Journal of the American Statistical Association

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/uasa20

A Deep Generative Approach to Conditional Sampling

Xingyu Zhou, Yuling Jiao, Jin Liu & Jian Huang

To cite this article: Xingyu Zhou, Yuling Jiao, Jin Liu & Jian Huang (2023) A Deep Generative Approach to Conditional Sampling, Journal of the American Statistical Association, 118:543, 1837-1848, DOI: 10.1080/01621459.2021.2016424

To link to this article: https://doi.org/10.1080/01621459.2021.2016424

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2023, VOL. 118, NO. 543, 1837–1848: Theory and Methods
https://doi.org/10.1080/01621459.2021.2016424

A Deep Generative Approach to Conditional Sampling

Xingyu Zhou[a*], Yuling Jiao[b*], Jin Liu[c], and Jian Huang[c]

[a]Department of Statistics and Actuarial Science, University of Iowa, Iowa City, IA; [b]School of Mathematics and Statistics, Wuhan University, Wuhan, Hubei, China; [c]Centre for Quantitative Medicine, Duke-NUS Medical School, Singapore, Singapore

ABSTRACT

We propose a deep generative approach to sampling from a conditional distribution based on a unified formulation of conditional distribution and generalized nonparametric regression function using the noise-outsourcing lemma. The proposed approach aims at learning a conditional generator, so that a random sample from the target conditional distribution can be obtained by transforming a sample drawn from a reference distribution. The conditional generator is estimated nonparametrically with neural networks by matching appropriate joint distributions using the Kullback-Liebler divergence. An appealing aspect of our method is that it allows either of or both the predictor and the response to be high-dimensional and can handle both continuous and discrete type predictors and responses. We show that the proposed method is consistent in the sense that the conditional generator converges in distribution to the underlying conditional distribution under mild conditions. Our numerical experiments with simulated and benchmark image data validate the proposed method and demonstrate that it outperforms several existing conditional density estimation methods. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received March 2021
Accepted December 2021

KEYWORDS
Distribution matching;
Generative learning;
High-dimensional data;
Neural networks;
Nonparametric estimation

1. Introduction

A fundamental problem in statistics and machine learning is how to model the relationship between a response Y and a predictor X. Such a model can be used for predicting the values of Y based on the new observations of X and for assessing the variation in Y for a given value of X. Regression models that focus on estimating the conditional mean or median of the response given the predictor have been widely used for such purposes. However, in problems when the conditional distribution is multimodal or asymmetrical, conditional mean and median are no longer adequate for modeling the relationship between Y and X. In general, to completely understand how the response depends on the predictor, it becomes necessary to learn the conditional distribution, which provides a full description of the relationship between the response variable and the predictor. Conditional distribution also plays a central role in many important areas, including representation learning (Bengio, Courville, and Vincent 2013), sufficient dimension reduction (Li 1991; Cook 1998), graphical models (Bishop 2006), nonlinear independent component analysis (Hyvärinen and Pajunen 1999), among others.

In this article, we propose a nonparametric generative approach to sampling from a conditional distribution. For convenience, we shall refer to the proposed method as the generative conditional distribution sampler (GCDS). For a given value of the predictor $X = x$, GCDS aims at estimating a function $G(\eta, x)$ of η and x, where η is a random variable from a simple reference distribution such as normal or uniform, such that $G(\eta, x)$ follows the conditional distribution of Y given $X = x$. Such a function is called a conditional generator. To sample from the conditional distribution, we only need to calculate $G(\eta, x)$ after generating η from the reference distribution. Therefore, the conditional generator G contains all the information about the conditional distribution of Y given X. We estimate the conditional generator G nonparametrically using neural networks. An appealing feature of GCDS is that it is applicable to the settings when either of or both X and Y are high-dimensional such as in the problems of image data analysis.

There is an extensive literature on nonparametric conditional density estimation. The prevailing approaches are based on smoothing methods, including kernel smoothing and local polynomials (Rosenblatt 1969; Scott 1992; Chen et al. 2001; Hall and Yao 2005; Bott and Kohler 2017). Typically, the joint density of (X, Y) and the marginal density of X are first estimated using unconditional kernel density estimators. Then, the conditional density estimator is obtained as the ratio of the estimated joint density over the estimated marginal density. Another approach is to transform the problem of estimating a conditional density to a suitably formulated regression problem (Fan, Yao, and Tong 1996; Fan and Yim 2004) and use the method for nonparametric regression for conditional density estimation. Nearest neighbors method has also been used in estimating conditional density and conditional quantiles through kernel smoothing (Bhattacharya and Gangopadhyay 1990). Approaches based on expanding the conditional density function in terms of certain basis functions have also been developed (Sugiyama et al. 2010; Izbicki and Lee

CONTACT Jian Huang jian-huang@uiowa.edu Department of Statistics and Actuarial Science, University of Iowa, Iowa City, IA 52242.
*Xinyu Zhou and Yuling Jiao constributed equally to this work.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.

2016). The method proposed by Izbicki et al. (2017) approximates the conditional density using orthogonal basis functions and transform the problem of conditional density estimation into a regression problem. A common feature of these methods is that they seek to estimate the functional form of the conditional density.

However, these existing conditional density estimation methods do not work well for problems with high-dimensional data. In particular, they suffer from the so-called curse of dimensionality, that is, their performance deteriorates dramatically as the dimensionality of the dependent variable or the response variable becomes relatively large. Indeed, most conditional density estimators can only effectively handle up to a few predictors. In addition, most of these methods focus on the case when the response Y is a scalar variable and cannot handle the case of a high-dimensional response vector.

The proposed approach is inspired by the recently developed generative adversarial networks (GAN) (Goodfellow et al. 2014). Instead of estimating the functional form of the conditional density, GCDS is a generative learning approach that seeks to estimate a conditional sampler. The basis of GCDS is a unified formulation of the conditional density estimation and the generalized nonparametric regression based on the noise-outsourcing lemma in probability theory (Kallenberg 2002; Austin 2015). By this lemma, the problem of nonparametric conditional density estimation is equivalent to a generalized nonparametric regression problem. This equivalency implies that, for any given $X = x$, we can estimate the conditional generator $G(\eta, x)$ so that a random sample from the conditional distribution can be obtained based on this function using a random sample η from a reference distribution, such as the uniform or the standard normal distribution. The estimation of the conditional generator G is achieved through matching appropriate joint distributions using the Kullback-Liebler divergence and its variational form. We take advantage of the abilities of neural networks in approximating high-dimensional functions and estimate G nonparametrically using deep neural networks.

There are several advantages of GCDS over the classical methods for conditional density estimation. First, there is no restriction on the dimensionality of the response variable, while the classical methods typically only consider the case of a scalar response variable. Indeed, our methods allow either of or both the predictor and the response to be high-dimensional. Second, GCDS can handle both continuous- and discrete-type predictors and responses. Third, since our method learns a generative function for the underlying conditional distribution based on a simple reference distribution, it is easy to obtain estimates of the summary measures of the underlying conditional distribution, including the conditional moments and quantiles by Monte Carlo. In comparison, it is cumbersome to do so based on the traditional conditional density estimation methods, since it involves numerical integrations that are difficult to implement in high-dimensional settings. Fourth, we demonstrate that the proposed method works for complex and high-dimensional data problems such as image generation and reconstruction. The traditional conditional density estimation methods are not able to deal with such problems. Finally, we show that GCDS is consistent in the sense that the samples it generates converge weakly to the underlying target conditional distribution. To the best of our knowledge, such a result is the first of its kind in the context of deep generative learning.

In the remainder of this article, we first describe a generative representation of conditional distribution based on the noise-outsourcing lemma, and explain that sampling from a conditional distribution can be achieved by using a conditional generator. This provides the theoretical foundation for the distribution matching method proposed in Section 3. The distribution matching is carried out by using the variational form of the f-divergence, which includes the Kullback-Liebler divergence as a special case. In Section 4, we establish the consistency of the conditional generator in the sense that the joint distribution of X and generated sample converges to the joint distribution of (X, Y). In Section 5, we conduct extensive simulation studies to evaluate the finite sample performance of the proposed method and illustrate its application to an image generation and reconstruction problems using benchmark image data. Concluding remarks are given in Section 6. Additional numerical experiment results and technical details are provided in the supplementary material.

2. Generative Representation of Conditional Distribution

Consider a pair of random vectors $(X, Y) \in \mathcal{X} \times \mathcal{Y}$, where X is a vector of predictors and Y is a vector of response variables or labels. For regression problems, we have $\mathcal{Y} \subseteq \mathbb{R}^q$ with $q \geq 1$; for classification problems, $\mathcal{Y}$ is a set of finite many labels. We assume $\mathcal{X} \subseteq \mathbb{R}^d$ with $d \geq 1$. The predictor X can contain both continuous and categorical components. Suppose $(X, Y) \sim P_{X,Y}$ with marginal distributions $X \sim P_X$ and $Y \sim P_Y$. Denote the conditional distribution of Y given X by $P_{Y|X}$. For a given value x of X, we also write the conditional distribution as $P_{Y|X=x}$. Let η be a random vector independent of X with a known distribution P_η. For example, we can take P_η to be the standard multivariate normal $N(\mathbf{0}, \mathbf{I}_m)$ for a given $m \geq 1$. We note that m does not need to be the same as q, the dimension of Y.

