment effects for different advertiser-query pairs. For example, we aim to detect differences in the importance of ad position when Coke bids on the word "Coke" (an "on-brand query") versus when Pepsi bids on "Coke" (an "off-brand query") versus when Coke bids on "www.coke.com" (an "on-nav query", occurring when a user types a url in the search box by mistake) versus when Pepsi bids on "www.coke.com" (an "off-nav query"). This question was studied extensively by Goldman & Rao (2014) using a nonparametric IV estimation approach that involved a detailed construction of optimal instruments, as well as a separate hand-coded classification of advertiser–query pairs into the four categories above.

Our goal is to replicate these results in an *automated* fashion. Advertiser position is correlated with latent user intent (e.g. when a user searches for "Coke", it is likely both that they will click and that "Coke" will be the top advertiser), so we need instruments to infer causation. The instruments proposed by Goldman & Rao (2014) are a series of indicators for experiments run by Bing in which advertiser–query pairs were randomly assigned to different algorithms that in turn scored advertisers differently in search auctions, resulting in random variation in position. Our estimation algorithm takes the experiment ID directly as an instrument.

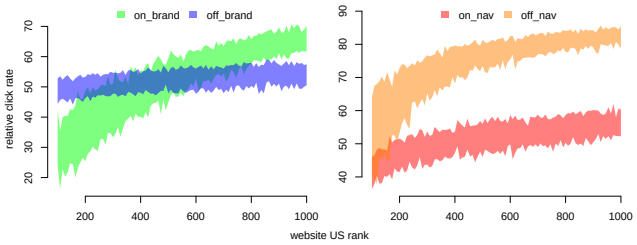

Figure 4. The inter-quartile range in advertiser-query specific estimates of the relative drop in click rate from moving from position 1 to position 2 (i.e. y-axis denotes $\frac{\text{cr}_1 - \text{cr}_2}{\text{cr}_1}$%), as the popularity of the advertiser varies along the x-axis (as measured by visit rank on Alexa.com), for on-brand versus off-brand queries (left panel) and on-nav versus off-nav queries (right panel) for a single combination of advertiser and search-query text.

As features, we gave the deep nets the query url and user query as text. The url was parsed into tokens on dashes and dots and these tokens were then parsed on punctuation and whitespace. Given the outcome variable (an indicator for user click), the features, and the instruments, we applied our methodology directly to the data without any of the additional feature engineering and construction of optimal instruments performed by Goldman & Rao (2014). This approach was tractable despite the fact that the dataset contained over 20 million observations. Figure 4 shows the results. We were able to replicate the original study's broad findings, namely that (i) that for "on-brand" queries, position was worth more to small websites; and (ii) that the value of position for on-nav queries was much smaller than for off-nav queries. A naive non-causal regression found an unrealistic average treatment effect (ATE; across sampled advertisers and queries) drop in click rate of 70% from 1st to 2nd position; in contrast, the causal estimate of the ATE was a more modest 12% drop.

5. Discussion

We have presented DeepIV, an approach that leverages instrument variables to train deep networks that directly minimize the counterfactual prediction error and validate the resulting models on held-out data. DeepIV significantly reduced counterfactual error measured in simulation experiments and was able to replicate previous IV experiments without extensive feature engineering. In future work, we plan to discuss interference techniques for the DeepIV framework and explore how this approach generalizes to other causal graphs given appropriate assumptions.

Acknowledgements

We would like to thank Susan Athey, Xiaohong Chen and Demian Pouzo for their helpful discussions and the anonymous reviewers for their useful comments on the paper. We would also like to thank Holger Hoos for the use of the Ada cluster, without which the experiments would not have been possible.

References

Angrist, J. D., Imbens, G.W., and Rubin, D. B. Identification of causal effects using instrumental variables. *Journal of the American Statistical Association*, 91(434):444–455, 1996.

Angrist, Joshua D and Pischke, Jörn-Steffen. *Mostly harmless econometrics: An empiricist's companion*. Princeton university press, 2008.

Athey, Susan and Imbens, Guido. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113:7353–7360, 2016.

Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Blundell, Richard, Chen, Xiaohong, and Kristensen, Dennis. Semi-nonparametric iv estimation of shape-invariant engel curves. *Econometrica*, 75:1630–1669, 2007.

Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

Bottou, Léon, Peters, Jonas, Quiñonero-Candela, Joaquin, Charles, Denis X, Chickering, D Max, Portugaly, Elon, Ray, Dipankar, Simard, Patrice, and Snelson, Ed. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research*, 14(1):3207–3260, 2013.

Bound, John, Jaeger, David A, and Baker, Regina M. Problems with instrumental variables estimation when the correlation between the instruments and the endogenous explanatory variable is weak. *Journal of the American Statistical Association*, 90:443–450, 1995.

Bousquet, Olivier and Bottou, Léon. The tradeoffs of large scale learning. In *Advances in neural information processing systems (NIPS)*, pp. 161–168, 2008.

Chen, X. and Pouzo, D. Estimation of nonparametric conditional moment models with possibly nonsmooth generalized residuals. *Econometrica*, 80:277–321, 2012.

Darolles, Serge, Fan, Yanqin, Florens, Jean-Pierre, and Renault, Eric. Nonparametric instrumental regression. *Econometrica*, 79:1541–1565, 2011.

Duchi, John, Hazan, Elad, and Singer, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12: 2121–2159, 2011.

Goldman, Mathew and Rao, Justin M. Experiments as instruments: heterogeneous position effects in sponsored search auctions. 2014.

Hall, Peter and Horowitz, Joel L. Nonparametric methods for inference in the presence of instrumental variables. *Ann. Statist.*, 33(6):2904–2929, 12 2005.

Hayfield, Tristen, Racine, Jeffrey S, et al. Nonparametric econometrics: The np package. *Journal of statistical software*, 27(5):1–32, 2008.

Johansson, F. D., Shalit, U., and Sontag, D. Learning representations for counterfactual inference. In *Proceedings of the 33nd International Conference on Machine Learning, (ICML)*, pp. 3020–3029, 2016.

Kingma, Diederik and Ba, Jimmy. ADAM: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2014.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 2015.

LeCun, Yann and Cortes, Corinna. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Newey, W. K. and Powell, J. L. Instrumental variable estimation of nonparametric models. *Econometrica*, 71(5): 1565–1578, 2003.

Pearl, J. *Causality*. Cambridge University Press, 2009.

Peysakhovich, A. and Eckles, D. Learning causal effects from many randomized experiments using regularized instrumental variables. *ArXiv e-prints*, January 2017.

Reiersøl., O. Confluence analysis by means of instrumental sets of variables. *Arkiv för Matematik, Astronomi och Fysik*, 32a(4):1–119, 1945.

Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

Rosenbaum, P. R. and Rubin, D. B. Assessing sensitivity to an unobserved binary covariate in an observational study with binary outcome. *Journal of the Royal Statistical Society. Series B (Methodological)*, 45(2):212–218, 1983.

Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks*, 2015.

Shalit, U., Johansson, F., and Sontag, D. Estimating individual treatment effect: generalization bounds and algorithms. *ArXiv e-prints*, June 2016.

Wager, Stefan and Athey, Susan. Inference of heterogeneous treatment effects using random forests. *arXiv:1510.04342*, 2015.

Wright, P. G. *The Tariff on Animal and Vegetable Oils*. Macmillan, 1928.

Zinkevich, Martin. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, 2003.