Our goal is to find a function $G : \mathbb{R}^m \times \mathcal{X} \mapsto \mathcal{Y}$ such that the conditional distribution of $G(\eta, X)$ given $X = x$ is the same as the conditional distribution of Y given $X = x$. Since η is independent of X, it is equivalent to finding a G such that

$$G(\eta, x) \sim P_{Y|X=x},\ x \in \mathcal{X}, \tag{1}$$

Thus to sample from the conditional distribution $P_{Y|X=x}$, we can first sample an $\eta \sim P_\eta$, then calculate $G(\eta, x)$. The resulting value $G(\eta, x)$ is a sample from $P_{Y|X=x}$.

Does such a function G exist? The existence of G is guaranteed by the noise-outsourcing lemma from probability theory under minimal conditions (Kallenberg 2002, theor. 5.10 ; Austin 2015, lem. 3.1).

Lemma 2.1. (Noise-outsourcing lemma). Let (X, Y) be a random pair taking values in $\mathcal{X} \times \mathcal{Y}$ with joint distribution $P_{X,Y}$. Suppose $\mathcal{Y}$ is a standard Borel space. Then there exist a random vector $\eta \sim P_\eta = N(\mathbf{0}, \mathbf{I}_m)$ for any given $m \geq 1$ and a Borel-measurable function $G : \mathbb{R}^m \times \mathcal{X} \to \mathcal{Y}$ such that η is independent of X and

$$(X, Y) = (X, G(\eta, X)) \text{ almost surely.} \tag{2}$$

Because η and X are independent, any G satisfying (2) also satisfies (1), that is, $G(\cdot, x)$ is a conditional generator for $P_{Y|X=x}$. In the original noise-outsourcing lemma, the distribution P_η is a uniform distribution on $[0, 1]$. In Lemma 2.1, P_η is taken to be $N(\mathbf{0}, \mathbf{I}_m)$ with $m \geq 1$. This is more convenient in applying GCDS when it is better to use a random vector in stead of a random scalar as the noise source. In the supplementary material, we show that we can indeed take P_η to be $N(\mathbf{0}, \mathbf{I}_m)$ based on the original noise-outsourcing lemma. The value of m should be chosen on a case-by-case basis in practice.

Lemma 2.1 provides a unified view of conditional distribution estimation and (generalized) nonparametric regression. To see this, it is informative to reverse the order of (1) and write it as

$$Y|X = x \sim G(\eta, x),\ x \in \mathcal{X}. \tag{3}$$

This expression shows that the problem of finding G is similar to that of estimating a generalized regression function nonparametrically by matching the conditional distributions. Therefore, G can also be considered a generalized regression function. The standard formulation of nonparametric regression with an additive error is a special case of (3). Indeed, if we assume $G(\eta, x) = G_0(x) + \eta$ with $\mathbb{E}(\eta|X) = 0$, then (3) leads to the standard nonparametric regression model $\mathbb{E}(Y|X = x) = G_0(x)$.

Sampling from a conditional distribution generally cannot be done by simply using the existing methods for sampling from an unconditional distribution. This can be explained as follows. For any given x, the problem is to find a function $G_x(\eta)$ such that $G_x(\eta) \sim P_{Y|X=x}$, where we use x as the subscript of G to indicate that the form of G depends on x. If X is a discrete random variable and only takes the values in a finite set, then we can simply find the function G_x for each x using the existing generative methods such as GAN (Goodfellow et al. 2014). However, this is not feasible if X is a continuous-type random variable. In general, the methods for generating samples from an unconditional distribution cannot be directly applied to find a function of η for generating samples from the conditional distribution $P_{Y|X}$.

To get around this difficulty, we note that matching the conditional distribution of $G(\eta, x)$ with $P_{Y|x}$ for a given $x \in \mathcal{X}$ is equivalent to matching the joint distribution of $(X, G(\eta, X))$ and the joint distribution of (X, Y), if the same marginal distribution of X is involved. This can be easily seen as follows. Let $T = G(\eta, X)$. Then $P_{T|X} = P_{Y|X}$ if and only if $P_{T|X}P_X = P_{Y|X}P_X$ on the support of (X, Y), that is, $P_{X,T} = P_{X,Y}$. We summarize this simple but key observation in the following lemma.

Lemma 2.2. Suppose that η is independent of X. Then $G(\eta, x) \sim P_{Y|X=x}$, $x \in \mathcal{X}$ if and only if

$$(X, G(\eta, X)) \sim (X, Y). \tag{4}$$

Because of (4), we refer to G as a conditional generator, since given $X = x$, $G(\eta, x) \sim P_{Y|X=x}$. Lemma 2.2 shows that finding a G such that (1) holds amounts to finding a G such that the joint distribution of $(X, G(\eta, X))$ is the same as that of (X, Y).

It is clear that the conditional generator satisfying (4) contains all the information about the conditional distribution of Y given X. For example, consider the conditional expectation $g(x) = \mathbb{E}(Y|X = x)$ and the conditional variance $v(x) = \mathrm{var}(Y|X = x)$. By (4), we have $g(x) = \mathbb{E}_{\eta\sim P_\eta} G(\eta, x)$ and $v(x) = \mathrm{var}_\eta[G(x, \eta)]$. Therefore, we can calculate g and v based on G. Although it is difficult to calculate these functions exactly, it is easy to approximate them via Monte Carlo. Specifically, let $\eta_1, \ldots, \eta_J$ be a random sample generated from P_η, then we can approximate $g(x)$ and $v(x)$ by

$$\tilde{g}(x) = \frac{1}{J}\sum_{j=1}^{J} G(\eta_j, x) \text{ and } \tilde{v}(x) = \frac{1}{J}\sum_{j=1}^{J}[G(\eta_j, x) - \tilde{g}(x)]^2. \tag{5}$$

Since it is easy and inexpensive to generate random samples from P_η, for any given x we can easily accurately approximate the summary measures such as moments and quantiles of the conditional distribution $P_{Y|X=x}$ based on $\{G(\eta_j, x), j = 1, \ldots, J\}$ for a sufficiently large J.

3. Distribution Matching Estimation

3.1. Adversarial Generative Networks

The generative adversarial networks (GAN) (Goodfellow et al. 2014) is an approach to learning a high-dimensional (unconditional) distribution. It is formulated as a minimax adversarial game between two players, a generator G and a discriminator D. The discriminator D is parameterized using a neural network that serves as a witness to distinguish between a sample Y from the data distribution and a sample from the generative model. The generator $G(\eta)$ maps samples η from the reference distribution P_η to the data distribution. The generator G is trained to maximally confuse the discriminator into believing that samples it generates come from the data distribution. Formally, GAN solves the minimax optimization problem

$$\min_G \max_D \mathbb{E}_{Y\sim P_{\text{data}}} \log D(Y) + \mathbb{E}_{\eta\sim P_\eta} \log[1 - D(G(\eta))]. \tag{6}$$

Conditional generative adversarial networks (cGAN) (Mirza and Osindero 2014) estimate the distribution of the images conditioning on some auxiliary information, especially class labels. Similar to GAN, it solves a two-player minimax game using an objective function with the same form as (6). See equation (2) in Mirza and Osindero (2014). cGAN performs the conditioning by feeding the class label information into the neural networks for the discriminator and the generator as additional input layer. However, cGAN does not work well for data generation with continuous conditions.

3.2. f-Divergence and Its Variational Form

While the minimax formulation has an attractive intuitive interpretation as a two-player game between the generator and the discriminator, it is helpful to understand it as the dual form of the primal problem of minimizing the Jensen-Shannon divergence between the data distribution and the distribution of the generator (Goodfellow et al. 2014). By considering a general discrepancy measure between two distributions such as the f-divergence, one can formulate a class of generative learning

methods including GAN as a special case (Nowozin, Cseke, and Tomioka 2016).

By Lemma 2.2, we can estimate G by matching the distribution of $(X, G(\eta, X))$ with the distribution of (X, Y). For this purpose, we first describe the f-divergence and its variational form. Let P and Q be two probability distributions on $\mathbb{R}^d$. Let p and q be the density functions of P and Q with respect to a common dominant measure, respectively. Suppose Q is absolutely continuous with respect to P. The f-divergence (Ali and Silvey 1966) of Q with respect to P is defined by

$$\mathbb{D}_f(q\|p) = \int f\left(\frac{q(z)}{p(z)}\right) p(z)dz, \tag{7}$$

where $f : \mathbb{R}^+ \to \mathbb{R}$ is a convex function with $f(1) = 0$ and is strictly convex at $x = 1$. A basic property of the f-divergence following from Jensen's inequality is that $\mathbb{D}_f(q\|p) \geq 0$ for every q, p and $\mathbb{D}_f(q\|p) = 0$ if and only if $q = p$.

The Kullback–Leibler (KL) divergence is an important special case with $f(x) = x \log x$, which has a simple expression

$$\begin{aligned}\mathbb{D}_{\mathrm{KL}}(q\|p) &= \int \frac{q(z)}{p(z)} \log\left(\frac{q(z)}{p(z)}\right) p(z)dz \\ &= \int \log\left(\frac{q(z)}{p(z)}\right) q(z)dz. \end{aligned}\tag{8}$$

Let $r = q/p$ be the density ratio of the densities q and p. It is convenient to express the KL divergence as

$$\mathbb{D}_{\mathrm{KL}}(q\|p) = \int \log\left(\frac{q(z)}{p(z)}\right) q(z)dz = \mathbb{E}_{Z\sim q}[\log r(Z)].$$

A useful representation of the f-divergence is its variational form. We will use it to construct an objective function for training the conditional generator G. The variational form is based on the Fenchel conjugate of f (Rockafellar 1970), defined as $f^*(t) = \sup_{x\in\mathbb{R}}\{tx - f(x)\}, t \in \mathbb{R}$. Then the f-divergence has the following variational formulation (Keziou 2003; Nguyen, Wainwright, and Jordan 2010).

Lemma 3.1. Let $\mathcal{D}$ be a class of measurable functions $D : \mathbb{R}^d \to \mathbb{R}$. Suppose f is a differentiable convex function. Then

$$\mathbb{D}_f(q\|p) \geq \sup_{D\in\mathcal{D}} [\mathbb{E}_{Z\sim q} D(Z) - \mathbb{E}_{W\sim p} f^*(D(W))], \tag{9}$$

where the equality holds if and only if $f'(q/p) \in \mathcal{D}$ and the supremum is attained at $D^* = f'(q/p)$.

Commonly used divergence measures, including the KL divergence, the Jensen-Shannon (JS) divergence and the χ^2-divergence, can be considered special cases of f-divergence. We give a proof of Lemma 3.1 and the expressions of the conjugate functions and variational forms of these divergence measures in the online supplementary material.

3.3. Distribution Matching Estimation via f-Divergence

We now apply Lemmas 2.2 and 3.1 to construct the objective function for estimating the conditional generator G. Let $p_{X,G(\eta,X)}$ and $p_{X,Y}$ be the densities of $(X, G(\eta, X))$ and (X, Y), respectively. At the population level, we seek to find a conditional generator G^* that minimizes the f-divergence $\mathbb{D}_f(p_{X,G(\eta,X)}\|p_{X,Y})$.

Lemma 3.2. A function $G^* : \mathbb{R}^m \times \mathcal{X} \to \mathcal{Y}$ is a minimizer of the f-divergence $\mathbb{D}_f(p_{X,G(\eta,X)}\|p_{X,Y})$,

$$G^* \in \mathrm{argmin}_G \mathbb{D}_f(p_{X,G(\eta,X)}\|p_{X,Y}) \tag{10}$$

if and only if $p_{X,G^*(\eta,X)} = p_{X,Y}$, that is, $(X, G^*(\eta, X)) \sim (X, Y)$.

This lemma is a direct consequence of the properties of the f-divergence. Let

$$r(z) = \frac{p_{X,G(\eta,X)}(z)}{p_{X,Y}(z)},\ z \in \mathbb{R}^d \times \mathbb{R}^q. \tag{11}$$

be the density ratio of $p_{X,G(\eta,X)}$ over $p_{X,Y}$. We only focus on the KL divergence below. By (8), we have

$$\mathbb{D}_{\mathrm{KL}}(p_{X,G(\eta,X)}\|p_{X,Y}) = \mathbb{E}_{(X,\eta)\sim p_X p_\eta}[\log r(X, G(\eta, X))].$$

Our goal is to minimize an empirical version of $\mathbb{D}_{KL}(p_{X,G(\eta,X)}\|p_{X,Y})$ with respect to G. The minimizer will serve as an estimator of G. The KL divergence depends on the unknown density functions $p_{X,G(\eta,X)}$ and $p_{X,Y}$ only through the density ratio r or the log-density ratio. Denote the log-density ratio by $D = \log r$. To estimate G, we will also need to estimate D. We note that estimating density ratio is usually easier than estimating individual densities separately. The log-density ratio D can be intuitively interpreted as a discriminator that quantifies the difference between the distributions of $(X, G(\eta, X))$ and (X, Y).

Therefore, in our problem, the loss function determined by the log-density ratio D needs to be estimated along with the parameter of interest G. For this purpose, we consider the variational form of the KL divergence. The dual of $f(x) = x \log x$ is $f^*(t) = \exp(t - 1)$ (Nguyen, Wainwright, and Jordan 2010). By Lemma 3.1, we can write the variational representation of the KL-divergence as

$$\begin{aligned}&\mathbb{D}_{\mathrm{KL}}(p_{X,G(\eta,X)}\|p_{X,Y}) \\ &= \sup_D\{\mathbb{E}_{(X,\eta)\sim P_X P_\eta}[D(X, G(\eta, X))] \\ &\quad -\mathbb{E}_{(X,Y)\sim P_{X,Y}}[\exp(D(X, Y) - 1)]\} \\ &= \sup_D\{\mathbb{E}_{(X,\eta)\sim P_X P_\eta}[D(X, G(\eta, X))] \\ &\quad -\mathbb{E}_{(X,Y)\sim P_{X,Y}}[\exp(D(X, Y))]\} + 1, \end{aligned}\tag{12}$$

where the second equality follows by change of variables from $D - 1$ to D in the supremum operation. For the purpose of estimating G by minimizing the KL divergence, we can ignore the constant 1 in (12). So we consider the criterion

$$\begin{aligned}\mathcal{L}(G, D) = &\mathbb{E}_{(X,\eta)\sim P_X P_\eta}[D(X, G(\eta, X))] \\ &-\mathbb{E}_{(X,Y)\sim P_{X,Y}}[\exp(D(X, Y))]. \end{aligned}\tag{13}$$

The variational form is convenient since it is easy to obtain its empirical version when random samples are available. Then at the population level, the target conditional generator G^* and the target discriminator D^* are characterized by the minimax problem

$$(G^*, D^*) = \mathrm{argmin}_G \mathrm{argmin}_D \mathcal{L}(G, D). \tag{14}$$

Suppose that $\{(X_i, Y_i), i = 1, \ldots, n\}$ are iid $P_{X,Y}$ and $\{\eta_i, i = 1, \ldots, n\}$ are independently generated from P_η. We consider the following empirical version of $\mathcal{L}(G, D)$:

$$\widehat{\mathcal{L}}(G, D) = \frac{1}{n}\sum_{i=1}^{n} D(X_i, G(\eta_i, X_i)) - \frac{1}{n}\sum_{i=1}^{n} \exp(D(X_i, Y_i)). \tag{15}$$

We estimate G nonparametrically using feedforward neural networks (FNN) (Goodfellow, Bengio, and Courville 2016) based on the objective function $\widehat{\mathcal{L}}(G, D)$ in (15). We use two FNNs: the conditional generator network $G_{\boldsymbol{\theta}}$ with parameter $\boldsymbol{\theta}$ for estimating G and the second network $D_{\boldsymbol{\phi}}$ with parameter $\boldsymbol{\phi}$ for estimating the discriminator D. For any function $f(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}^d$, denote $\|f\|_\infty = \sup_{\boldsymbol{x}\in\mathcal{X}} \|f(\boldsymbol{x})\|$, where $\|\cdot\|$ is the Euclidean norm.

- The generator network $G_{\boldsymbol{\theta}}$: let $\mathcal{G} \equiv \mathcal{G}_{\mathcal{H},\mathcal{W},\mathcal{S},\mathcal{B}}$ be the set of ReLU neural networks $G_{\boldsymbol{\theta}} : \mathbb{R}^m \times \mathbb{R}^d \to \mathbb{R}^q$ with parameter $\boldsymbol{\theta}$, depth $\mathcal{H}$, width $\mathcal{W}$, size $\mathcal{S}$, and $\|G_{\boldsymbol{\theta}}\|_\infty \le \mathcal{B}$. Here the depth $\mathcal{H}$ refers to the number of hidden layers, so the network has $\mathcal{H} + 1$ layers in total. A $(\mathcal{H} + 1)$-vector $(w_0, w_1, \ldots, w_{\mathcal{H}})$ specifies the width of each layer, where $w_0 = d$ is the dimension of the input data and $w_{\mathcal{H}} = q$ is the dimension of the output. The width $\mathcal{W} = \max\{w_1, \ldots, w_{\mathcal{H}}\}$ is the maximum width of the hidden layers. The size $\mathcal{S} = \sum_{i=0}^{\mathcal{H}}[w_i \times (w_i + 1)]$ is the total number of parameters in the network. For multilayer perceptrons with equal-width hidden layers except the output layer, we have $\mathcal{S} = \mathcal{W}(m + 1) + (\mathcal{W}^2 + \mathcal{W})(\mathcal{H} - 1) + \mathcal{W} + q$.
- The discriminator network $D_{\boldsymbol{\phi}}$: Similarly, denote $\mathcal{D} \equiv \mathcal{D}_{\tilde{\mathcal{H}},\tilde{\mathcal{W}},\tilde{\mathcal{S}},\tilde{\mathcal{B}}}$ as the set of ReLU neural networks $D_{\boldsymbol{\phi}} : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}$, with parameter $\boldsymbol{\phi}$, depth $\tilde{\mathcal{H}}$, width $\tilde{\mathcal{W}}$, size $\tilde{\mathcal{S}}$, and $\|D_{\boldsymbol{\phi}}\|_\infty \le \tilde{\mathcal{B}}$.

Then $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are estimated by solving the empirical version of the minimax problem (14), that is,

$$(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}) = \operatorname{argmin}_{\boldsymbol{\theta}} \operatorname{argmax}_{\boldsymbol{\phi}} \widehat{\mathcal{L}}(G_{\boldsymbol{\theta}}, D_{\boldsymbol{\phi}}). \tag{16}$$

The estimated conditional generator is $\hat{G} = G_{\hat{\boldsymbol{\theta}}}$ and the estimated discriminator is $\hat{D} = D_{\hat{\boldsymbol{\phi}}}$. It is natural to compute $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}})$ by alternately minimizing $L(\boldsymbol{\theta}, \boldsymbol{\phi})$ with respect to $\boldsymbol{\theta}$ fixing $\boldsymbol{\phi}$ and maximizing $L(\boldsymbol{\theta}, \boldsymbol{\phi})$ with respect to $\boldsymbol{\phi}$ fixing $\boldsymbol{\theta}$. We provide the implementation details in Section 5.

4. Weak Convergence of Conditional Sampler

In this section, we provide sufficient conditions under which $(X, \hat{G}_n(\eta, X))$ converges in distribution to (X, Y). This implies that for given $X = x$ with $p_X(x) > 0$, the conditional distribution of $\hat{G}_n(\eta, x)$ given $X = x$ converges to the conditional distribution of Y given $X = x$. We focus on the case when X and Y are continuous-type random vectors. We establish a slightly stronger result by showing that the total variation norm

$$\|p_{X,\hat{G}(\eta,X)} - p_{X,Y}\|_{L_1} = \int_{\mathcal{X}\times\mathcal{Y}} |p_{X,\hat{G}(\eta,X)}(x, y) - p_{X,Y}(x, y)|dxdy$$

converges to zero.

Let $\mathcal{L}(G, D)$ be defined in (13). For any measurable function $G : \mathbb{R}^m \times \mathbb{R}^d \mapsto \mathbb{R}^q$, define

$$\mathbb{L}(G) = \sup_D \mathcal{L}(G, D). \tag{17}$$

For a fixed G, let p_{XG} be the joint density of $(X, G(\eta, X))$. Lemma 3.1 implies that the optimal D is $D^*(z) = \log(p_{XG}(z)/p_{XY}(z)) = \log r(z)$. Thus the optimal discriminator is the log-likelihood ratio serving as a critic of the resemblance between p_{XY} and p_{XG}. Substituting this expression into (17) yields $\mathbb{L}(G) = \mathbb{E}_{(X,\eta)\sim P_X P_\eta}[\log r(X, G(\eta, X))]$. Let $G^* \in \operatorname{argmin}_G \mathbb{L}(G)$. We have $P_{(X,G^*(X,\eta))} = P_{X,Y}$ by Lemmas 3.1 and 3.2.

We assume the following conditions.

(A1) The target conditional generator $G^* : \mathbb{R}^m \times \mathcal{X} \to \mathcal{Y}$ is continuous with $\|G^*\|_\infty \le C_0$ for some constant $0 < C_0 < \infty$.

(A2) For any $G \in \mathcal{G} \equiv \mathcal{G}_{\mathcal{D},\mathcal{W},\mathcal{S},\mathcal{B}}$, $r_G(z) = p_{X,G(\eta,X)}(z)/p_{X,Y}(z) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is continuous and $0 < C_1 \le r_G(z) \le C_2$ for some constants $0 < C_1 \le C_2 < \infty$.

We also make the following assumptions on the network parameters of the conditional generator $G_{\boldsymbol{\theta}}$ and the discriminator $D_{\boldsymbol{\phi}}$.

(N1) The network parameters of $\mathcal{G}$ satisfies

$$\mathcal{H}\mathcal{W} \to \infty \quad \text{and} \quad \frac{\mathcal{B}\mathcal{S}\mathcal{H}\log(\mathcal{S})\log n}{n} \to 0, \text{ as } n \to \infty.$$

(N2) The network parameters of $\mathcal{D}$ satisfies

$$\tilde{\mathcal{H}}\tilde{\mathcal{W}} \to \infty \quad \text{and} \quad \frac{\tilde{\mathcal{B}}\tilde{\mathcal{S}}\tilde{\mathcal{H}}\log(\tilde{\mathcal{S}})\log n}{n} \to 0, \text{ as } n \to \infty.$$

Theorem 4.1. Suppose that Assumptions (A1) and (A2) hold. If the network parameters of $\mathcal{G}$ and $\mathcal{D}$ satisfies the specifications (N1) and (N2), then

$$\mathbb{E}_{(X_i,Y_i,\eta_i,)_{i=1}^n} \|p_{X,\hat{G}_{\boldsymbol{\theta}}(\eta,X)} - p_{X,Y}\|_{L_1}^2 \to 0, \text{ as } n \to \infty. \tag{18}$$

A direct corollary of Theorem 4.1 is the following convergence result in terms of the conditional density functions.

Corollary 4.1. Suppose that the assumptions (A1) and (A2) hold and the network parameters of $\mathcal{G}$ and $\mathcal{D}$ satisfies the specifications (N1) and (N2). Then

$$\mathbb{E}_{X\sim P_X}\left[\int_{\mathcal{Y}} \left|p_{\hat{G}_{\boldsymbol{\theta}}(\eta,X)}(y|X) - p_{Y|X}(y|X)\right| dy\right] \to_P 0, \text{ as } n \to \infty.$$

Theorem 4.1 and Corollary 4.1 provide strong theoretical support for the proposed method under mild conditions. They are proved using the empirical process method (Bartlett and Mendelson 2002; Bartlett et al. 2019) and the recent results on approximating smooth functions by deep neural networks (Shen, Yang, and Zhang 2020). The main challenge of the proof is that the objective function is a minimax process indexed by two classes of neural networks. Details are given in the supplementary material.

Conditions (A1) and (A2) are mild regularity conditions that are often assumed in nonparametric estimation problems. Conditions (N1) and (N2) concern the depths, widths and sizes of the generator and the discriminator networks. These conditions require that the size of the network increases with the sample size, the product of the depth and the width increases with the sample size. We note that the conditions are flexible with respect to the network architecture. In particular, they allow either the depth or the width remain fixed. For example, we can have a deep network with fixed width or a wide network with fixed depth. A restriction of the conditions is that they require the network size to be smaller than the sample size. This restriction stems from the use of empirical process theory (Van der Vaart and Wellner 1996; Bartlett and Mendelson 2002; Bartlett et al. 2019) to control the stochastic error of the estimated generator and discriminator.

In nonparametric regression, there has been much recent work on convergence analysis of nonparametric estimators using deep neural networks. Two types of assumptions on the underlying model have been used in the analysis. The first type of assumptions postulates that the regression function has a compositional structure so that the intrinsic dimension of the function is lower than the ambient dimension (Bauer and Kohler 2019; Kohler and Langer 2020; Schmidt-Hieber 2020; Shen et al. 2021a). The second type assumes that the distribution of X is supported on a lower-dimensional manifold (Chen et al. 2019; Nakada and Imaizumi 2019; Jiao et al. 2021; Shen et al. 2021b). These works also require the size of the neural network used in the nonparametric regression to be smaller than the sample size to ensure the consistency of the estimators.

In the current problem of sampling from conditional distributions, convergence analysis is substantially more difficult than that in nonparametric regression. The main reason is that in the current setting, optimization is a minimax problem that leads to an *estimated nonparametric loss function*, that is, there is a second neural network involved for estimating the discriminator in the dual form of KL divergence, in addition to the neural network for estimating the conditional generator. In comparison, nonparametric regression with a given loss function such as least-square loss only has a single neural network for estimating the regression function.

We now give a high-level description of the proof for Theorem 4.1, the details are provided in the supplementary material. Lemma 3.2 implies that $\mathbb{L}(G^*) = 0$. For notational simplicity, write $\hat{G} = \hat{G}_{\boldsymbol{\theta}}$. By Pinsker's inequality (Tsybakov 2008), we have

$$\|p_{X,\hat{G}(\eta,X)} - p_{X,Y})\|_{L^1}^2 \le 2(\mathbb{L}(\hat{G}) - \mathbb{L}(G^*)). \tag{19}$$

So it suffices to show that the right side in (19) converges to zero in expectation. By the definition of $\mathbb{L}(G)$, we can write the excess risk as

$$\mathbb{L}(\hat{G}) - \mathbb{L}(G^*) = \sup_D \mathcal{L}(\hat{G}, D) - \sup_D \mathcal{L}(G^*, D).$$

A key step in the proof is the following decomposition of the excess risk. For any $\bar{G} \in \mathcal{G}$, we decompose the right-hand side in (19) as follows:

$$\begin{aligned}
\mathbb{L}(\hat{G}) - \mathbb{L}(G^*) = {} & \sup_D \mathcal{L}(\hat{G}, D) - \sup_{D\in\mathcal{D}} \mathcal{L}(\hat{G}, D) && (20)\\
& + \sup_{D\in\mathcal{D}} \mathcal{L}(\hat{G}, D) - \sup_{D\in\mathcal{D}} \widehat{\mathcal{L}}(\hat{G}, D) && (21)\\
& + \sup_{D\in\mathcal{D}} \widehat{\mathcal{L}}(\hat{G}, D) - \sup_{D\in\mathcal{D}} \widehat{\mathcal{L}}(\bar{G}, D) && (22)\\
& + \sup_{D\in\mathcal{D}} \mathcal{L}(\bar{G}, D) - \sup_{D} \mathcal{L}(\bar{G}, D) && (23)\\
& + \sup_D \mathcal{L}(\bar{G}, D) - \sup_D \mathcal{L}(G^*, D).
\end{aligned}$$

Since the terms in (21) and (23) are nonpositive, and the terms in (20) and (22) are smaller than $\sup_{D\in\mathcal{D},G\in\mathcal{G}} |\mathcal{L}(G, D) - \widehat{\mathcal{L}}(G, D)|$, we have

$$\begin{aligned}
\mathbb{L}(\hat{G}) - \mathbb{L}(G^*) \le {} & \sup_D \mathcal{L}(\hat{G}, D) - \sup_{D\in\mathcal{D}} \mathcal{L}(\hat{G}, D)\\
& + 2 \sup_{D\in\mathcal{D},G\in\mathcal{G}} |\mathcal{L}(G, D) - \widehat{\mathcal{L}}(G, D)|\\
& + \sup_D \mathcal{L}(\bar{G}, D) - \sup_D \mathcal{L}(G^*, D).
\end{aligned}$$

Note that $\bar{G}$ is arbitrary. By taking infimum with respect to $\bar{G}$ over $\mathcal{G}$ on both sides of the above display, we obtain

$$\mathbb{L}(\hat{G}) - \mathbb{L}(G^*) \le \Delta_1 + \Delta_2 + \Delta_3, \tag{24}$$

where

$$\begin{aligned}
\Delta_1 &= \sup_D \mathcal{L}(\hat{G}, D) - \sup_{D\in\mathcal{D}} \mathcal{L}(\hat{G}, D),\\
\Delta_2 &= 2 \sup_{G\in\mathcal{G},D\in\mathcal{D}} |\mathcal{L}(G, D) - \hat{\mathcal{L}}(G, D)|,\\
\Delta_3 &= \inf_{\bar{G}\in\mathcal{G}} [\mathbb{L}(\bar{G}) - \mathbb{L}(G^*)].
\end{aligned}$$

The first and the third terms Δ_1 and Δ_3 are the approximation errors and the second term Δ_2 is the statistical error. In the supplementary material, we derive (24) and show that these error terms converge to zero.

5. Implementation

We describe the implementation of GCDS. For training the generator $G_{\boldsymbol{\theta}}$ and the discriminator $D_{\boldsymbol{\phi}}$, we use the rectified linear unit (ReLU) as the activation function in $G_{\boldsymbol{\theta}}$ and $D_{\boldsymbol{\phi}}$. We train the discriminator and the generator iteratively by updating $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ alternately as follows:

(a) Fix $\boldsymbol{\theta}$, update the discriminator by ascending the stochastic gradient of the loss (15) with respect to $\boldsymbol{\phi}$.
(b) Fix $\boldsymbol{\phi}$, update the generator by descending the stochastic gradient of the loss (15) with respect to $\boldsymbol{\theta}$.

The training process is described below.

Algorithm: Training GCDS
Input: (a) Pairs $\{(X_i, Y_i),\ i = 1, \ldots, n\}$; (b) Samples $\{\eta\}_{i=1}^n$ from P_η
Output: Conditional generator $G_{\hat{\boldsymbol{\theta}}}$ and discriminator $D_{\hat{\boldsymbol{\theta}}}$
While not converged do

- Compute $\tilde{Y}_i = G_{\boldsymbol{\theta}}(\eta_i, X_i)$, $i = 1, 2, \ldots, n$. Let $S_1 = \{(X_i, Z_i, V_i) = (X_i, Y_i, 1),\ i = 1, \ldots, n\}$ and $S_2 = \{(X_i, Z_i, V_i) = (X_{i-n}, \tilde{Y}_i, -1),\ i = n+1, \ldots, 2n\}$.
- Randomly select $B/2$ samples from S_1 and another $B/2$ samples from S_2. Denote the subscripts of the selected samples by $\{b_i : i = 1, \ldots, B\}$.
- Update $D_{\boldsymbol{\phi}}$ by ascending its stochastic gradient:

$$\nabla_{\boldsymbol{\phi}} \left\{ \frac{1}{B} \sum_{i=1}^{B} [D_{\boldsymbol{\phi}}(X_{b_i}, Z_{b_i}) \mathbb{1}_{\{V_{b_i}=-1\}} - \exp(D_{\boldsymbol{\phi}}(X_{b_i}, Z_{b_i})) \mathbb{1}_{\{V_{b_i}=1\}}] \right\}.$$

- Randomly select B samples from $\{(X_i, Y_i),\ i = 1, \ldots, n\}$. Denote the subscripts of the selected samples by $\{b_i : i = 1, \ldots, B\}$
- Update $G_{\boldsymbol{\theta}}$ by descending its stochastic gradient:

$$\nabla_{\boldsymbol{\theta}} \left\{ \frac{1}{B} \sum_{i=1}^{B} D_{\boldsymbol{\phi}}(X_{b_i}, G_{\boldsymbol{\theta}}(\eta_{b_i}, X_{b_i})) \right\}.$$

End while

We implement the GCDS algorithm in TensorFlow (Abadi et al. 2016).

6. Numerical Experiments

In this section, we carry out numerical experiments to assess the performance of GCDS. We use both simulated and real datasets in the experiments. In addition to the results reported in this section, additional numerical results are given in the online supplementary material, including results from experiments evaluating how the performance of GCDS depends on the network architecture, the dimension m of the noise vector η and the sample size n.

6.1. Simulation Studies

We conduct simulation studies to evaluate the finite sample performance of GCDS. We also compare it with several existing conditional density estimation methods, including the nearest neighbor kernel conditional density estimation (NNKCDE; Dalmasso et al. 2020), the conditional kernel density estimation (CKDE, implemented in the `R` package `np`, Hall, Racine, and Li 2004), and the basis expansion method FlexCode (Izbicki et al. 2017). We implement GCDS in TensorFlow (Abadi et al. 2016) and use the stochastic gradient descent algorithm Adam (Kingma and Ba 2015) in training the neural networks. We use the conditional distributions based on the following models in the simulation studies.

1. (M1). A nonlinear model with an additive error term: $Y = X_1^2 + \exp(X_2 + X_3/3) + \sin(X_4 + X_5) + \varepsilon$, $\varepsilon \sim N(0, 1)$.
2. (M2). A model with an additive error term whose variance depends on the predictors: $Y = X_1^2 + \exp((X_2 + X_3/3)) + X_4 - X_5 + (0.5 + X_2^2/2 + X_5^2/2) \times \varepsilon$, $\varepsilon \sim N(0, 1)$.
3. (M3). A model with a multiplicative non-Gassisan error term: $Y = (5 + X_1^2/3 + X_2^2 + X_3^2 + X_4 + X_5) * \exp(0.5 \times \varepsilon)$, where $\varepsilon \sim \mathbb{I}_{\{U<0.5\}} \times N(-2, 1) + \mathbb{I}_{\{U>0.5\}} \times N(2, 1)$ with $U \sim \text{Uniform}(0, 1)$, $X \in \mathbb{R}^{30}$.
4. (M4). A mixture of two normal distributions: $Y = \mathbb{I}_{\{U<0.5\}} N(-X_1, 0.25^2) + \mathbb{I}_{\{U>0.5\}} N(X_1, 0.25^2)$, where $U \sim \text{Uniform}(0, 1)$.

In each of the models above, the covariate vector X is generated from standard multivariate normal distribution.

The neural networks used in the simulations are specified as follows. For models (M1)-(M3), the generator network has 1 hidden layer with width 50, and the discriminator has 2 hidden layers with widths (50, 25); for models (M4), the generator network has 2 hidden layers with widths (40, 15), and the discriminator has 2 hidden layers with widths (50, 25). For the values of m, the dimension of the noise random vector η, we set $m = 3$ for models (M1)-(M3), and $m = 4$ for model (M4).

For the conditional density estimation method NNKCDE, the tuning parameters are chosen using cross-validation. The bandwidth of the conditional kernel density estimator CKDE is chosen by the rule-of-thumb using the standard formula $h_j = 1.06\sigma_j n^{-1/(2*K+J)}$ where σ_j is a measure of spread of the jth continuous variable defined as $\min(SD, IQR/1.349)$, n the number of observations, K the order of the kernel, and J the number of continuous variables. The basis expansion based method FlexCode uses Fourier basis. The maximum number of bases is 40 and the actual number of bases is selected using cross-validation.

We calculate the mean squared error (MSE) of the estimated conditional mean $E(Y|X)$ and the estimated conditional standard deviation $\text{SD}(Y|X)$. We use a test dataset $\{x_1, \ldots, x_k\}$ of size $k = 2000$. The MSE of the estimated conditional mean is $\text{MSE(mean)} = (1/k) \sum_{i=1}^{k} [\hat{E}(Y|X = x_i) - E(Y|X = x_i)]^2$. For GCDS, the estimate of $E(Y|X = x)$ is based on (5) using Monte Carlo. For other methods, the estimate is calculated by numerical integration $\hat{E}(Y|x) = \int y f(y|x) dy$ using 1000 subdivisions. Similarly, the MSE of the estimated conditional standard deviation is $\text{MSE(sd)} = (1/k) \sum_{i=1}^{k} [\hat{\text{SD}}(Y|X = x_i) - \text{SD}(Y|X = x_i)]^2$.

For GCDS, we first generate J samples $\{\eta_j : j = 1, \ldots, J\}$ from the reference distribution P_η and calculate conditional samples $\{\hat{G}(\eta_j, x_i), j = 1, \ldots, J\}$. We take $J = 10,000$. The estimated conditional standard deviation is calculated as the sample standard deviation of the conditional samples. The estimated conditional standard deviation of other methods are computed by numerical integration $\hat{\text{SD}}(Y|x_i) = \sqrt{\int [y - \hat{E}(Y|x_i)]^2 f(y|x_i) dy}$ using 1000 subdivisions.

Table 1. Mean squared error (MSE) of the estimated conditional mean, the estimated standard deviation and the corresponding simulation standard errors (in parentheses).

		GCDS	NNKCDE	CKDE	FlexCode
M1	Mean	**0.259**(0.015)	1.367(0.010)	0.491(0.024)	0.610(0.008)
	SD	**0.022**(0.004)	0.258(0.004)	0.233(0.005)	0.170(0.007)
M2	Mean	**0.312**(0.017)	4.668(0.046)	1.707(0.060)	2.408(0.063)
	SD	**0.247**(0.012)	0.793(0.008)	0.857(0.017)	2.384(0.602)
M3	Mean	**3.377**(0.196)	4.926(0.080)	39.084(0.929)	9.015(0.341)
	SD	**2.082**(0.126)	8.131(0.235)	15.70(0.488)	11.53(1.140)
M4	Mean	0.016(0.003)	**0.004**(0.001)	0.063(0.002)	0.006(0.002)
	SD	**0.027**(0.005)	0.131(0.001)	0.076(0.001)	0.046(0.001)

NOTE: The smallest MSEs are in bold font.

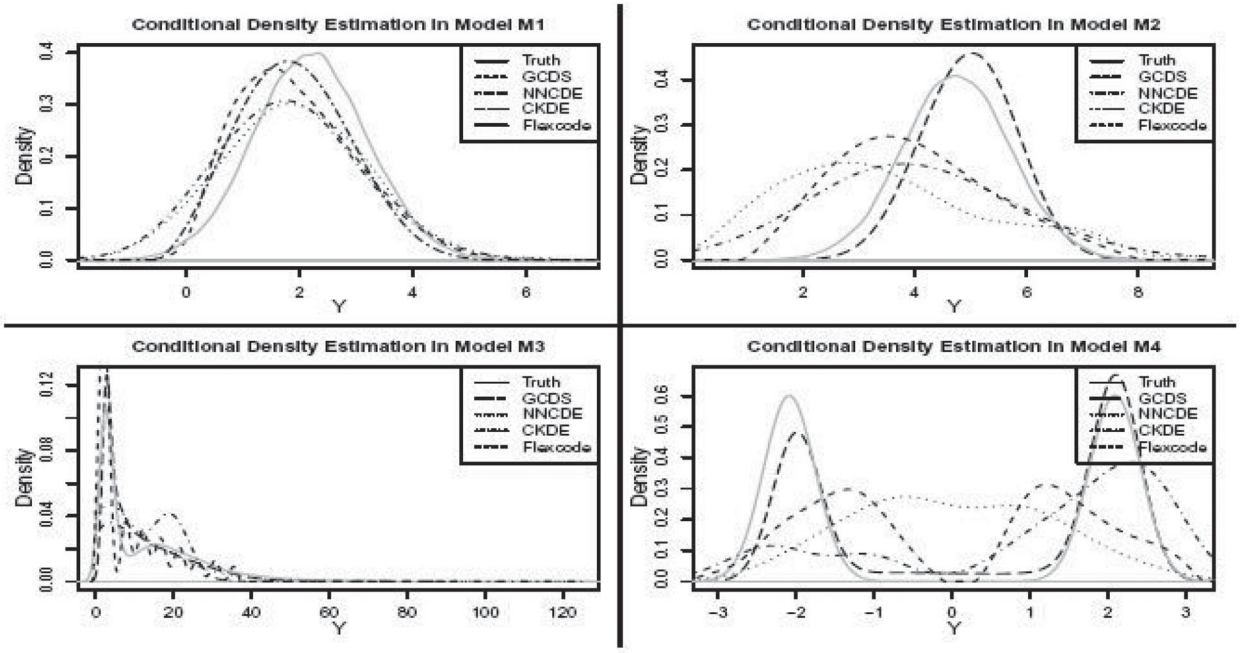

Figure 1. Comparison of density estimation in models (M1) to (M4). The conditional density function corresponding to GCDS is estimated using the samples generated from GCDS with kernel smoothing.

We repeat the simulations 10 times. The average MSEs and simulation standard errors are summarized in Table 1. We see that, comparing with CKDE and FlexCode and NNKCED, GCDS has the smallest MSEs for estimating conditional mean and conditional SD in most cases.

In Figure 1, we display the estimated conditional density functions for a randomly generated value of X. The true conditional distributions of $Y|X$ for models (M1) to (M4) are: (M1), $N(2.19, 1)$; (M2), $N(4.75, 0.96^2)$; (M3): the mixture of half 7.42 $\times$ log-normal$(-1, 0.5^2)$ and half 7.42 $\times$ log-normal$(1, 0.5^2)$; (M4): the mixture of half $N(-2.09, 0.25^2)$ and half $N(2.09, 0.25^2)$. The conditional density function corresponding to GCDS is estimated based the samples generated based on GCDS using kernel smoothing. This plot shows that GCDS yields better conditional density estimates than CKDE, NNCDE and FlexCode.

6.2. The Abalone Dataset

The abalone dataset is available at UCI machine learning repository (Dua and Graff 2017). It contains the number of rings of abalone and other physical measurements. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope, a time-consuming process. Other measurements, which are easier to obtain, are used to predict the number of rings that determines the age. This dataset contains 9 variables. They are *sex, length, diameter, height, whole weight, shucked weight, viscera weight, shell weight*, and *rings*. Except for the categorical variable *sex*, all the other variables are continuous. The variable *sex* codes three groups: female, male and infant, since the gender of an infant abalone is not known. The sample size is 4177. We use 90% of the data for training and 10% of the data as the testing set. The neural networks used in the analysis are specified as follows: the generator network is a fully connected network with 2 hidden layers with widths 50 and 20; the discriminator networks is a fully connected network with 2 hidden layers with widths 50 and 25. The dimension of the noise vector of the noise vector is set to be $m = 5$.

We take *rings* as the response $Y \in \mathbb{R}$ and the other measurements as the covariate vector $X \in \mathbb{R}^9$. Figure 2 shows the estimated conditional density based on the training dataset for 3 groups: female, male and infant, at the value of the group means of the remaining covariates. We see that the values of *rings* of the infant group are smaller than those of the female and male groups. The female abalones tend to have slightly higher numbers of rings than male abalones. In addition, the conditional distributions are skewed to the right for all the three groups.

To examine the prediction performance of the estimated conditional density, we construct the 90% prediction interval for the number of rings of each abalone in the testing set. The prediction intervals are shown in Figure 3. The actual number of rings is plotted as a solid dot. The actual coverage for all 418 cases in the testing set is 89.71%, close to the nominal level of 90%. The numbers of rings that are not covered by the prediction intervals are the largest ones in each group.

6.3. MNIST Handwritten Digits

We now illustrate the application of GCDS to high-dimensional data problems and demonstrate that it can easily handle the models when either of both of X and Y are high-dimensional. The data example we use is the MNIST handwritten digits dataset (LeCun, Cortes, and Burges 2010), which contains 60,000 images for training and 10,000 images for testing. The images are stored in 28 $\times$ 28 matrices with gray color intensity from 0 to 1. Each image is paired with a label in $\{0, 1 \ldots, 9\}$. We

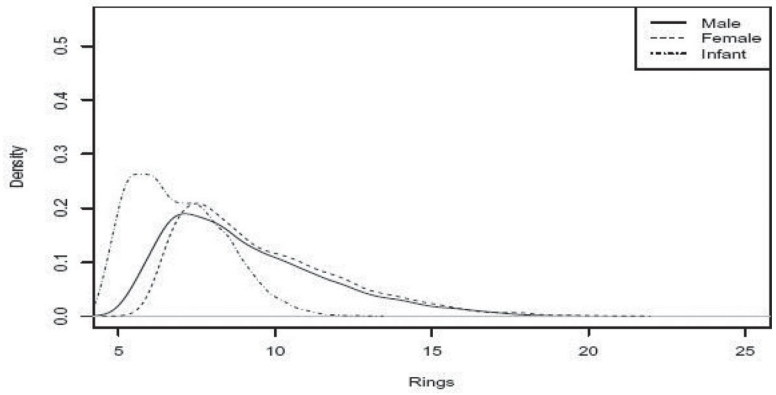

Figure 2. Estimated conditional densities for the female, male, and infant groups in the abalone dataset. Each line represents the kernel conditional density estimation based on the samples generated using GCDS given the group average values of the covariates.

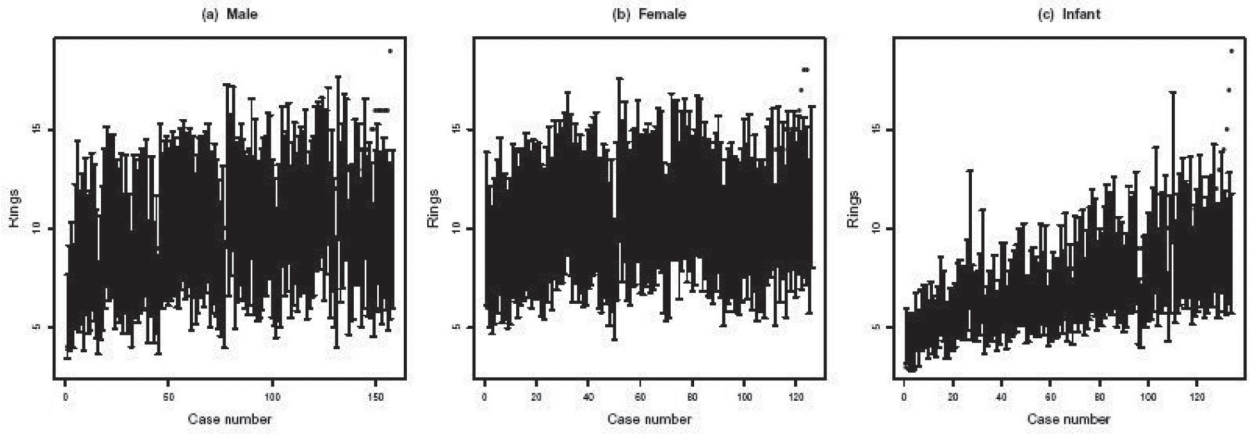

Figure 3. The prediction intervals for the testing set. All 418 abalones in the testing set are divided into three groups, (a) male, (b) female, and (c) infant.

use GCDS to perform two tasks: generating images from labels and reconstructing the missing part of an image.

Generating images from labels. We generate images of handwritten digits given the label. In this problem, the predictor X is a categorical variable representing the ten digits: $\{0, 1, \ldots, 9\}$ and the response Y represents 28×28 images. We use one-hot vectors in $\mathbb{R}^{10}$ to represent these ten categories. So the dimension of X is 10 and the dimension of Y is $28 \times 28 = 784$. The response $Y \in [0, 1]^{28\times 28}$ is a matrix representing the intensity values. For the discriminator D, we use a convolutional neural network (CNN) with 3 convolution layers with 128, 256, and 256 filters to extract the features of the image and then concatenate with the label information (repeated 10 times to match the dimension of the features). The concatenated information is sent to a fully connected layer and then to the output layer. For the generator G, we concatenate the label information with random noise of dimension 100. Then it is fed to a CNN with 3 deconvolution layers with 256, 128, and 1 filters.

Figure 4 shows the real images (left panel) and generated images (right panel). We see that the generated images are similar to the real images and it is hard to distinguish the generated ones from the real images. Also, there are some differences in the generated images, reflecting the random variations in the generating process.

Reconstructing missing part of an image. We now illustrate using GCDS to reconstruct an image when part of the image is missing with the MNIST dataset. Suppose we only observe $1/4, 1/2$, or $3/4$ of an image and would like to reconstruct the missing part of the image. For this problem, let X be the observed part of the image and let Y be the missing part of the image. Our goal is to reconstruct Y based on X. For the discriminator, we use two convolutional networks to process X and Y separately. The filters are then concatenated together and fed into another convolution layer and fully connected layer before output. For the generator, X is processed by a fully connected layer followed by 3 deconvolution layers.

In Figure 5, three plots from left to right corresponds to the situations when $1/4$, $1/2$, and $3/4$ of an image are given. In each subplot, the first column contains the true images in the testing set. The gray boxes show the given areas. Each row contains six reconstructions of the image. The digits "0," "1," and "7" are easy to reconstruct. Even when only 1/4 of their images are given, GCDS can correctly reconstruct them. The other digits are more difficult. If only 1/4 of their images are given, then it is impossible to reconstruct them. However, as the given area increases from 1/4 to 1/2 and then 3/4 of the images, GCDS is able to reconstruct the images correctly, and the reconstructed images become less variable and more similar to the true image. For example, for the digit "2," if only the left lower 1/4 of the image is given, the reconstructed images tend to be incorrect; the reconstruction is only successful when 3/4 of the image is given.

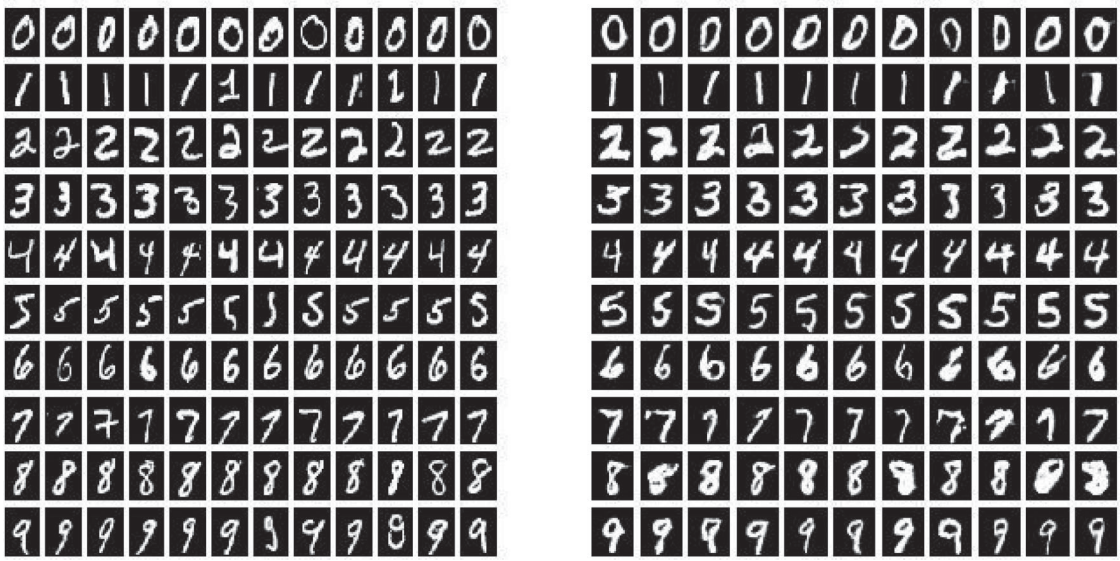

Figure 4. MNIST dataset: real images (left panel) and generated images given the labels (right panel).

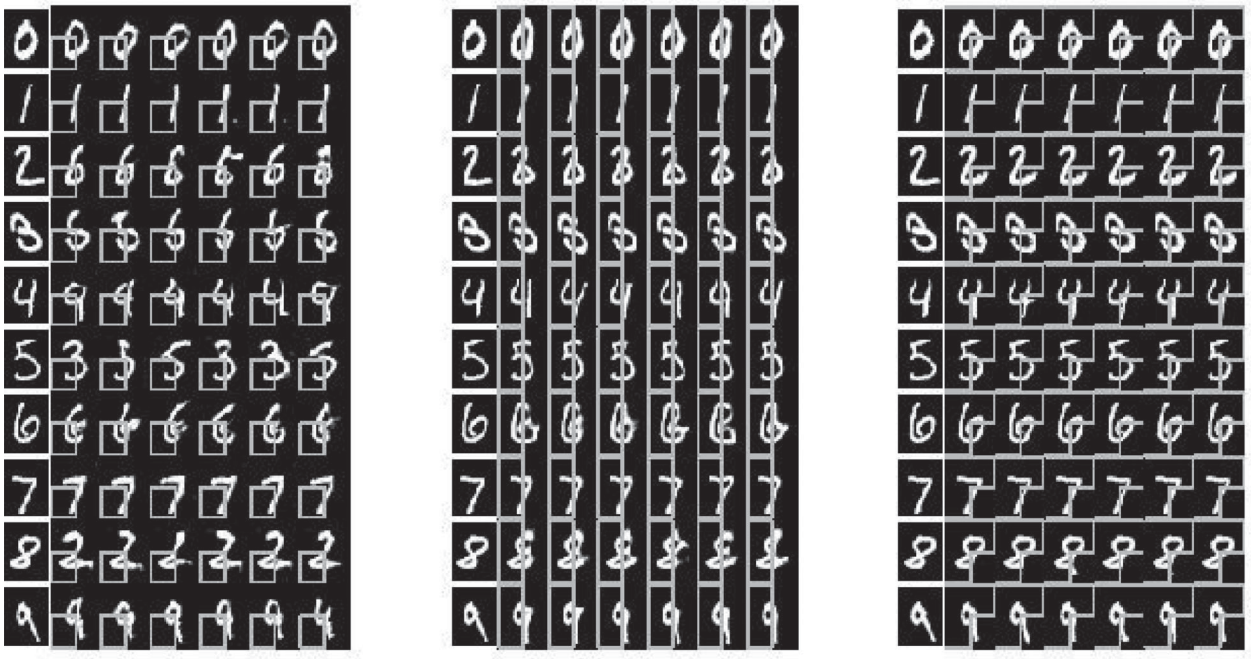

Figure 5. Reconstructed images given partial image in MNIST dataset. The first column in each panel consists of the true images, the other columns give the constructed images. In the left panel, the left lower 1/4 of the image is given; in the middle panel, the left 1/2 of the image is given; in the right panel, 3/4 of the image is given.

7. Conclusion

In this article, we propose GCDS, a generative approach to sampling from a conditional distribution. We provide theoretical support for GCDS by showing that the conditional generator converge in distribution to the underlying target conditional distribution under mild conditions. Our numerical experiments demonstrate that it works well in a variety of situations from the standard nonparametric conditional density estimation problems to more complex image data problems.

Several questions deserve further investigation. First, it would be interesting to derive the convergence rate of the sampling distribution that strengthens the consistency result in Theorem 4.1 and provide conditions under which the number of coefficients in the deep neural network is allowed to be greater than the sample size. Second, while the conditional generator provides all the information of the conditional distribution, it is still useful to obtain an estimate of the functional form of the conditional density. How to obtain a good estimator of the conditional density function in the present framework is an open question, especially when the dimension of (X, Y) is high. Finally, as a proof of concept we demonstrated that GCDS yields reasonable results for some simple image analysis tasks with the MNIST dataset. It would be interesting to apply GCDS to more complex image analysis problems. We intend to study these problems in the future.

Supplementary Material

Additional numerical experiment results and technical details are provided in the online supplementary material.

Acknowledgments

We thank the editor, the associate editor and two anonymous reviewers for their constructive comments, which led to significant improvements in the paper.

Funding

The work of X. Zhou and J. Huang is supported in part by the U.S. National Science Foundation (grant DMS-1916199). Y. Jiao is supported in part by the National Science Foundation of China (No. 11871474) and by the research fund of KLATASDSMOE of China. The work of J. Liu is supported by the Duke-NUS Medical School (R-913-200-098-263) and MOE2018-T2-2-006 from the Ministry of Education, Singapore.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X. (2016), "Tensorflow: A System for Large-Scale Machine Learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283. [1843]

Ali, S. M., and Silvey, S. D. (1966), "A General Class of Coefficients of Divergence of One Distribution From Another," *Journal of the Royal Statistical Society*, Series B, 28, 131–142. [1840]

Austin, T. (2015), "Exchangeable Random Measures," *Annales de l'I.H.P. Probabilités et statistiques*, 51, 842–861. [1838]

Bartlett, P. L., Harvey, N., Liaw, C., and Mehrabian, A. (2019), "Nearly-Tight VC-Dimension and Pseudodimension Bounds for Piecewise Linear Neural Networks," *Journal of Machine Learning Research*, 20, 1–17. [1841,1842]

Bartlett, P. L., and Mendelson, S. (2002), "Rademacher and Gaussian Complexities: Risk Bounds and Structural Results," *Journal of Machine Learning Research*, 3, 463–482. [1841,1842]

Bauer, B., and Kohler, M. (2019), "On Deep Learning as a Remedy for the Curse of Dimensionality in Nonparametric Regression," *Annals of Statistics*, 47, 2261–2285. [1842]

Bengio, Y., Courville, A., and Vincent, P. (2013), "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 1798–1828. [1837]

Bhattacharya, P. K., and Gangopadhyay, A. K. (1990), "Kernel and Nearest-Neighbor Estimation of a Conditional Quantile," *Annals of Statistics*, 18, 1400–1415. [1837]

Bishop, C. (2006), *Pattern Recognition and Machine Learning*, New York: Springer. [1837]

Bott, A.-K., and Kohler, M. (2017), "Nonparametric Estimation of a Conditional Density," *Annals of the Institute of Statistical Mathematics*, 69, 189–214. [1837]

Chen, M., Jiang, H., Liao, W., and Zhao, T. (2019), "Nonparametric Regression on Low-Dimensional Manifolds Using Deep Relu Networks." arXiv:1908.01842. [1842]

Chen, X., Linton, O., and Robinson, P. M. (2001), "The Estimation of Conditional Densities," in *Asymptotics in Statistics and Probability*, ed. M. L. Puri, Festschrift for George Roussas, Berlin, Boston: De Gruyter, pp. 71–84. [1837]

Cook, R. (1998), *Regression Graphics: Ideas for Studying Regressions Through Graphics*, Wiley Series in Probability and Statistics, New York: Wiley. [1837]

Dalmasso, N., Pospisil, T., Lee, A. B., Izbicki, R., Freeman, P. E., and Malz, A. I. (2020), "Conditional Density Estimation Tools in Python and R With Applications to Photometric Redshifts and Likelihood-Free Cosmological Inference," *Astronomy and Computing*, p. 100362. [1843]

Dua, D., and Graff, C. (2017), "UCI Machine Learning Repository." Available at *https://archive.ics.uci.edu/ml/index.php*. [1844]

Fan, J., Yao, Q., and Tong, H. (1996), "Estimation of Conditional Densities and Sensitivity Measures in Nonlinear Dynamical Systems," *Biometrika*, 83, 189–206. [1837]

Fan, J., and Yim, T. H. (2004), "A Crossvalidation Method for Estimating Conditional Densities," *Biometrika*, 91, 819–834. [1837]

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014), "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, 27, 2672–2680. [1838,1839]

Goodfellow, I. J., Bengio, Y., and Courville, A. (2016), *Deep Learning*, Cambridge, MA: MIT Press. *http://www.deeplearningbook.org*. [1841]

Hall, P., Racine, J., and Li, Q. (2004), "Cross-Validation and the Estimation of Conditional Probability Densities," *Journal of the American Statistical Association*, 99, 1015–1026. [1843]

Hall, P., and Yao, Q. (2005), "Approximating Conditional Distribution Functions Using Dimension Reduction," *Annals of Statistics*, 33, 1404–1421. [1837]

Hyvärinen, A., and Pajunen, P. (1999), "Nonlinear Independent Component Analysis: Existence and Uniqueness Results," *Neural Networks*, 12, 429–439. [1837]

Izbicki, R., and Lee, A. B. (2016), "Nonparametric Conditional Density Estimation in a High-Dimensional Regression Setting," *Journal of Computational and Graphical Statistics*, 25, 1297–1316. [1838]

Izbicki, R., and Lee, A. B. (2017), "Converting High-Dimensional Regression to High-Dimensional Conditional Density Estimation," *Electronic Journal of Statistics*, 11, 2800–2831. [1838,1843]

Jiao, Y., Shen, G., Lin, Y., and Huang, J. (2021), "Deep Nonparametric Regression on Approximately Low-Dimensional Manifolds." arXiv 2104.06708. [1842]

Kallenberg, O. (2002), *Foundations of Modern Probability*, 2nd ed., New York: Springer-Verlag. [1838]

Keziou, A. (2003), "Dual Representation of φ-Divergences and Applications," *Comptes Rendus Mathématique*, 336, 857–862. [1840]

Kingma, D. P., and Ba, J. (2015), "Adam: A Method for Stochastic Optimization," in *Proceedings of the 3rd International Conference on Learning Representation*. [1843]

Kohler, M., and Langer, S. (2020), "On the Rate of Convergence of Fully Connected Very Deep Neural Network Regression Estimates." arXiv: 1908.11133. (to appear in *Annals of Statistics*). [1842]

LeCun, Y., Cortes, C., and Burges, C. (2010), "Mnist Handwritten Digit Database," *AT&T Labs* [Online]. Available at *http://yann.lecun.com/exdb/mnist*, 2. [1844]

Li, K.-C. (1991), "Sliced Inverse Regression for Dimension Reduction," *Journal of the American Statistical Association*, 86, 316–327. [1837]

Mirza, M., and Osindero, S. (2014), "Conditional Generative Adversarial Nets," cite arxiv:1411.1784. [1839]

Nakada, R., and Imaizumi, M. (2019), "Adaptive Approximation and Estimation of Deep Neural Network With Intrinsic Dimensionality," arXiv:1907.02177. [1842]

Nguyen, X., Wainwright, M. J., and Jordan, M. I. (2010), "Estimating Divergence Functionals and the Likelihood Ratio by Convex Risk Minimization," *IEEE Transactions on Information Theory*, 56, 5847–5861. [1840]

Nowozin, S., Cseke, B., and Tomioka, R. (2016), "f-gan: Training Generative Neural Samplers Using Variational Divergence Minimization," in *Advances in Neural Information Processing Systems*, pp. 271–279. [1840]

Rockafellar, T. R. (1970), *Convex Analysis*, Princeton, NJ: Princeton University Press. [1840]

Rosenblatt, M. (1969), "Conditional Probability Density and Regression Estimators," in Multivariate Analysis II, Ed. P. R. Krishnaiah, pp. 25–31, New York: Academic Press. [1837]

Schmidt-Hieber, J. (2020), "Nonparametric Regression Using Deep Neural Networks With ReLU Activation Function (With Discussion)," *Annals of Statistics*, 48, 1916–1921. [1842]

Scott, D. W. (1992), *Multivariate Density Estimation: Theory, Practice and Visualization*, New York: Wiley. [1837]

Shen, G., Jiao, Y., Lin, Y., Horowitz, J. L., and Huang, J. (2021a), "Deep Quantile Regression: Mitigating the Curse of Dimensionality Through Composition." arXiv: 2107.04907. [1842]

Shen, G., Jiao, Y., Lin, Y., and Huang, J. (2021b), "Robust Nonparametric Regression With Deep Neural Networks." arXiv: 2107.10343. [1842]

Shen, Z., Yang, H., and Zhang, S. (2020), "Deep Network Approximation Characterized by Number of Neurons," *Communications in Computational Physics*, 28, 1768–1811. [1841]

Sugiyama, M., Takeuchi, I., Suzuki, T., Kanamori, T., Hachiya, H., and Okanohara, D. (2010), "Least-Squares Conditional Density Estimation," *IEICE Transactions on Information and Systems*, 93, 583–594. [1837]

Tsybakov, A. (2008), *Introduction to Nonparametric Estimation*, Springer Science & Business Media, New York: Springer. [1842]

Van der Vaart, A. W., and Wellner, J. A. (1996), *Weak Convergence and Empirical Processes: With Applications to Statistics*, New York: Springer. [1842]